Medical Data Privacy and Ethics in the Age of Artificial Intelligence

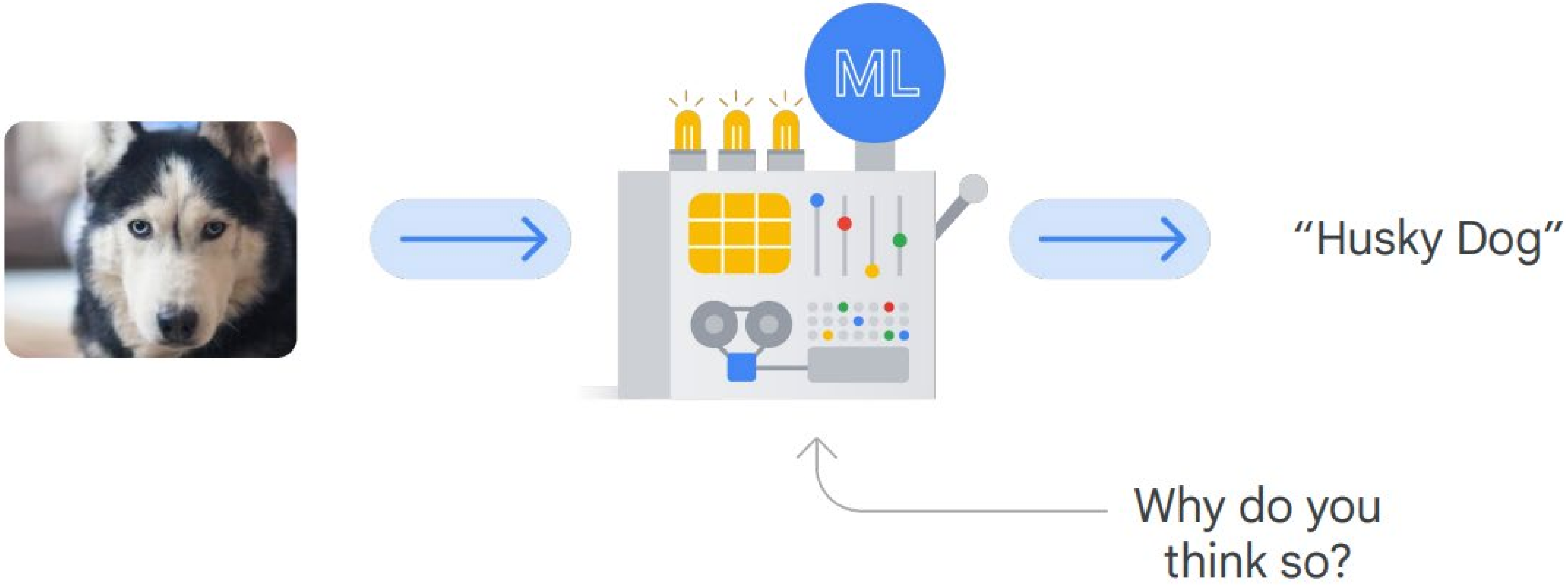
Lecture 7: Transparency and Interpretability Techniques

Zhiyu Wan, PhD (wanzhy@shanghaitech.edu.cn)
Assistant Professor of Biomedical Engineering
ShanghaiTech University

