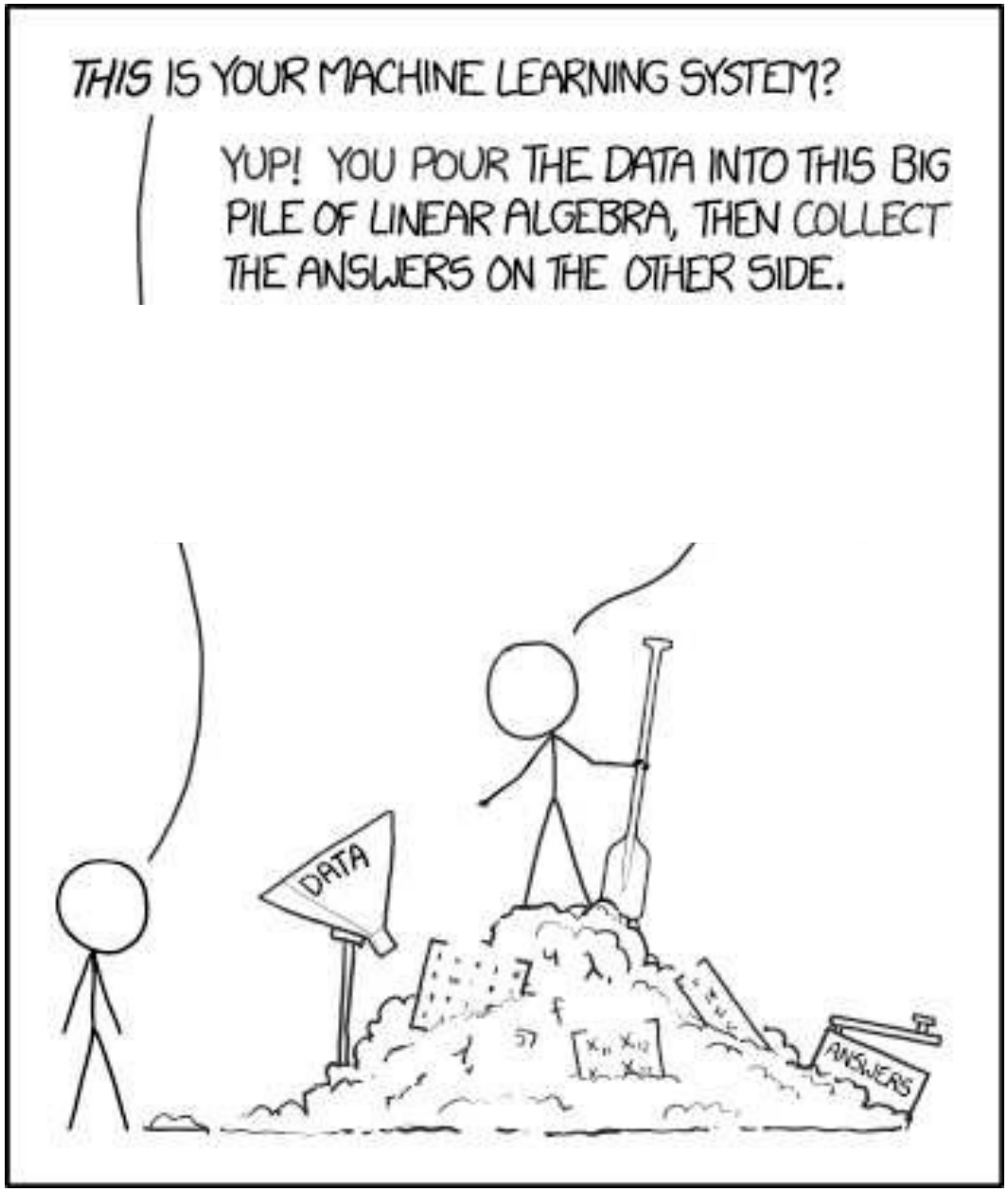
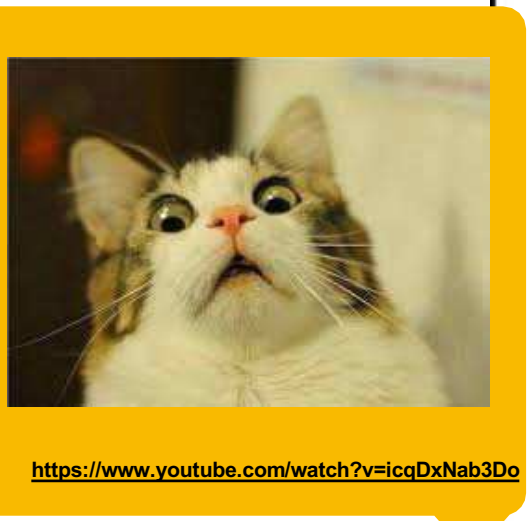
Interpreting Machine Learning

Dan Ofer



https://xkcd.com/

https://xkcd.com/

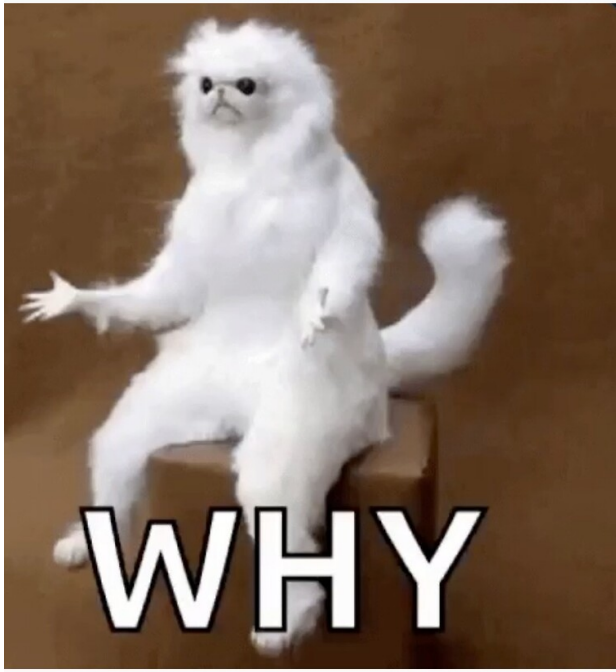


https://www.youtube.com/watch?v=icqDxNab3Do

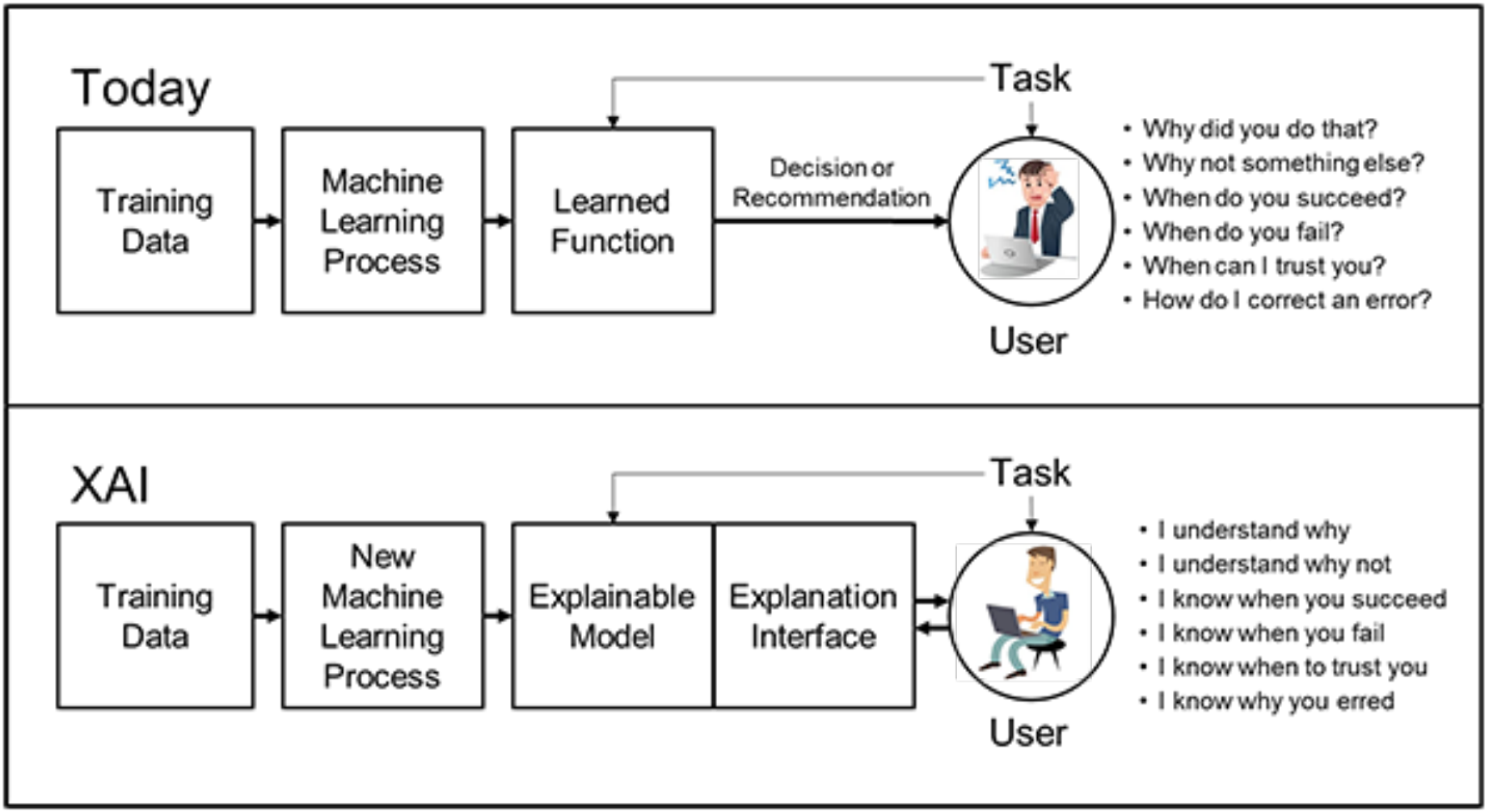
https://xkcd.com/

Agenda

- **When** and **why** interpretability
- **Dimensions** of interpretability
- **Overview** of interpretability methods



Explainable AI - XAI



Source: DARPA xAI: https://www.darpa.mil/program/explainable-artificial-intelligence

This is not a new problem. Why now?

Complexity and wide-spread use in new areas

In the past:

- Mostly linear or rule-based models, simpler data
- Less use of ML, especially in sensitive or regulated domains!

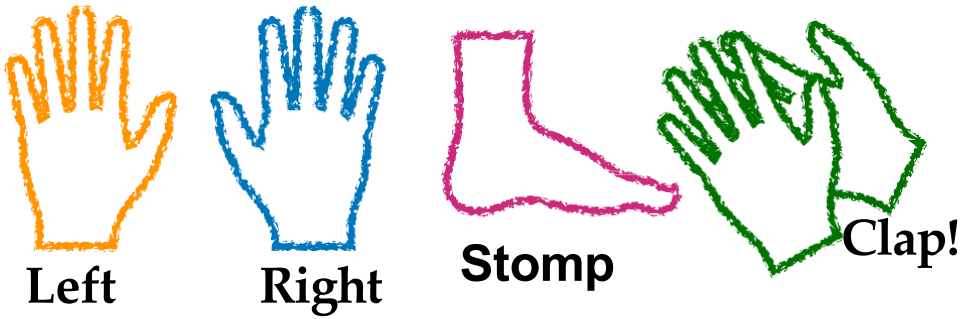
I heard you can just use decision trees…

Can we go home now?