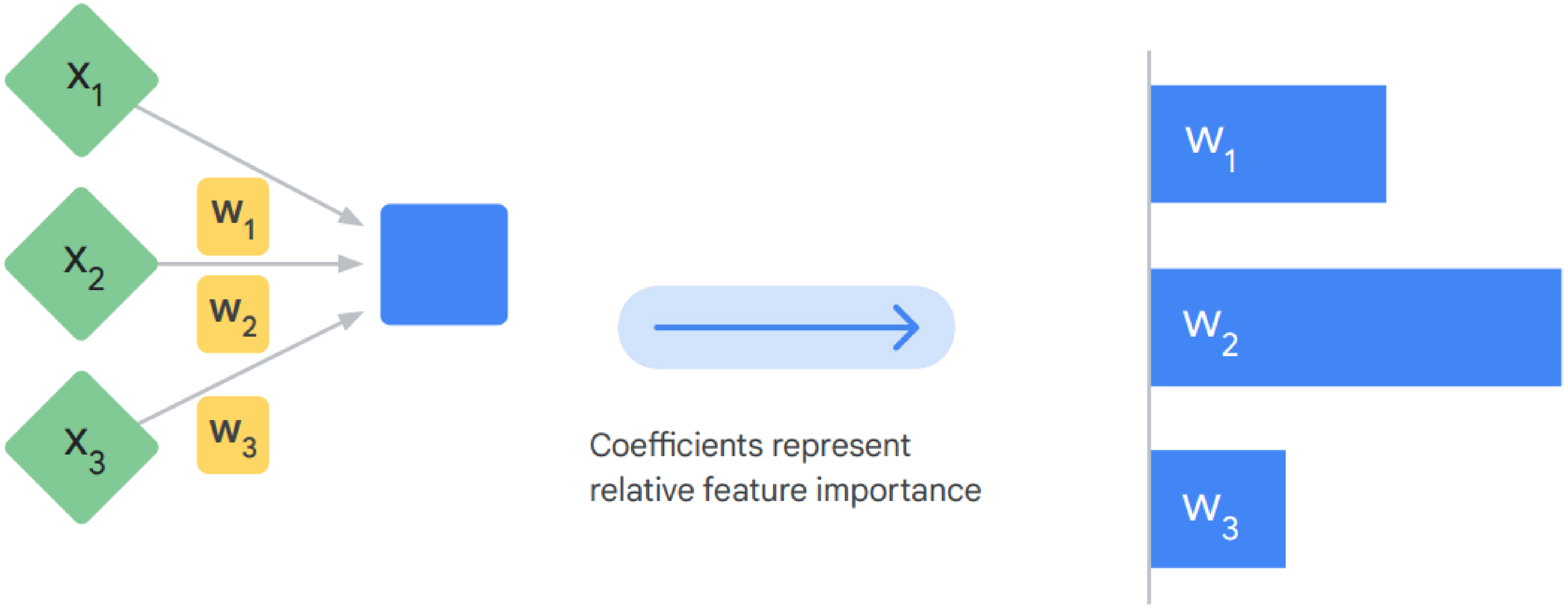
March 19, 2025

Learning Objectives of This Lecture

- Understand 4 interpretability techniques
- Know 2 interpretability tools

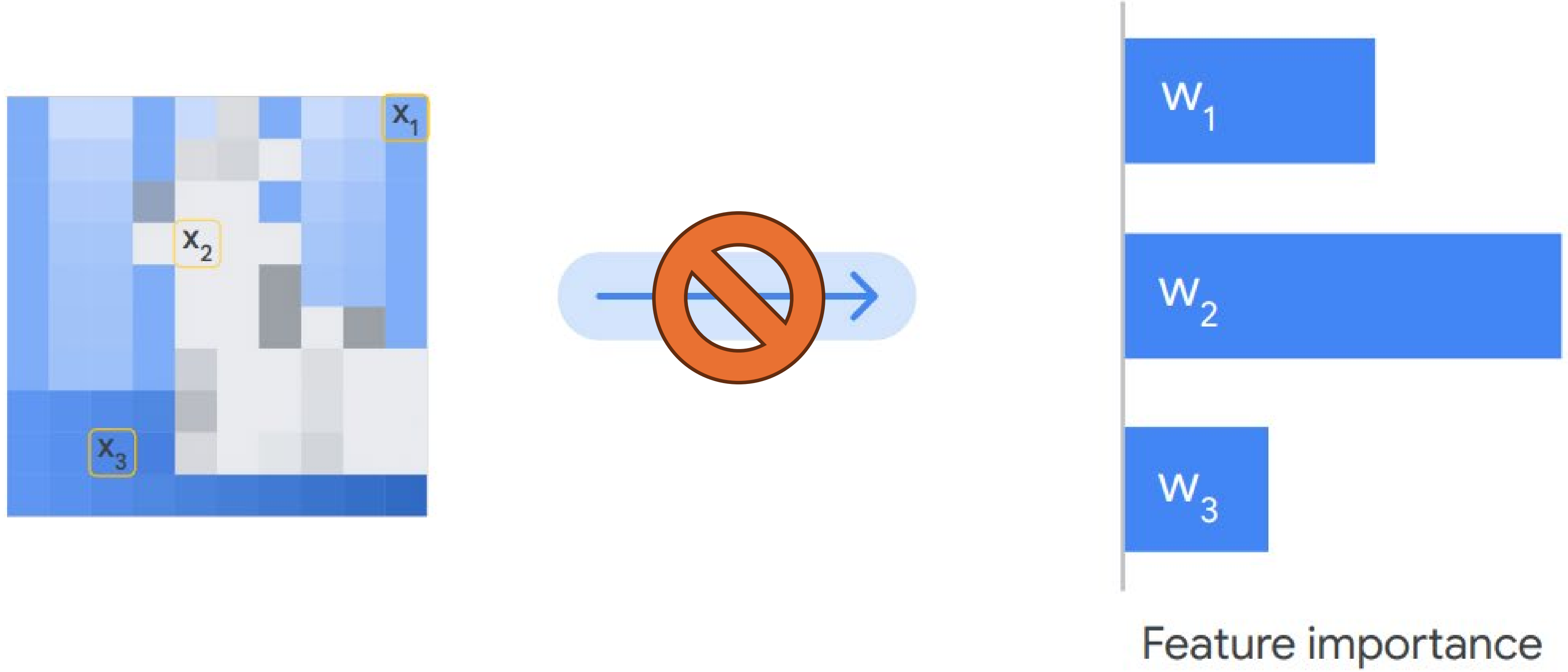


https://www.coursera.org/learn/responsible-ai-for-developers-interpretabilitytransparency