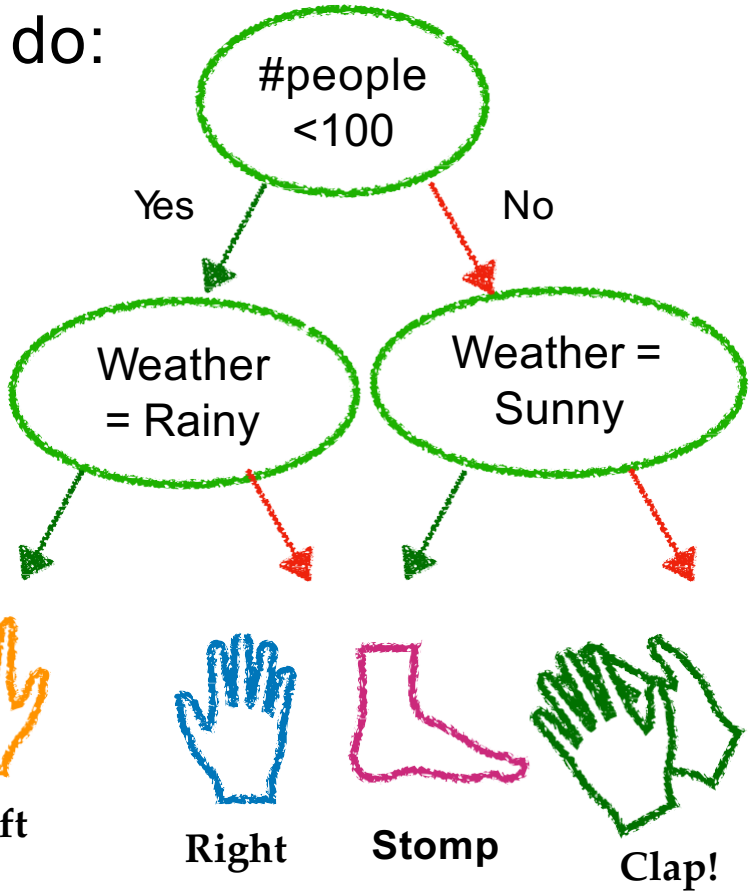
http://www.ogroup.com.au/raise-your-hand-when-you-should-and-why-you-should/

Experiment.

- I will show you a decision tree. Follow the right path given a data point, and you do:

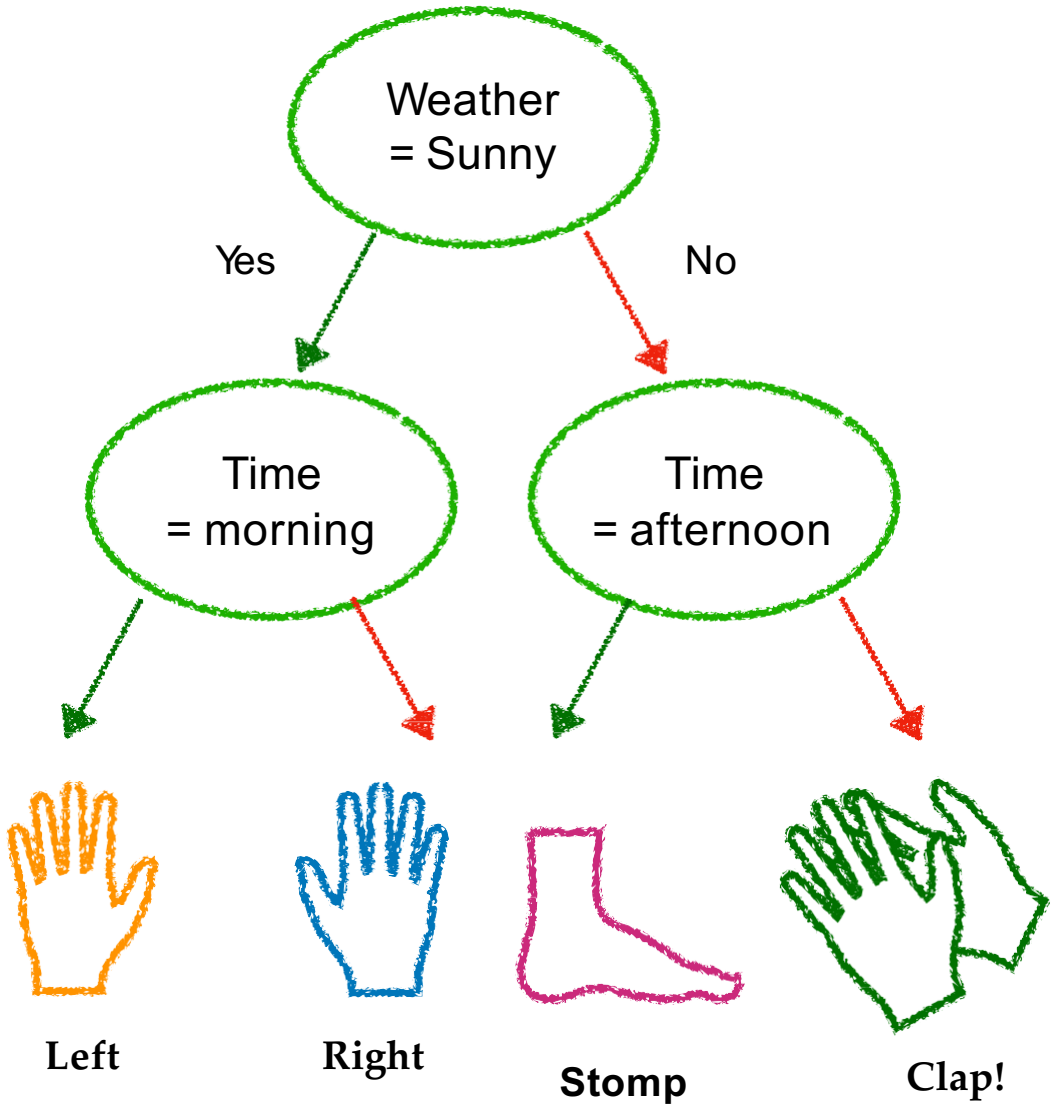


Data = [Sunny, 200]



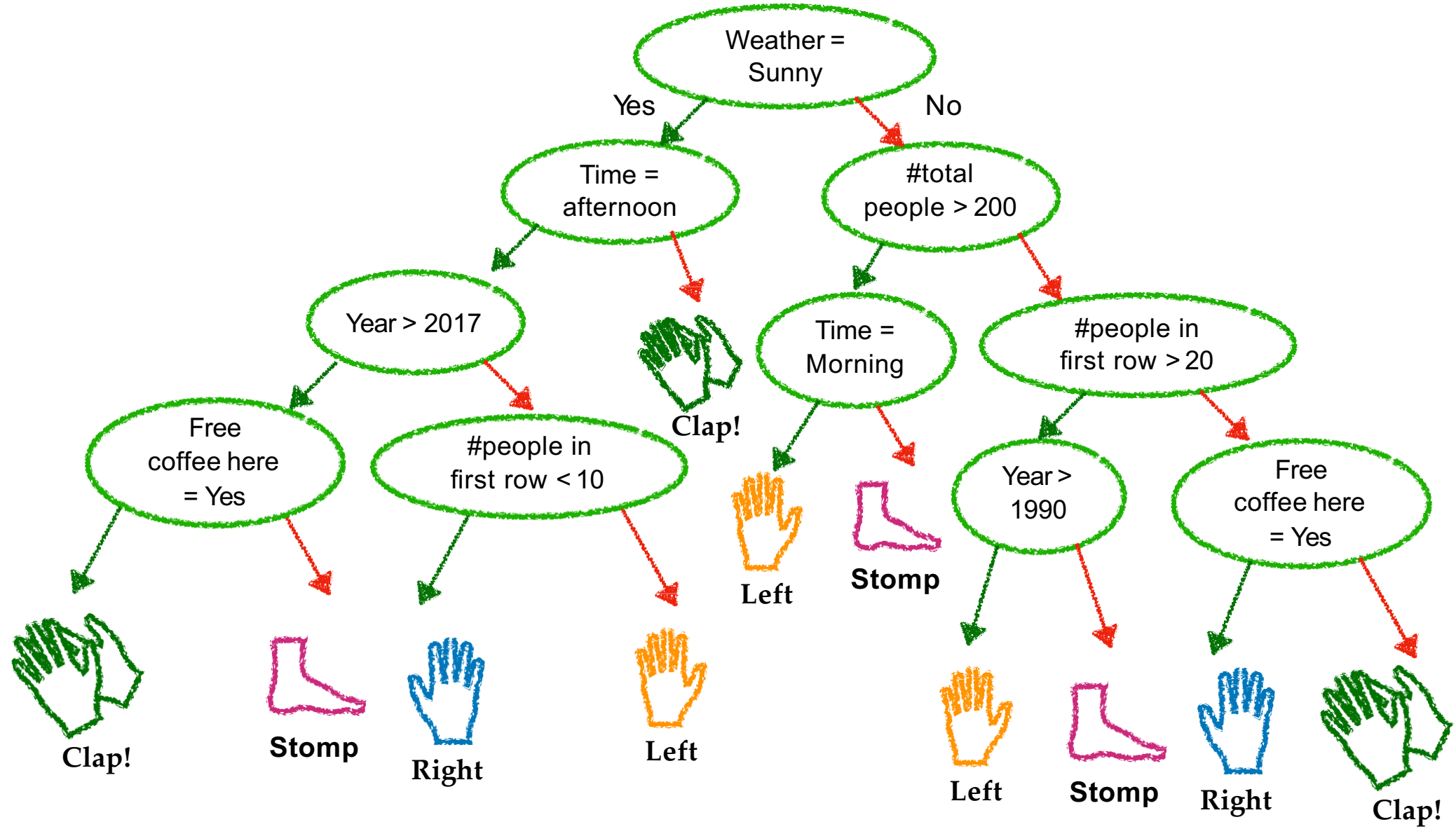
Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Sample decision tree #1



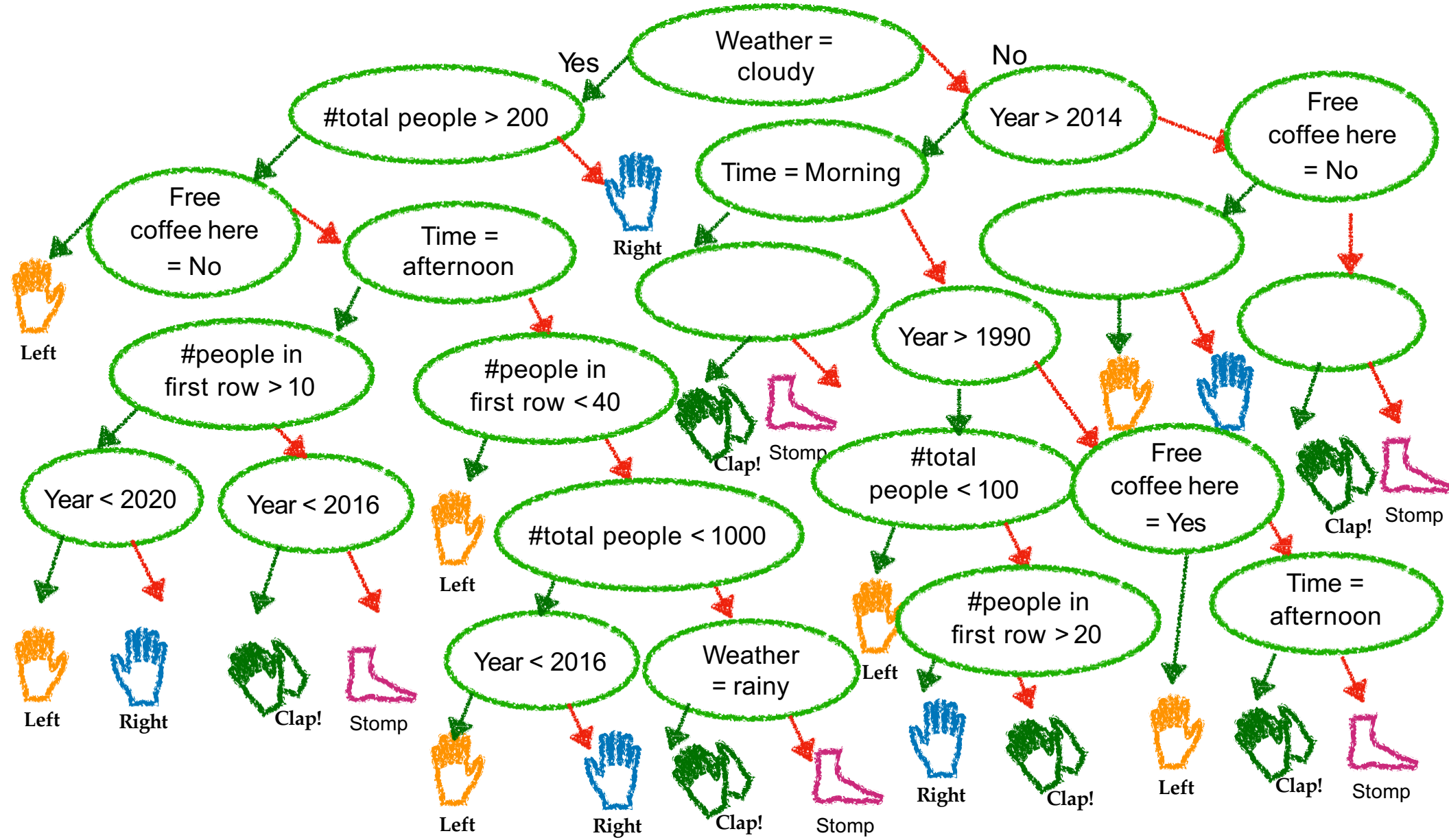
Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Sample decision tree #2