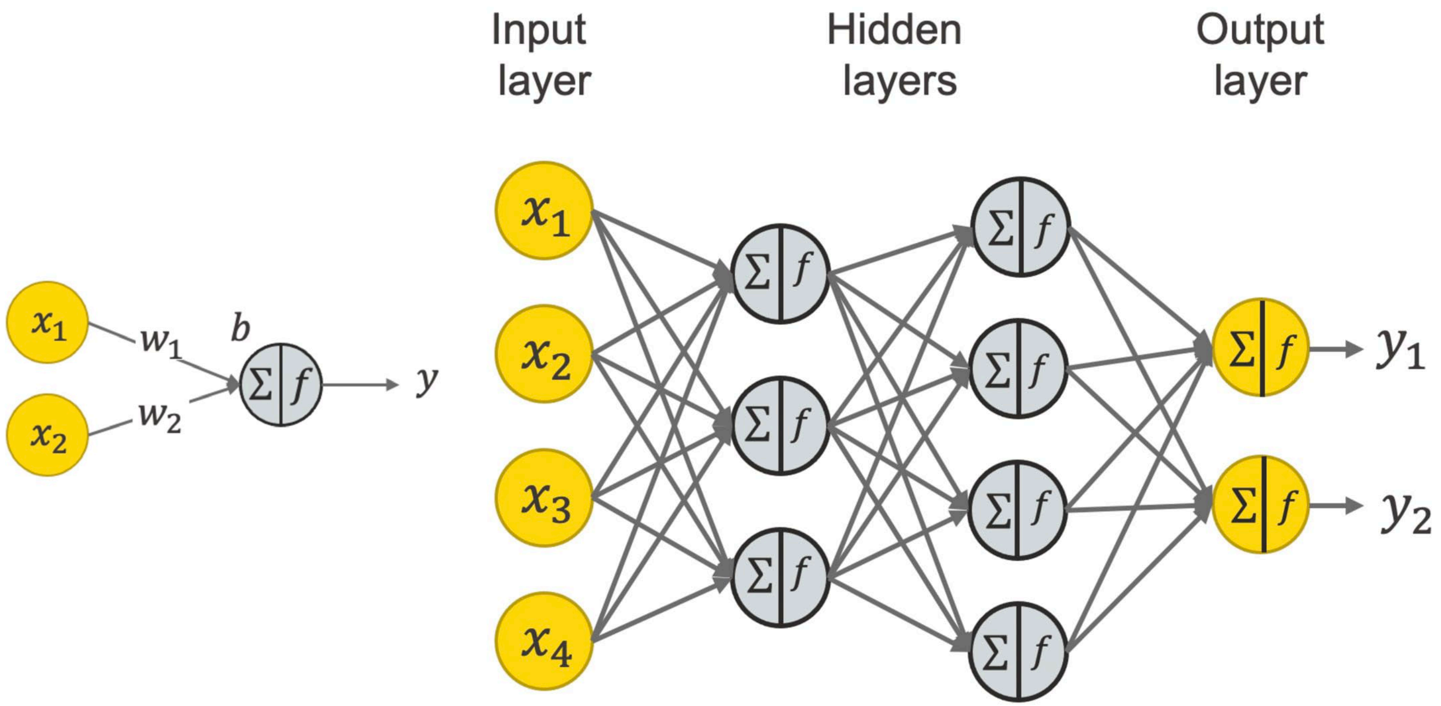
$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 \ldots$$

Feature importance



x_1
x_2
x_3
w_1
w_2
w_3
Feature importance

[Deep] Neural Networks



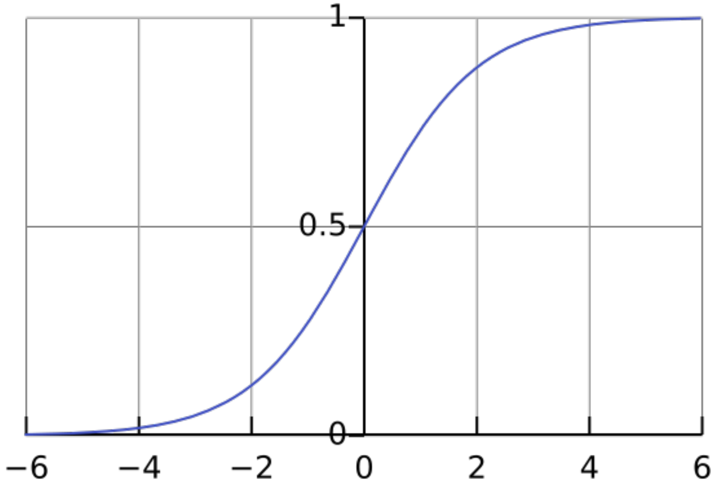


Figure 1. The standard logistic function

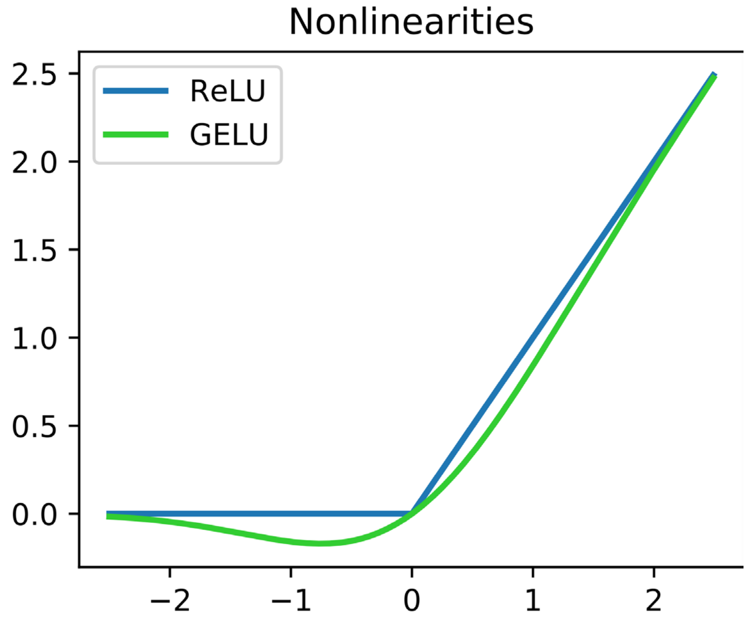
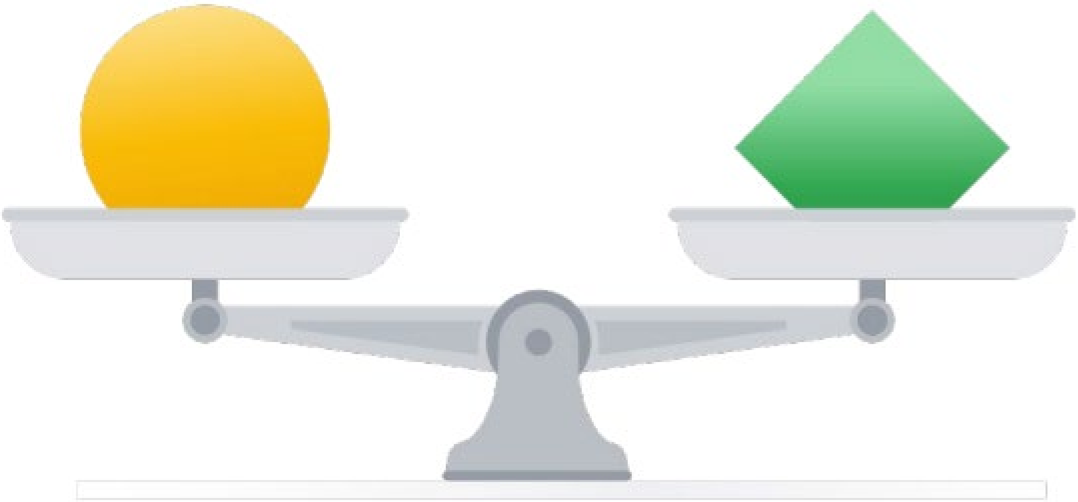


Figure 2. Rectified linear unit (ReLU) and Gaussian Error Linear Unit (GELU)

Complexity – Interpretability Tradeoff

Model Complexity & Performance



Model Interpretability

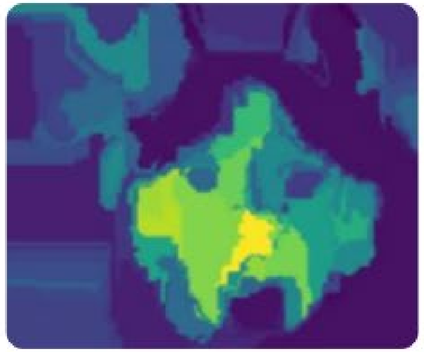
How to explain an ML model

- Intrinsic
  - Linear regression, Decision tree, Bayesian networks
- **Post-hoc (after training)**
  - Local (individual predictions)
    - Model agnostic (**Shapley values**, **LIME**)
    - Model specific (Integrated Gradients, SmoothGrad, XRAI, Grad-CAM)
  - Global (entire model)
    - Model agnostic (**Partial Dependence Plots, Permutation Importance**)
    - Hybrid (**SHAP**, Integrated Gradients)
    - Model specific (Tree Gain-based Importance, TCAV)

Why do you classify this image as a "Husky Dog"?