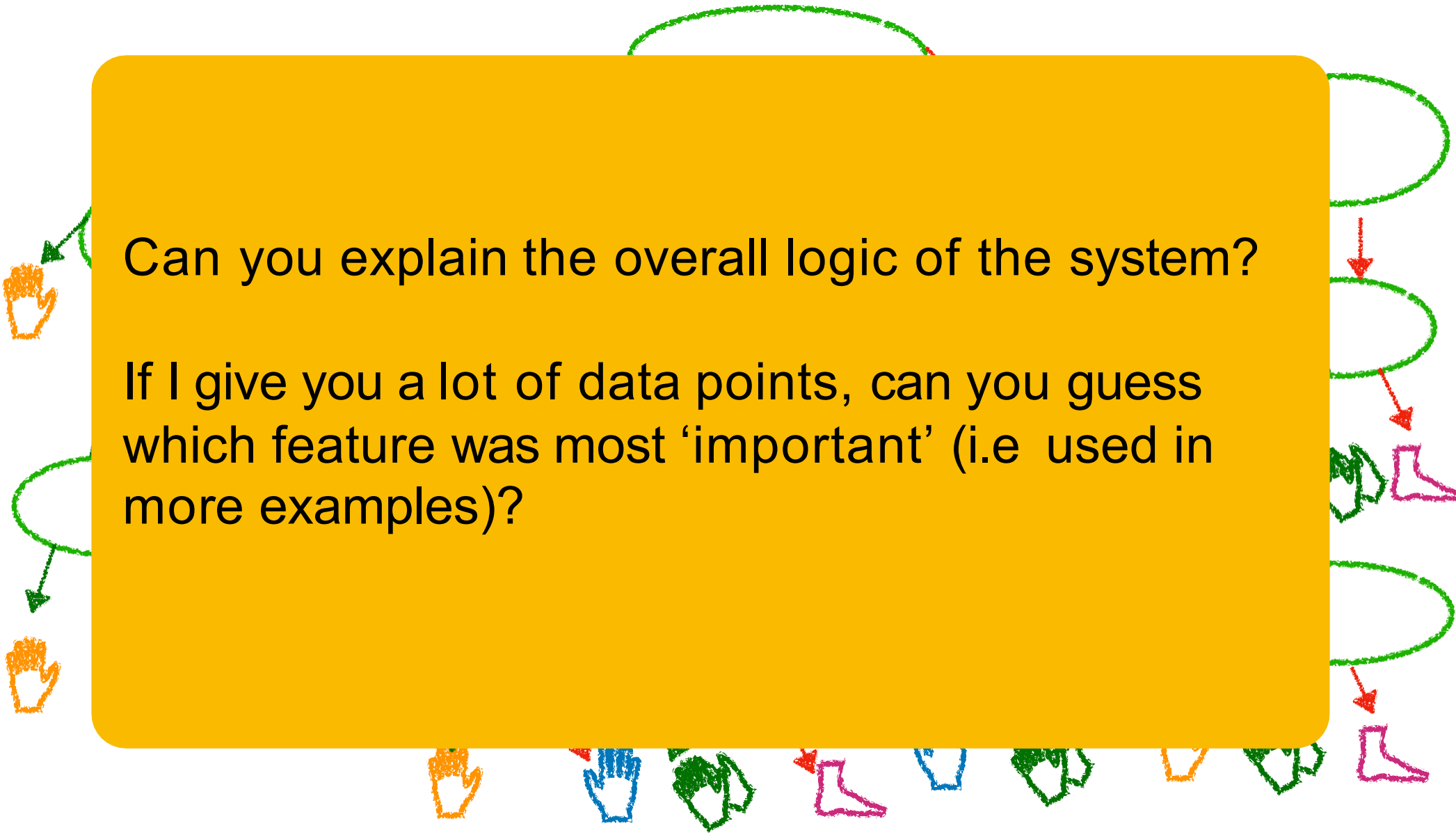
Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Sample decision tree #3



Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Sample decision tree #3



Can you explain the overall logic of the system?

If I give you a lot of data points, can you guess which feature was most ‘important’ (i.e used in more examples)?

Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Common misunderstanding:
Decision trees and linear models are always interpretable.

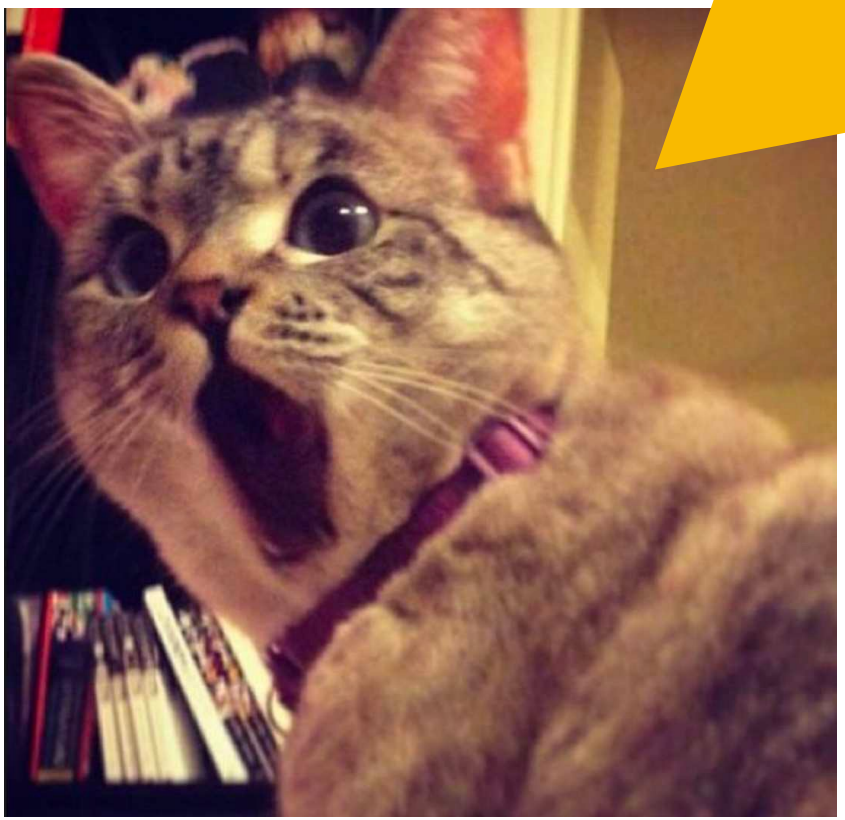
Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Do we need a different model? How about rule lists?

If (sunny and hot)	then	go swim
Else if (sunny and cold)	then	go ski
Else if (wet and weekday)	then	go work
Else if (free coffee)	then	attend tutorial
Else if (cloudy and hot)	then	go swim
Else if (snowing)	then	go ski
Else if (New Rick and Morty)	then	watch TV
Else if (paper deadline)	then	go work
Else if (hungry)	then	go eat
Else if (tired)	then	watch TV
Else if (advisor might come)	then	go work
Else if (code running)	then	watch TV
Else	then	go work

Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Decision trees and rule lists don't work?!



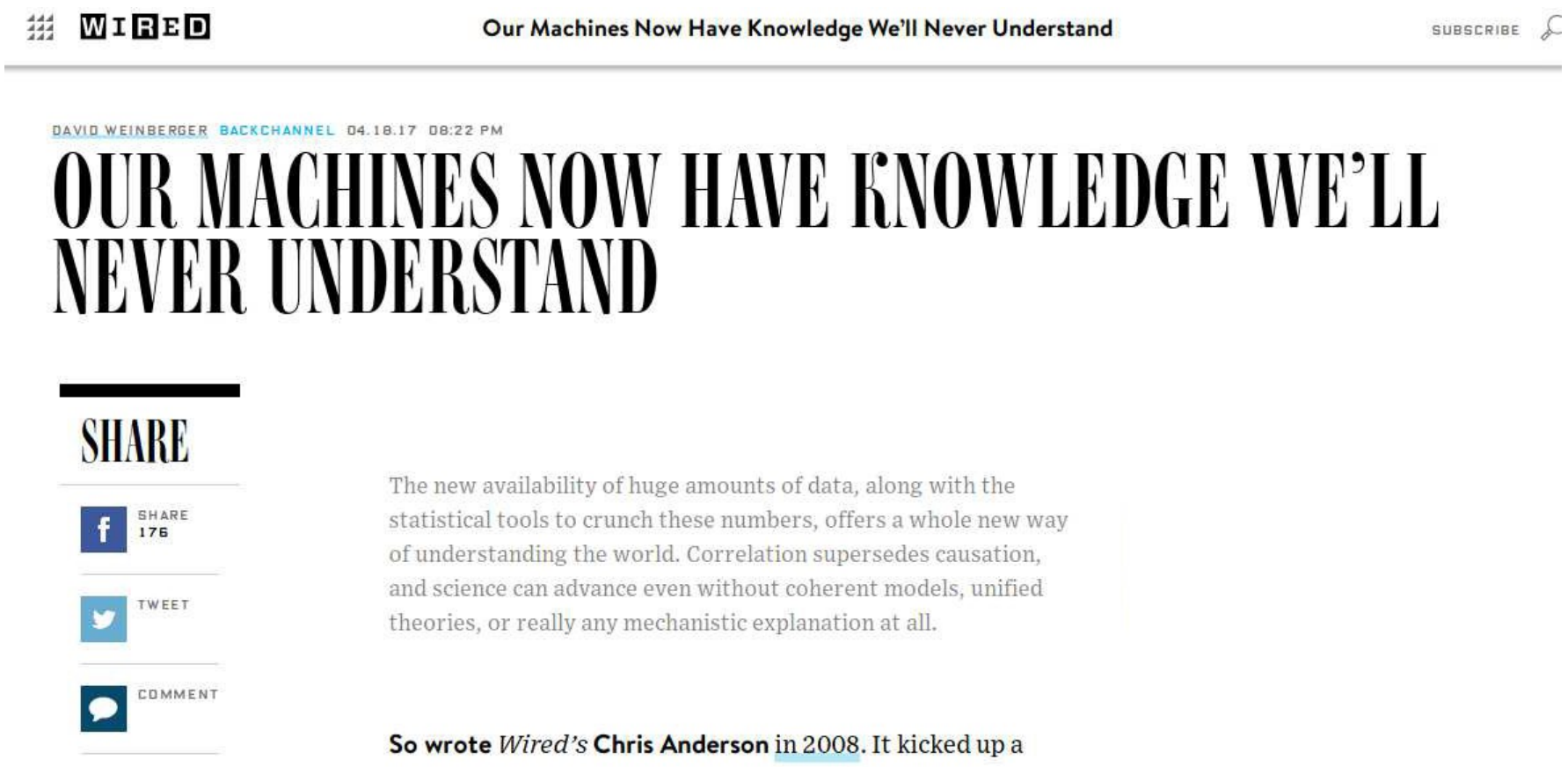
Decision trees, rule lists or other methods may work for your case!

The point here is that there is no one-size-fits-all method.

http://blog.xfree.hu/myblog.tvn?SID=&from=20&pid=&pev=2016&pho=02&pnap=&kat=1083&searchkey=&hol=&n=sarkadykati

Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Is interpretability possible at all?



WIRED Our Machines Now Have Knowledge We'll Never Understand SUBSCRIBE

DAVID WEINBERGER BACKCHANNEL 04.18.17 08:22 PM

OUR MACHINES NOW HAVE KNOWLEDGE WE'LL NEVER UNDERSTAND

SHARE

SHARE 176

TWEET

COMMENT

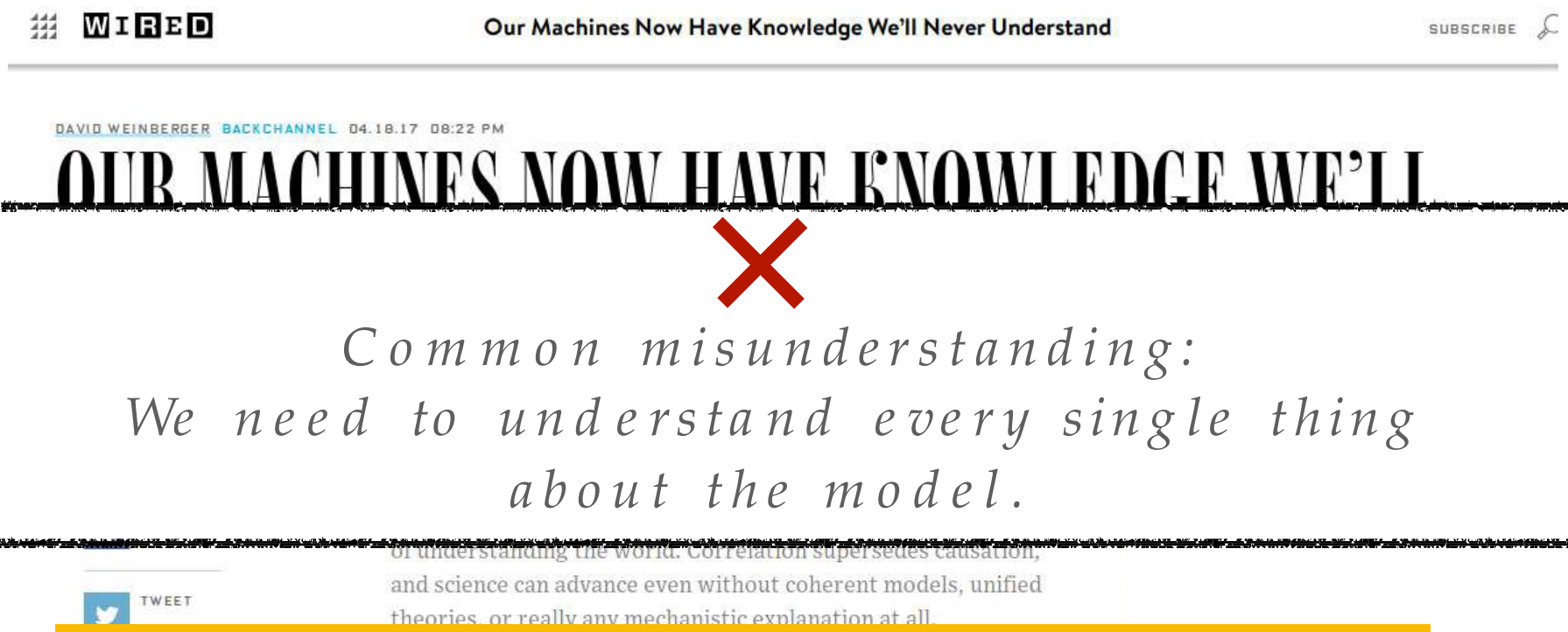
The new availability of huge amounts of data, along with the statistical tools to crunch these numbers, offers a whole new way of understanding the world. Correlation supersedes causation, and science can advance even without coherent models, unified theories, or really any mechanistic explanation at all.

So wrote *Wired's* **Chris Anderson** in 2008. It kicked up a

https://www.wired.com/story/our-machines-now-have-knowledge-well-never-understand/

Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Is interpretability possible at all?



Common misunderstanding:
We need to understand every single thing about the model.

Key Point:
Interpretability is NOT about understanding all bits and bytes of the model for all data points.

It is about knowing enough for your goals/downstream tasks.

Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Why interpretability? High-Stakes Decisions

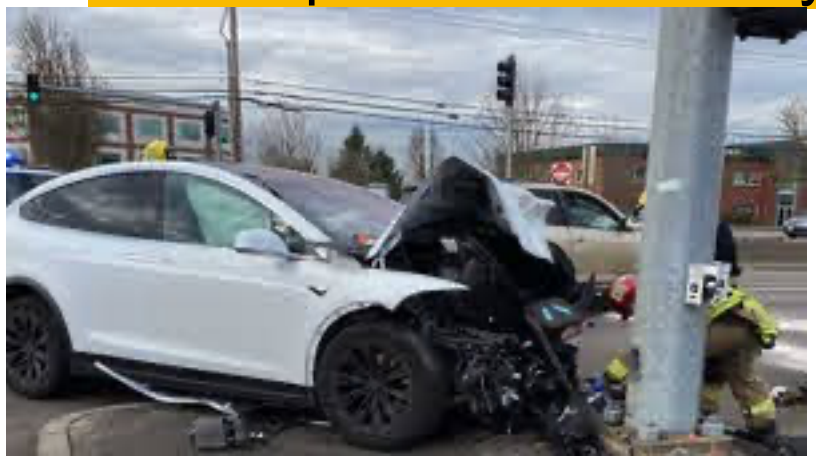
Healthcare: What treatment to recommend to the patient?

Criminal Justice: Should the defendant be released on bail?

High-Stakes Decisions: Impact on human well-being

Why interpretability?

Example 1: Cost/Safety



- Safety – risk of injury/death, regulations
  - Medicine, healthcare
  - Autonomous vehicles

- "Cost" - high cost of each error
  - Churn
  - Real estate

  - Slow feedback

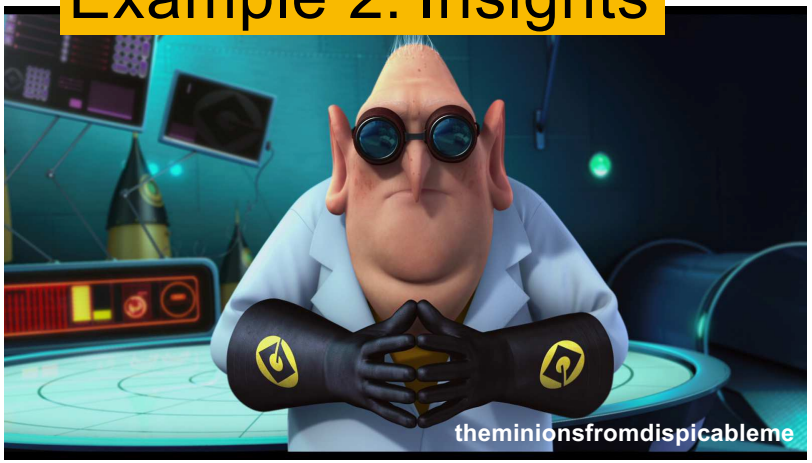
High cost, but fast & frequent feedback:

- Advertisements – instant feedback: click or no click
- Stock market trading: an Octopus can (and does) do it; microseconds to know the outcome!

- Doesn't apply when *safety* is an issue

Why interpretability?

Example 2: Insights

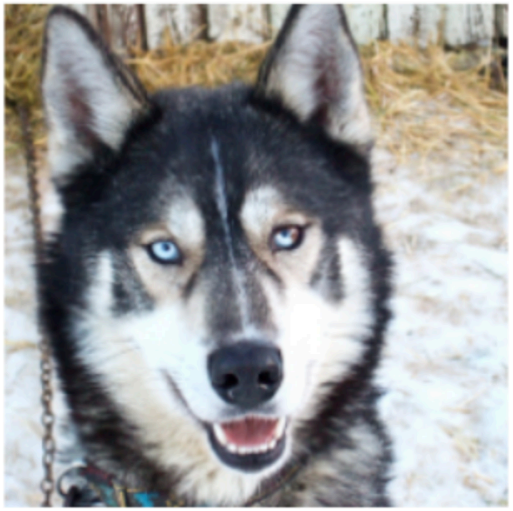


2: New understandings on the problem

- “What drives user churn?”
- “Why doesn’t this medicine affect those tumors?”
- “What makes diabetics at risk of deterioration?”

Why interpretability?

Example 3: mismatched objectives



(a) Husky classified as wolf

(b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

Mismatched objectives = **Target leaks**

"Cheating the exam, instead of actually learning the material"

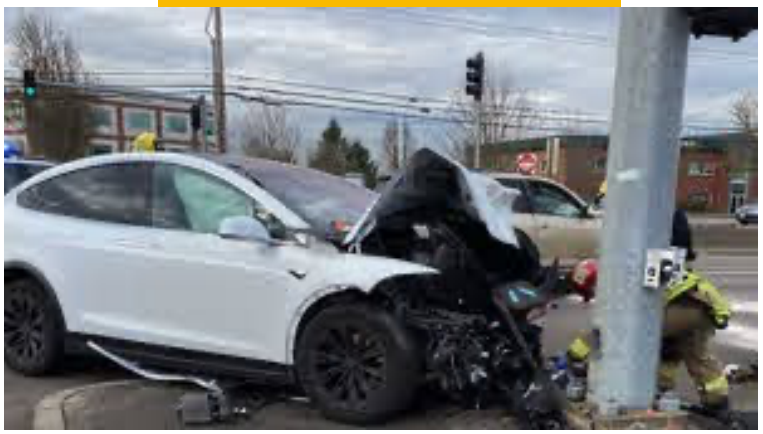
Examples:

- "MonthlySalary" column when predicting "YearlySalary"
- "MinutesLate" when predicting "IsLate"
- "NumOfLatePayments" when predicting "ShouldGiveLoan"
- "Index number" (proxy for time, or ordered data labelling)
- Date_last_modified >= "Churn_date" (i.e use of future data)
- x-rays learning image "metadata": https://www.datarobot.com/blog/identifying-leakage-in-computer-vision-on-medical-images

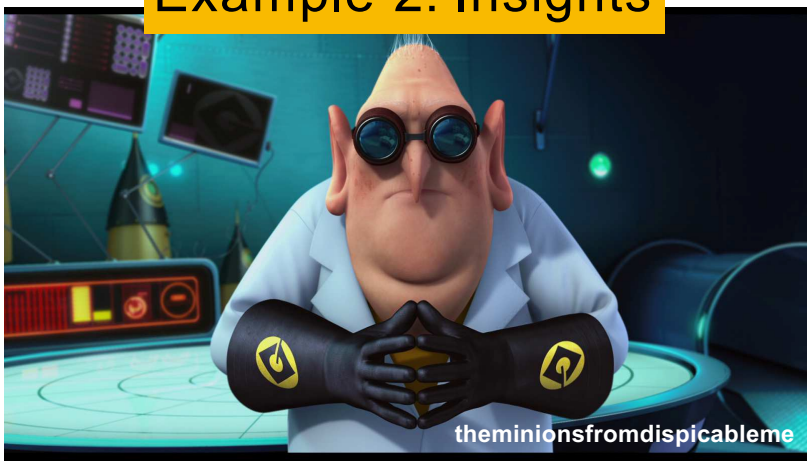
More examples: https://aiukraine.com/wp-content/uploads/2018/09/12_00-Yuriy-Guts-Target-Leakage-in-Machine-Learning-.pdf

Why interpretability?

Example 1: Cost/ Safety



Example 2: Insights



Example 3: mismatched objectives

Example 4: Regulations & Discrimination

SR 11-7: Guidance on Model Risk Management

BOARD OF GOVERNORS
OF THE FEDERAL RESERVE SYSTEM
WASHINGTON, D.C. 20551

Why Explainability: Laws against Discrimination

Citizenship

Immigration Reform and Control Act

Sex

Equal Pay Act of 1963;
Civil Rights Act of 1964

Age

Age Discrimination in Employment Act of 1967

Race

Civil Rights Act of 1964

Disability status

Rehabilitation Act of 1973;
Americans with Disabilities Act of 1990

And more...

GDPR Concerns Around Lack of Explainability in AI

"

Companies should commit to ensuring systems that could fall under GDPR, including AI, will be compliant. The threat of ***sizeable fines of €20 million or 4% of global turnover*** *provides a sharp incentive.*

Article 22 of GDPR empowers ***individuals*** *with the* ***right to demand an explanation of how an AI system made a decision that affects them.***

"

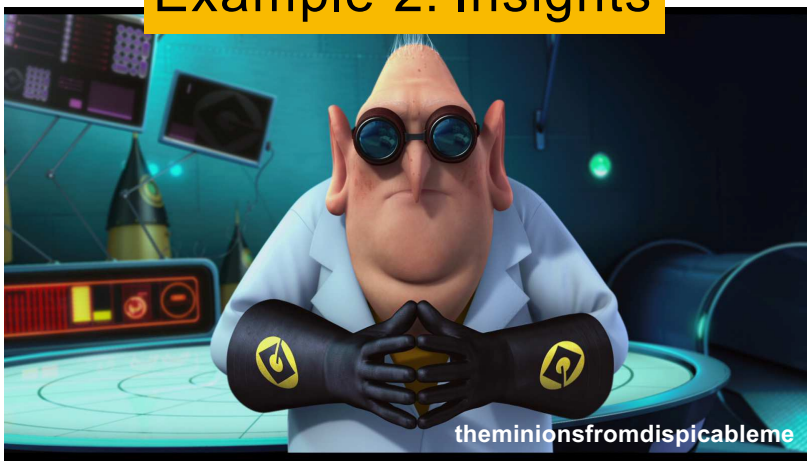
- European Commision

VP, European Commision

Why interpretability?