Which feature do you prioritize most in general?

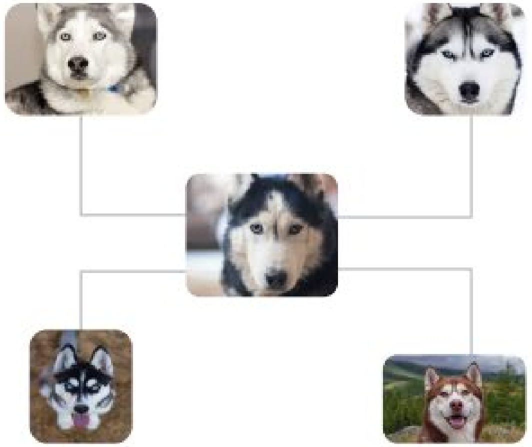
Feature-based Explanation



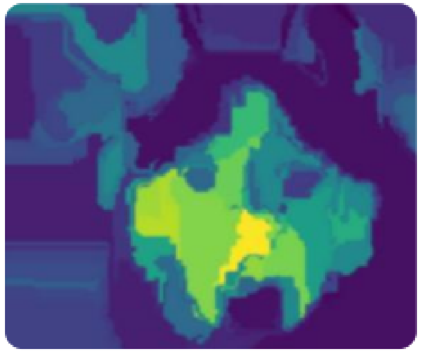
Concept-based Explanation

Erect Ear: 0.47
White Fur: 0.23
Long Nose: 0.64
Well-furred Tail: 0.12
...

Example-based Explanation



Images

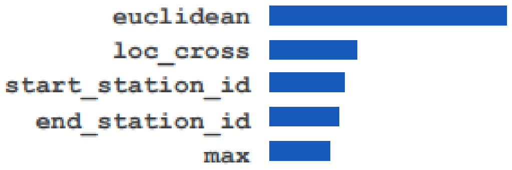


Class: Husky Dog

Image classification

Tabular

Name	Future value	Attrib.
distance	1395.51	-2.44478
temp	16.168	-0.12629
dew_point	7.83396	0.0110318
prcp	0.03	-0.00134132



Prediction: 56.7

Classification / regression

Text

The cake tastes delicious!