Example 1: Cost/ Safety

Example 2: Insights



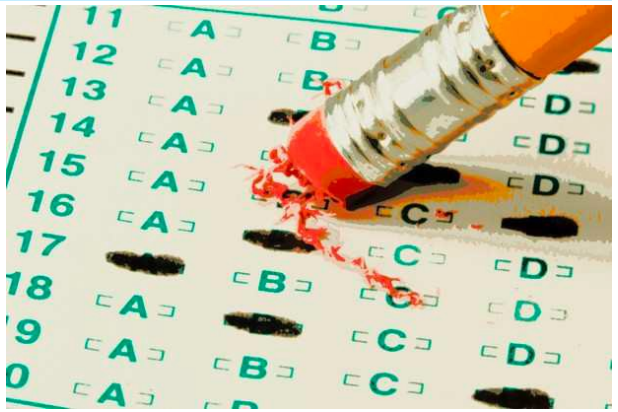
Fundamental **underspecification** in the problem

Common misunderstanding:
More data or more clever algorithm will solve interpretability.

When we may **not** need interpretability

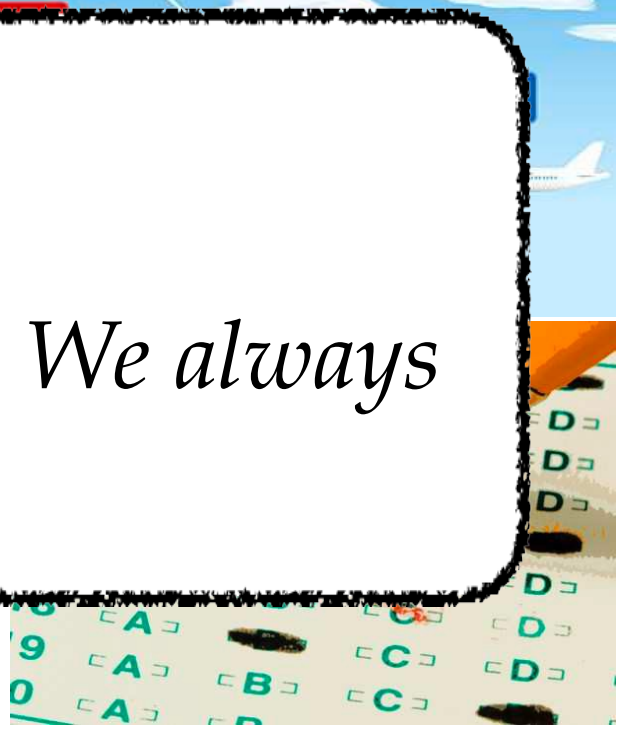
- No significant consequences or when predictions are all you need.

- Sufficiently well-studied problem



http://www.cinemablend.com/pop/Netflix-Using-Amazon-Cloud-Explore-Artificial-Intelligence-Movie-Recommendations-62248.html

When we may **not** need interpretability

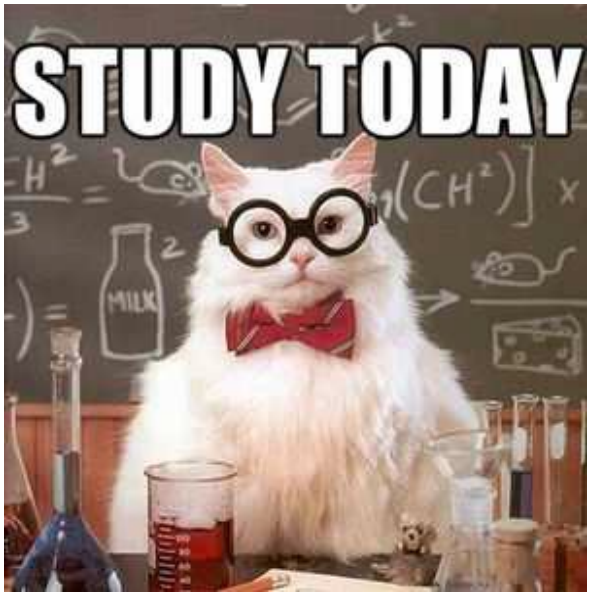


Common misunderstanding: We always need Interpretability

http://www.cinemablend.com/pop/Netflix-Using-Amazon-Cloud-Explore-Artificial-Intelligence-Movie-Recommendations-62248.html

Agenda

- **When** and **why** interpretability
- **Dimensions** of interpretability
- **Overview** of interpretability methods



www.memecenter.com

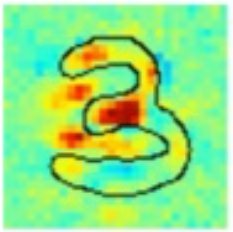
Based on slides by Been Kim, Google Brain - https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf

Dimensions of Interpretability

Different dimensions of "interpretability"

prediction

"Explain why a certain pattern x has been classified in a certain way f(x)."



model

"What would a pattern belonging to a certain category typically look like according to the model."

data

"Which dimensions of the data are most relevant for the task."

Source: MICCAI'18 Tutorial on Interpretable Machine Learning

Feature "importance" is underspecified

"Important" features = ?

- Important in general? For sub-populations? For a specific prediction?
- If I remove it and model is not affected; not important??
  - Not if it's correlated! E.g. age and pension plan. Ice-cream sales, summer and drowning
- Is it important if model A uses it but model B doesn't?
- Depends on the question we care about – business/domain knowledge

Levels of explanations

- Global vs Local
  - Overall importance vs explaining a single prediction
- Model based vs Model Agnostic
  - Feature is always important, or only important for model X

Model based vs Model Agnostic

- Explains only a specific, fitted model ?

Example –

- Model based feature importance
- SHAP – Shapley Values
- LIME – Locally Interpretable Model Agnostic Explanations
- Attention
- Gradients (DL)

- Applies to any model

Example –

- Statistical tests of feature importance (ANOVA, T-test)
- Mutual Information
- Causal analysis

Global vs Local

Overall importance

- ”What are the most important features?
- Explanations for all possible outcomes

Examples:

- Model “Feature Importance”
- Feature selection
- SHAP summary

Explain an individual prediction?

- “What affected predictions for a specific sample?”

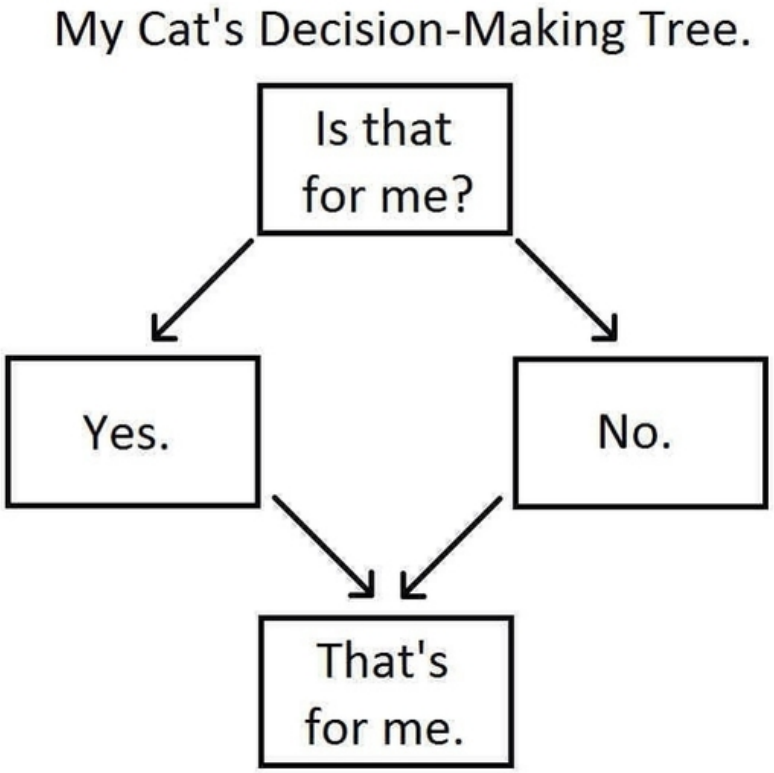
Examples:

- SHAP – per instance
- LIME
- Heatmaps (DL)
- Cluster of similar predictions

Local explanations

Explain individual predictions

"Why did the model predict Y, for this **specific** instance"



https://memeguy.com/photo/30851/my-cats-decision-making-tree

Real World Scenario: Bail Decision

U.S. police make about 12M arrests each year

Release vs. Detain is a high-stakes decision

Pre-trial detention can go up to 9 to 12 months

Consequential for jobs & families of defendants as well as crime

Release on Bail: 0/1?

Bail Decision

Release

Detain

Fail to appear
Non-violent crime
Violent crime

Unfavorable

None of the above

Favorable

Spends time in jail

Judge is making a prediction:
Will the defendant commit 'crime' if released on bail?

Experiment: Provide explanation

If Current-Offense = Felony:
- If Prior-Felony = Yes and Prior-Arrests ≥ 1, then Crime
- If Crime-Status = Active and Owns-House = No and Has-Kids = No, then Crime
- If Prior-Convictions = 0 and College = Yes and Owns-House = Yes, then No Crime

If Current-Offense = Misdemeanor and Prior-Arrests > 1:
- If Prior-Jail-Incarcerations = Yes, then Crime
- If Has-Kids = Yes and Married = Yes and Owns-House = Yes, then No Crime
- If Lives-with-Partner = Yes and College = Yes and Pays-Rent = Yes, then No Crime

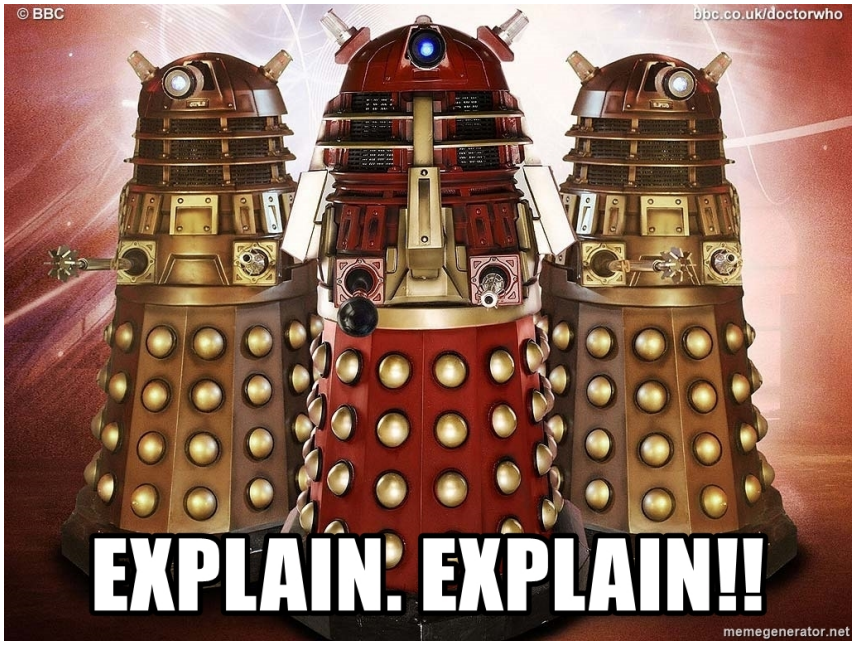
If Current-Offense = Misdemeanor and Prior-Arrests ≤ 1:
- If Has-Kids = No and Owns-House = No and Moved_10times_5years = Yes, then Crime
- If Age ≥ 50 and Has-Kids = Yes, then No Crime

Default: No Crime

Judges were able to make decisions 2.8 times faster and 38% more accurately (compared to no explanation and only prediction) !