Sentiment score: 0.9

Text classification

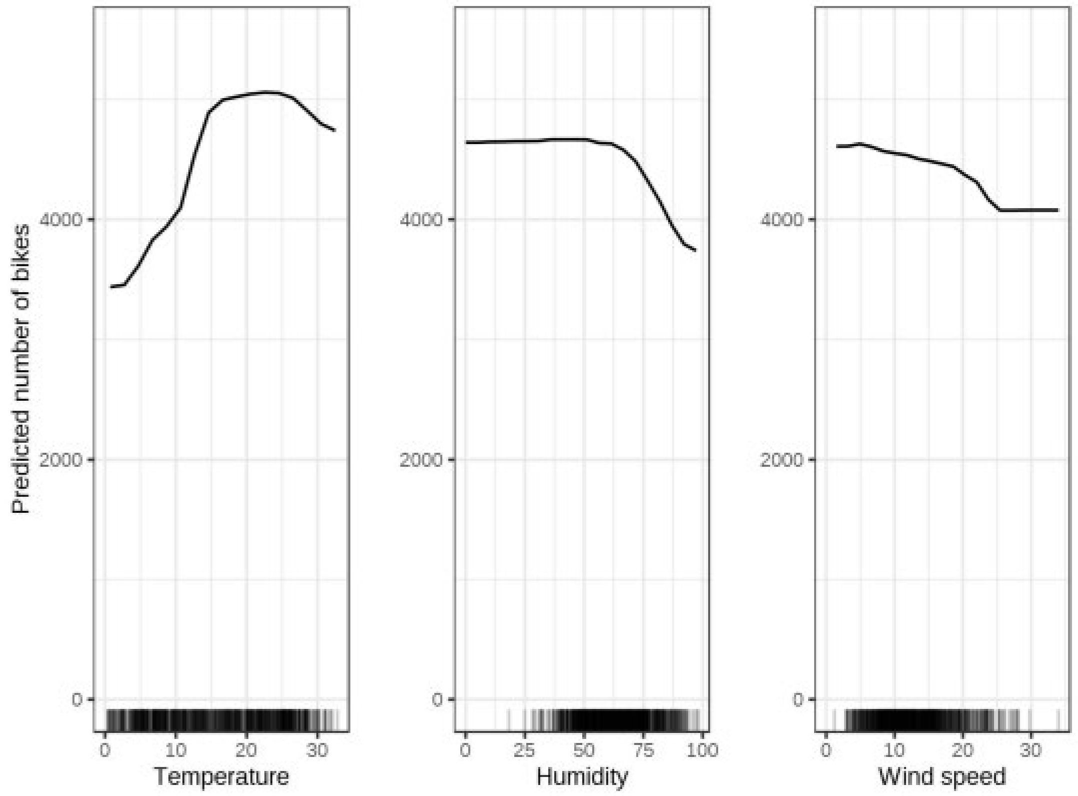
Permutation Feature Importance

- **Post-hoc, global, model agnostic**
  - Randomly Shuffles values of a single feature and observes the resulting change in the model's error rate. The higher the increase in error, the more important the feature is considered to be.
  - It can be intuitive and is easy to implement, but sometimes it can be misleading.

Height at age 20 (cm)	Height at age 10 (cm)	…	Socks owned at age 10
182	155	…	20
175	147	…	10
…	…	…	…
156	142	…	8
153	130	…	24

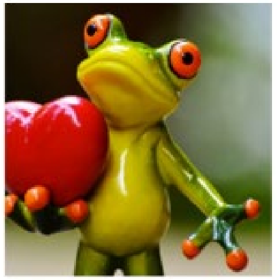
Partial Dependence Plots (PDPs)

- **Post-hoc, global, model agnostic**
  - Used to visualize the relationship between a model's predictions and the values of specific input features
  - Show how the model's predictions change as we vary the values of one input feature while holding all other features constant
  - They can help identify important features, detect nonlinear relationships, and uncover **potential biases** in the model

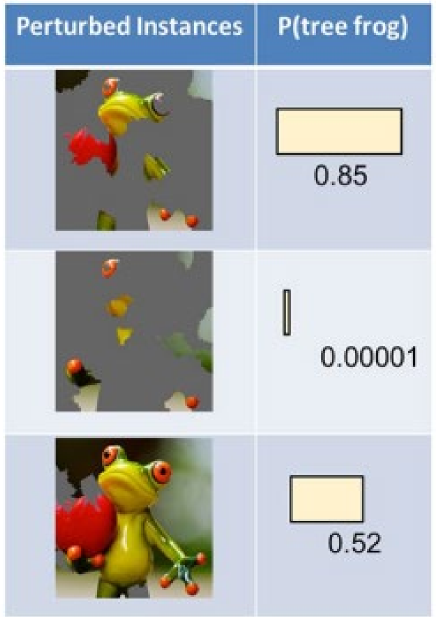


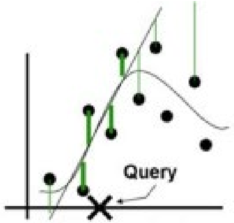
Local Interpretable Model-Agnostic Explanations (LIME)

- **Post-hoc, local, model agnostic**
- Creates an explanation by approximating the underlying model locally, with an interpretable one.
- A linear model or a decision tree is often used.