Agenda

- **When** and **why** interpretability
- **Dimensions** of interpretability
- **Overview** of interpretability methods



Feature importance & Explainability Techniques

Universal Feature importance

How important is a feature (e.g. “age ==18”) overall – **regardless of the model used**

- EDA (Exploratory Data Analysis)
- Statistical tests – Mutual information, t-test, chi square, variance etc’
- Support: How many missing values, how many instances covered (absolute and %)
- Redundancy of feature - Colinearity, correlation with other feats
- Domain knowledge (“age is important in healthcare, and is the causal driver for other features”)
- “Lift” – Target distribution given feature value(s).
- E.g.: When gender==Male: 40% of target are True, vs 10% in background -> feature has 4X lift for target==True
- Good features are **universal** (model agnostic)
- Gold standard for ***actionability***!

Model based Feature importance

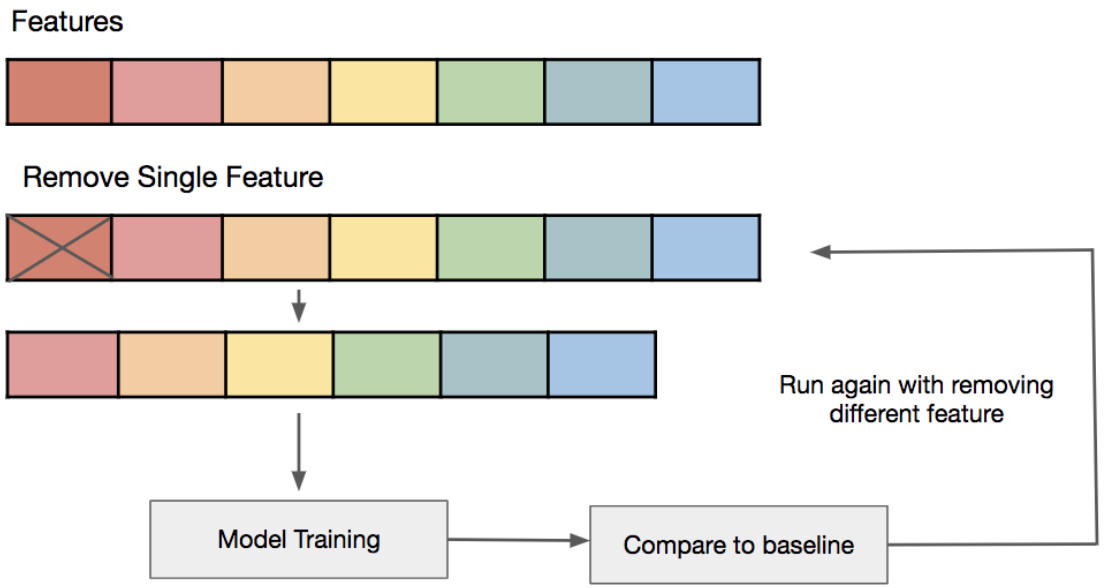
Explain a specific, fitted model's ranking of features

- Model based feature importance (coefficients, gini, weights etc')
- Ablation
- Permutation
- SHAP – Shapley Values
- LIME – Locally Interpretable Model Agnostic Explanations
- Attention
- Gradients (DL)

Feature Ablation

TL;DR: Remove a feature, retrain model, measure degredation in model performance. Repeat for all features.

=> "What are the most important features for my model"



Drawbacks:

- Slowwwwwww
- What about correlated features?
  - When features are collinear, dropping one will have little effect on the models performance because it can get the same information from a correlated feature
- Retraining model – no guarantee it will behave the same

(2) Feature Ablation: Recursive Feature Elimination

Improvement: Remove many features at once, starting with ”weakest” features, retrain model, repeat.

+ Far faster (but still not fast). Works “close enough”
- Dependent on model’s internal ranking
- Sensitive/misguided with correlated features
- RFECV: tends to overfit in terms of estimated model performance with subset of features

+ Easy to use (can use Cross Validation):

```
from sklearn.feature_selection import RFECV
rfecv = RFECV(estimator=svm, cv=5, scoring='accuracy')
rfecv.fit(X, y)
```

Trick for correlated feats: Cluster first and keep 1 feat per cluster https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance_multicollinear.html

Permutation Importance

Big Idea: Instead of dropping features and retraining model, replace them with noise and check degradation of predictions. Run multiple times for stability.

- + Only need to train model a few times
- Same Pros (+) and cons (–) apply as for ablation (bad for correlated features)
- Permutation importance does **not** reflect to the intrinsic predictive value of a feature by itself but **how important this feature is for a particular model**
- See also: Boruta

```
from sklearn.inspection import permutation_importance
from eli5.permutation_importance import get_score_importances
https://github.com/scikit-learn-contrib/boruta_py
```

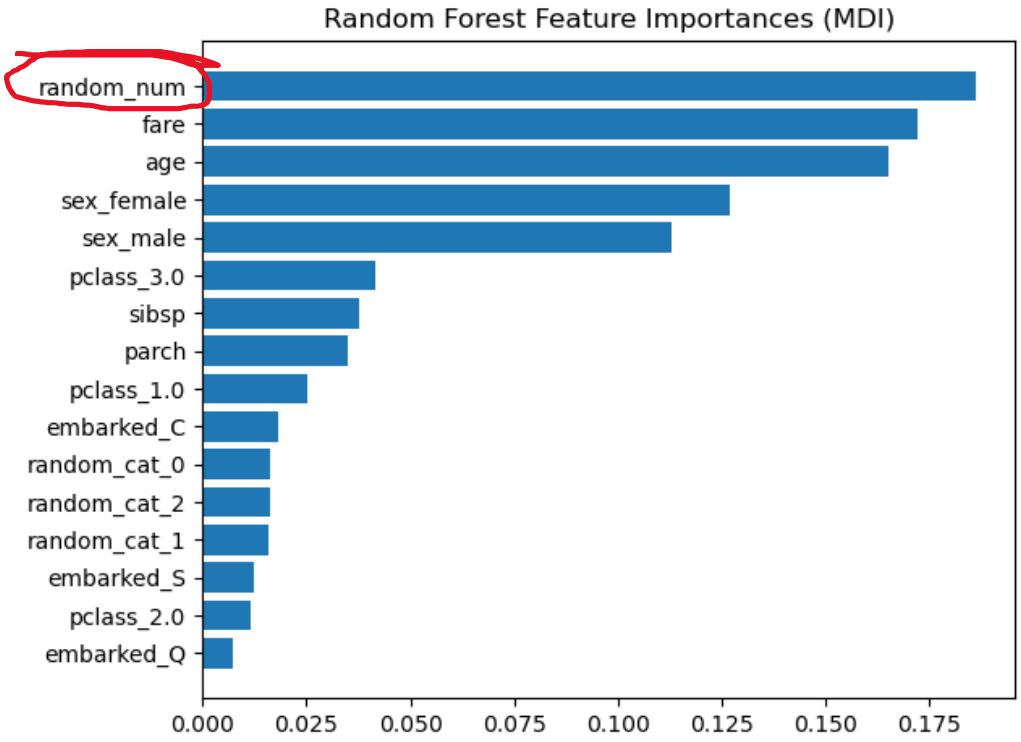
Permutation Importance VS Model importance

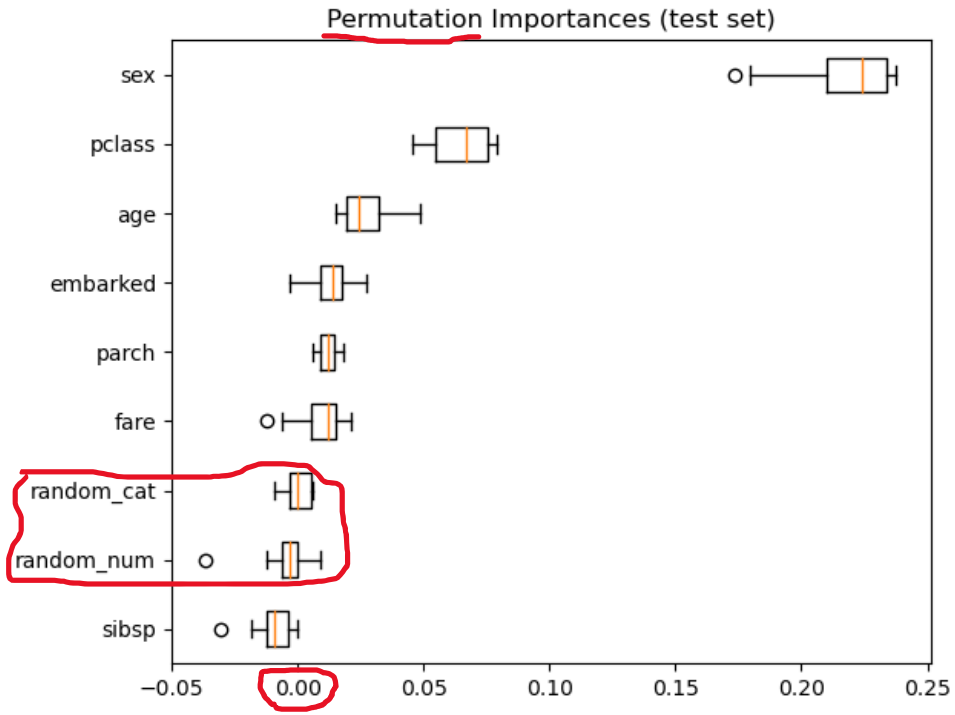
Decision trees' feature importance is based on mean decrease in impurity (entropy/gini)

Impurity-based feature importance for trees are **strongly biased** and **favor high cardinality features** (typically numerical features) over low cardinality features such as categoricals with a small number of possible categories.

Additionally, doesn't check if attributes have predictive power (e.g. on held out dataset).

Permutation Importance VS Model importance: Titanic



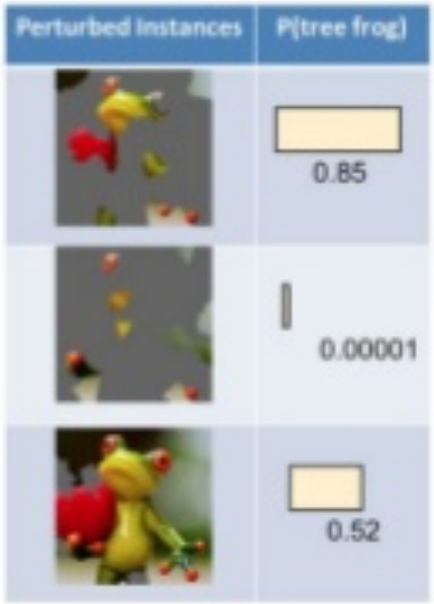


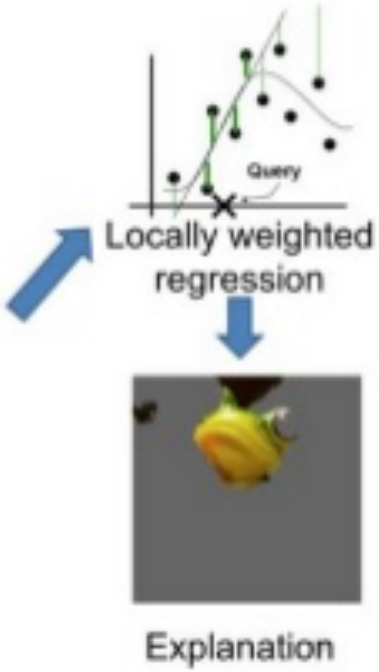
RF test accuracy: 81.7%

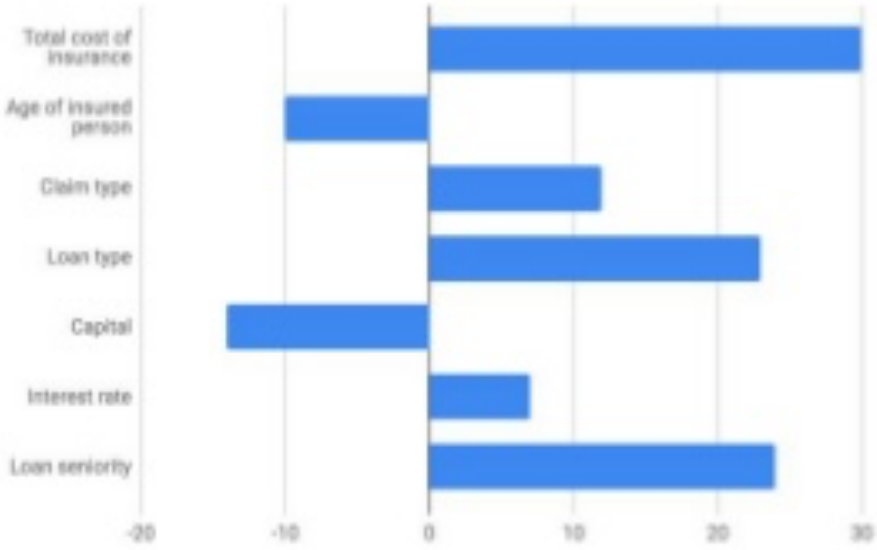
https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance.html

LIME (Local Interpretable Model-Agnostic Explanations)

- Model Agnostic! Approximate a black-box model by a simple linear surrogate model locally
- Learned on perturbations of the original instance in some cases ***faster*** than SHAP
- It doesn't work out-of-the-box on all models.







Disadvantages of LIME

Although LIME has the desirable property of **additivity** (sum of the individual impact is equal to the total impact);
It lacks:

- **Stability**
- **Consistency** (changing the model does not decrease the attribution of a variable if its contribution increases or remains the same)
- **Missingness** (missing variable should have 0 attribution)

Also, “local”/similar is undefined

- All three properties are fulfilled by SHAP

SHAP - SHapley Additive exPlanations

- Explains individual predictions
- SHAPley values can also give ”global” feature importance
- Good, fast, pretty, easy implementations with TreeSHAP – Catboost, XGBoost, Random Forest etc’!
  - Supports other models (deep learning) with KernelSHAP
- Shapley values satisfy desirable properties: local accuracy, consistency, missingness, efficiency
- Based on computational game theory – formal guarantees of correctness!

My go-to method!

https://github.com/slundberg/shap
Lundberg, Scott, and Lee. "A unified approach to interpreting model predictions."

1. Local accuracy

~ your explanation actually matches what happens

2. Missingness

~ things that are not there have no impact

3. Consistency

~things that matter more in the explanation, also matter more in the real model

(or; the numbers make sense)

SHAP

Shapley values satisfy desirable properties:

- **Consistency** (changing the model does not decrease the attribution of a variable if its contribution increases or remains the same)
- **Missingness** (missing variables should have 0 impact)
- **Local accuracy**: The explanation model matches the original model
- **Efficiency:** fairly distributes the gain of each feature.

SHAP - Big idea: Additive, “marginalized” importance

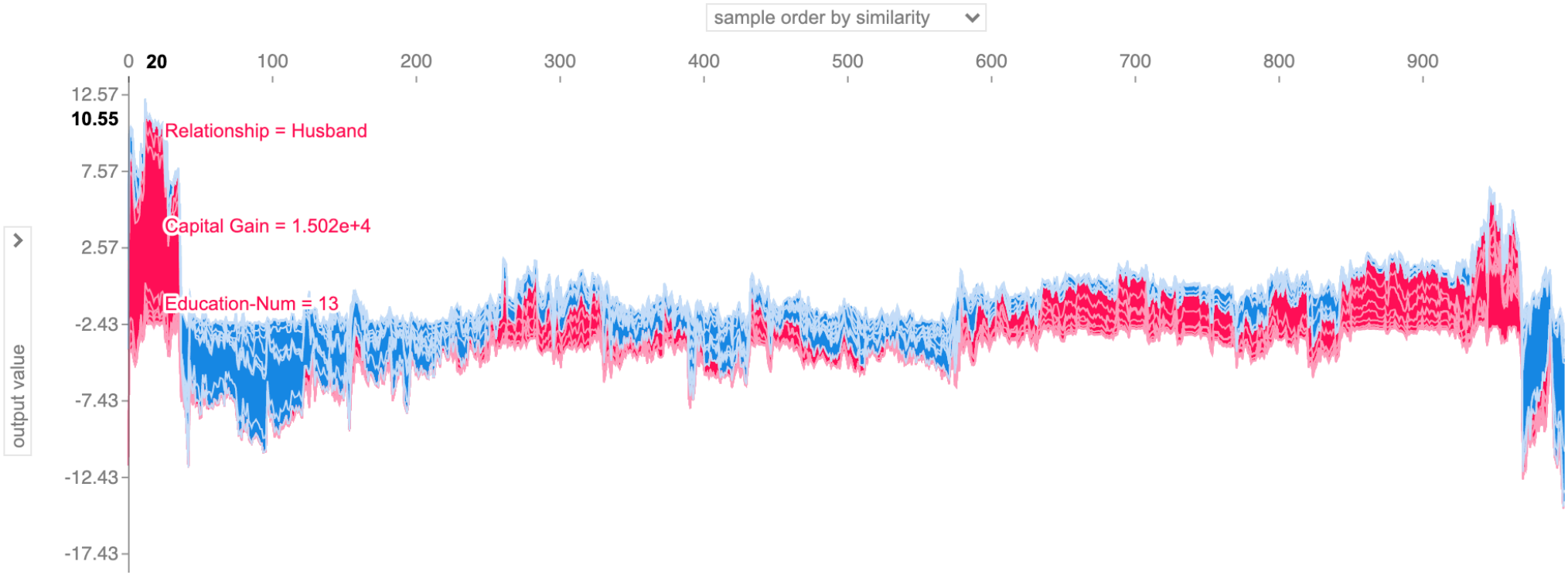
- What is a given feature’s contribution to the output, given all other features?
- “SHAP (SHapley Additive exPlanation) values attribute to each feature the change in the expected model prediction when conditioning on that feature“

SHAP: **Visualize a single prediction**

```
model = lgb.train(X,y)
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X)
shap.force_plot(explainer.expected_value[1], shap_values[1][0,:], X.iloc[0,:])
```

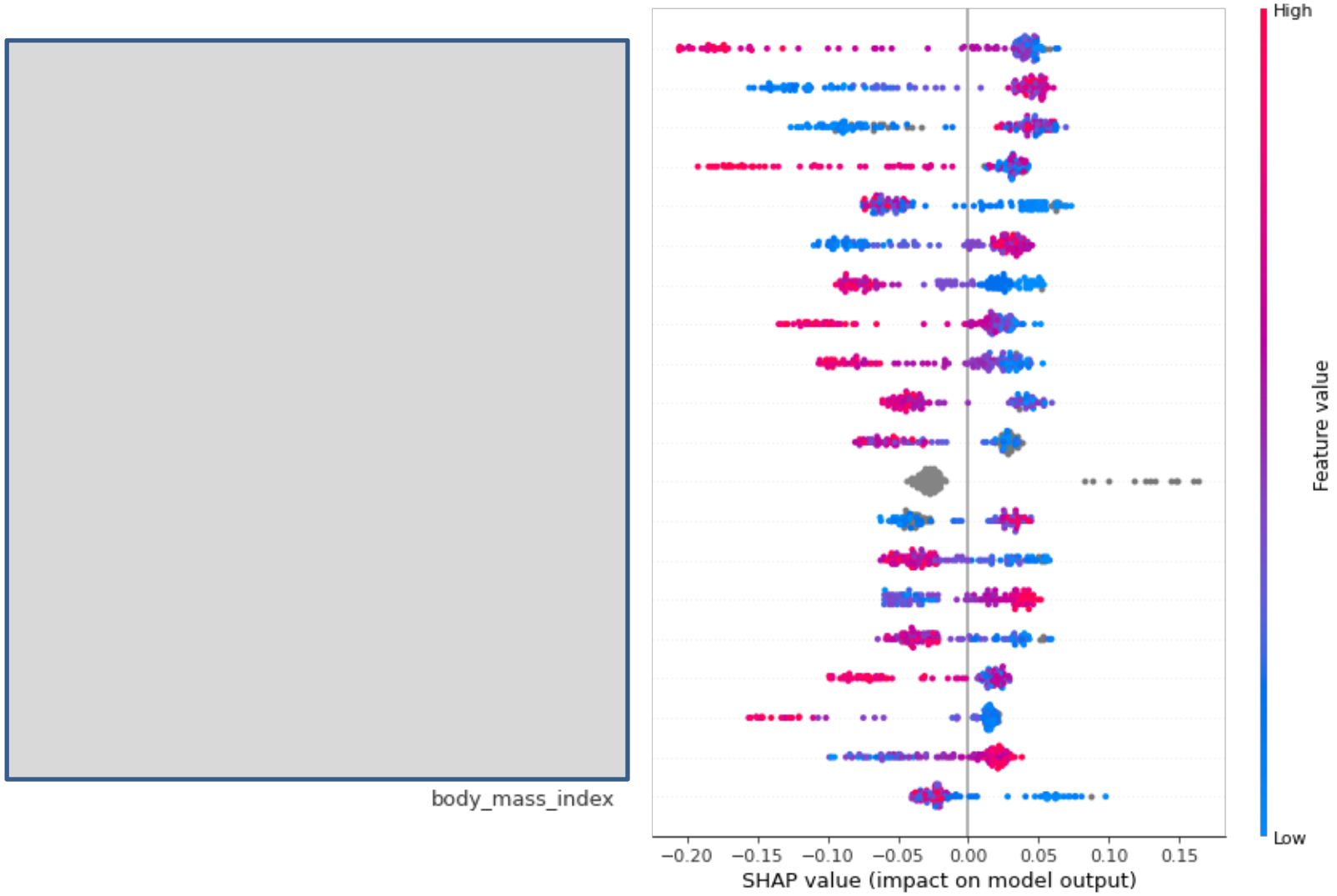
SHAP: **Visualize many predictions**

shap.force_plot(explainer.expected_value[1], shap_values[1][:1000,:], X_display.iloc[:1000,:])



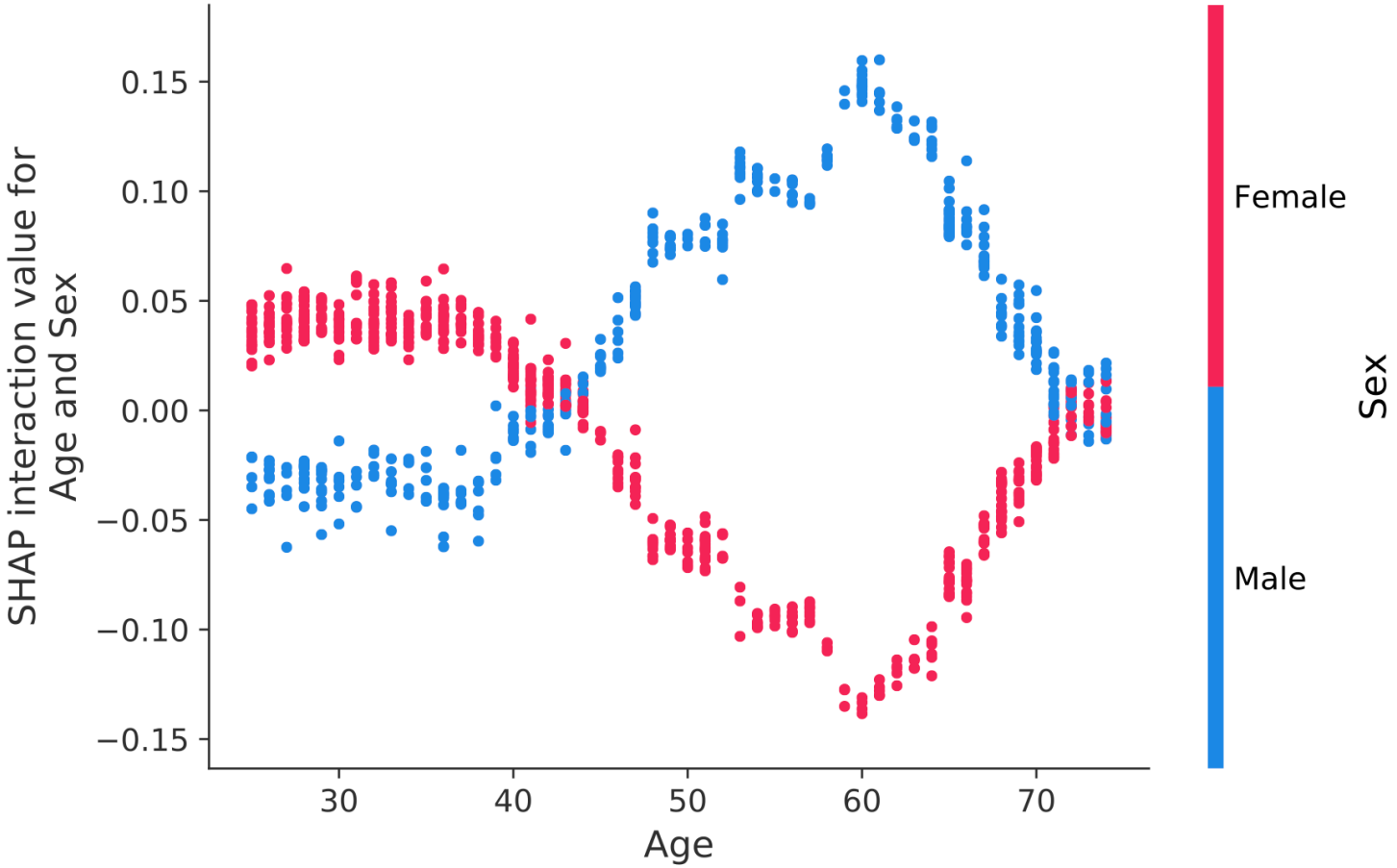
SHAP – Global summary

Predicting Spine fusion surgery complications




```
model = CatBoostClassifier).fit(X,y)
shap_values_ks = model.get_feature_importance(X,y,type="ShapValues")
shap.summary_plot(shap_values_ks [:, :-1], X)
```

SHAP: Feature Interactions



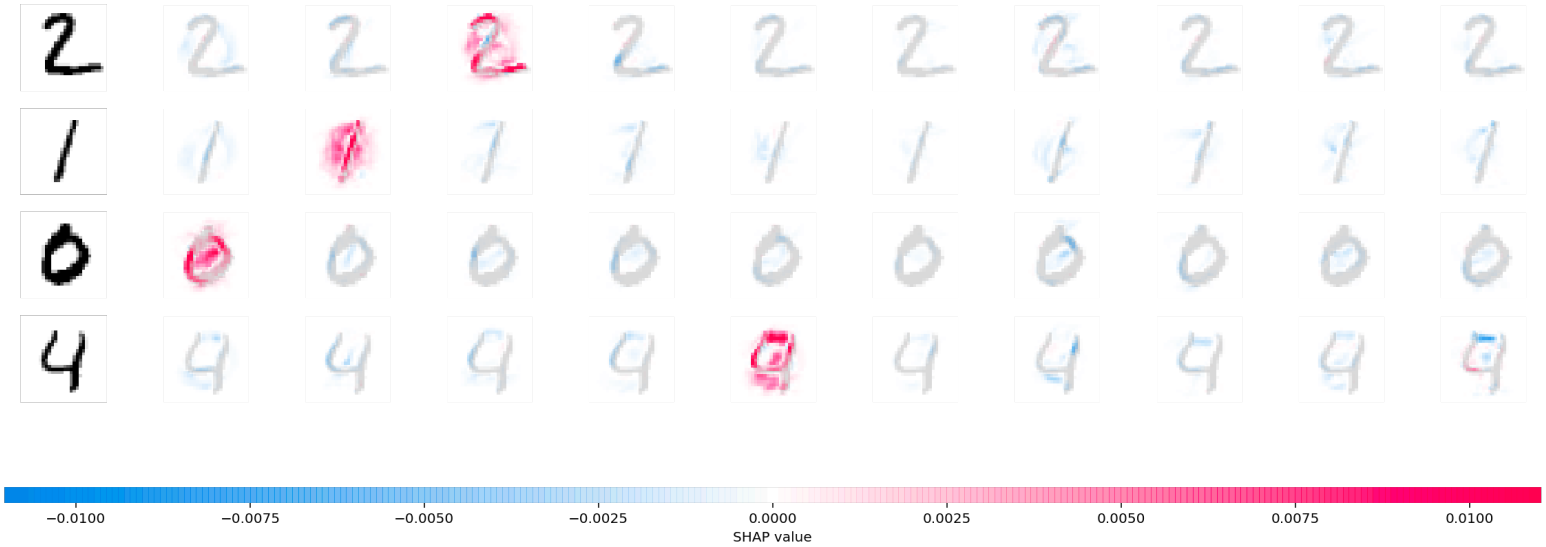
NHANES dataset: Target (Death) = 1:

- Increased risk of death peaks for men at age 60

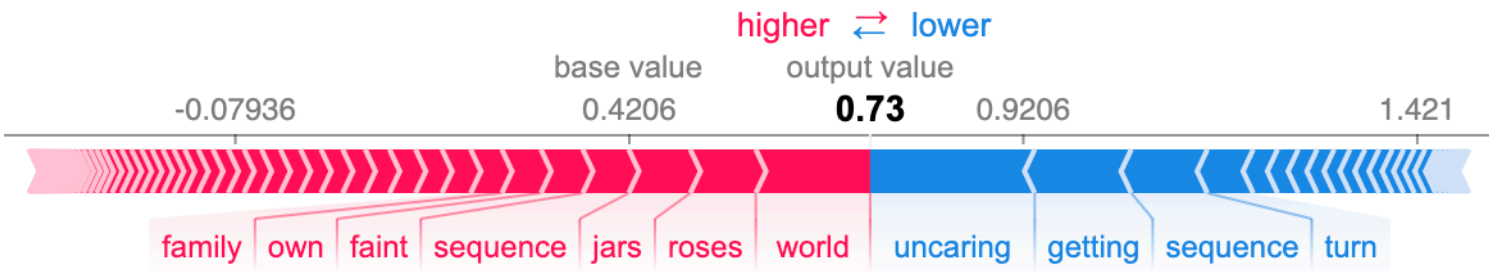
https://github.com/slundberg/shap#shap-interaction-values

SHAP & DL - DeepExplainer

MNIST (Image classification with a CNN):



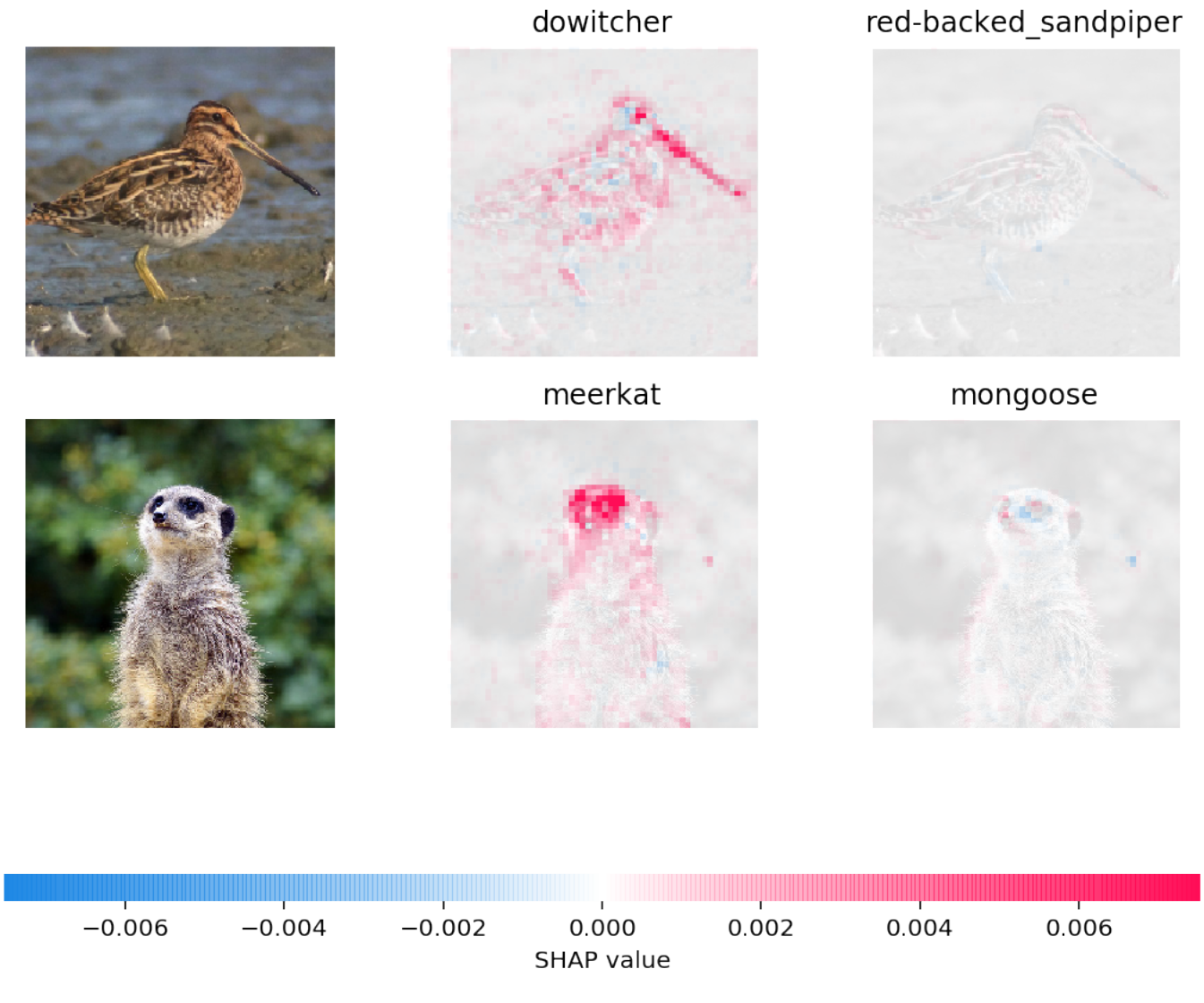
NLP: text sentiment classification with LSTM – explain a single prediction:



See also: DeepLIFT

https://github.com/slundberg/shap#deep-learning-example-with-deepexplainer-tensorflowkeras-models
https://slundberg.github.io/shap/notebooks/deep_explainer/Keras%20LSTM%20for%20IMDB%20Sentiment%20Classification.html

SHAP & DL - GradientExplainer



Global Surrogates/"Student" models

Train a simpler, interpretable model on the target, or predictions of complicated model.

-> Explain outputs of the "simple" model!

- e.g. fit a linear model (LogReg, GAM) or decision tree on the outputs of XGBoost, or a deep network

- I don't like – better to use a interpretable model in the first case.
- No guarantees that it will behave the same – noisy approximation
- If a simple model can approximate, the simple model can typically learn the original problem!

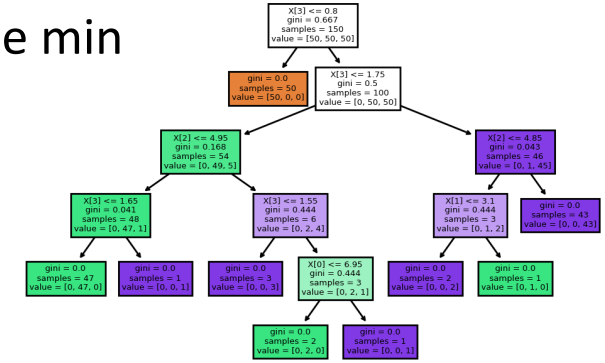
https://christophm.github.io/interpretable-ml-book/global.html

Some great model families

- Linear models
  - Add polynomial features to help understand feature interactions
  - Logistic & linear regression, GLMs, GAMs (non gaussian distributions) etc'
  - Coefficients size – great for regression!

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

- Decision trees, Random forests (RF)
  - GBMs - Gradient boosting trees: Catboost, XGBoost, LightGBM
  - RuleFit (decision trees + sparse linear models)
  - Rule Lists/sets: "If X then Y": can be learned by dedicated model or extracted/pruned from DT/RF models (e.g. take the tree that classifies the most samples)
  - Set hyperparameters – reduce maximum depth, increase min samples per leaf..
  - Analyze single decision trees



Causal analysis

- Use **Causal** approaches – DAGs, bayesian networks, "Causal" model variants (e.g. Causal RF), causal unit effect estimations (CATE)…
- My favorite trick: *Control* for a variable (reweight or split population into subgroups and build separate models), and then get feature importance in the "controlled for" data
  - E.g. controlling for # of prior offenses: does race still appear as a predictive feature for criminal recividism

Interpretable features

- Don’t make complicated models, make complicated features!
- Feature engineering
  - “Penalize” overcomplex engineered features
  - Explicitly take things into account, e.g. “is_missing” features instead of just imputation
- Add Feature interactions (”crossings”)
- Time series – add features, try **shapelets, SaX**, visualize
- …

One way to evaluation interpretability…

"You know it when you see it"

Thank you very much!

Questions? Explanations?