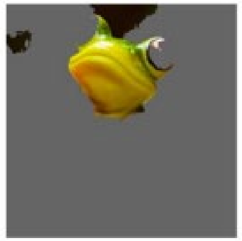


Original Image
P(tree frog)=0.54





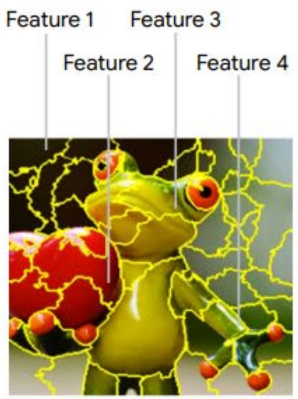
Locally weighted regression



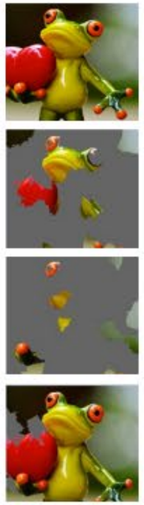
Explanation

Sources: Macro Tulio Ribeiro

1



2



Present -> 1
Absent -> 0

3

4

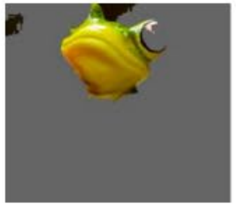
Feature					Label	
Feature1	Feature2	Feature3	Feature4	...	P(tree frog)	weight
1	1	1	1	...	0.54	1
1	0	0	1	...	0.85	0.3
0	0	1	0	...	0.00001	0.01
1	1	1	0	...	0.52	0.89

Proximity from the original data

Interpretable Model
(e.g. Lasso Regression)

5

6



Important Area

Perturbation procedure varies according to the data type

- Images
  - Create pixel groups (super-pixels)
  - Replace super-pixels with gray values
- Text
  - Replace word tokens with a magic token (e.g., UNK)
- Tabular
  - Sample from a normal distribution with mean and standard deviation taken from the feature

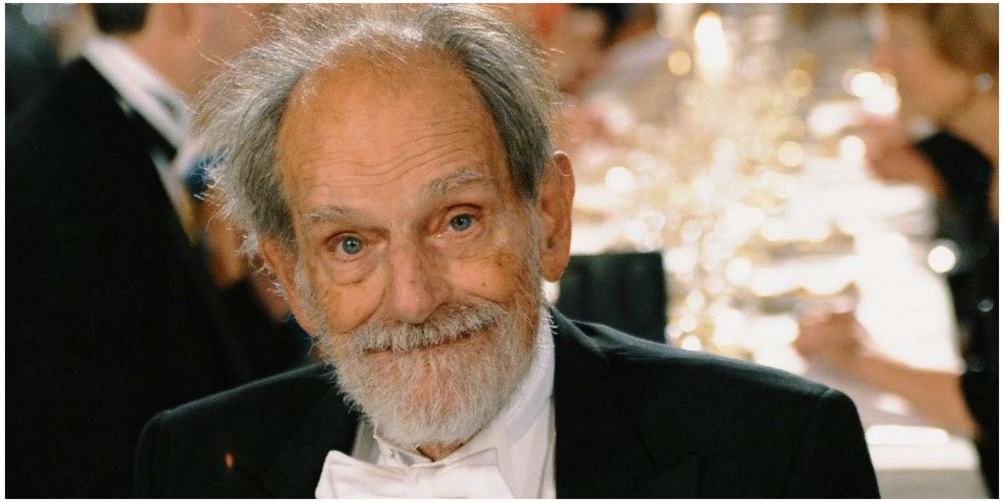
LIME proposes a [UNK] implementation of local [UNK] model based Explanation.

Temp	Humid	Wind
32	52	19

32	74	28

Shapley values

- **Post-hoc, local, model agnostic**
- Come from an area of mathematics known as **cooperative game theory**
- Tries to quantify the contribution of each player in a cooperative situation



Lloyd Shapley won 2012 Nobel Memorial Prize in Economic Sciences

Split based on individual contributions

Ao, Bing

Ne, Zha

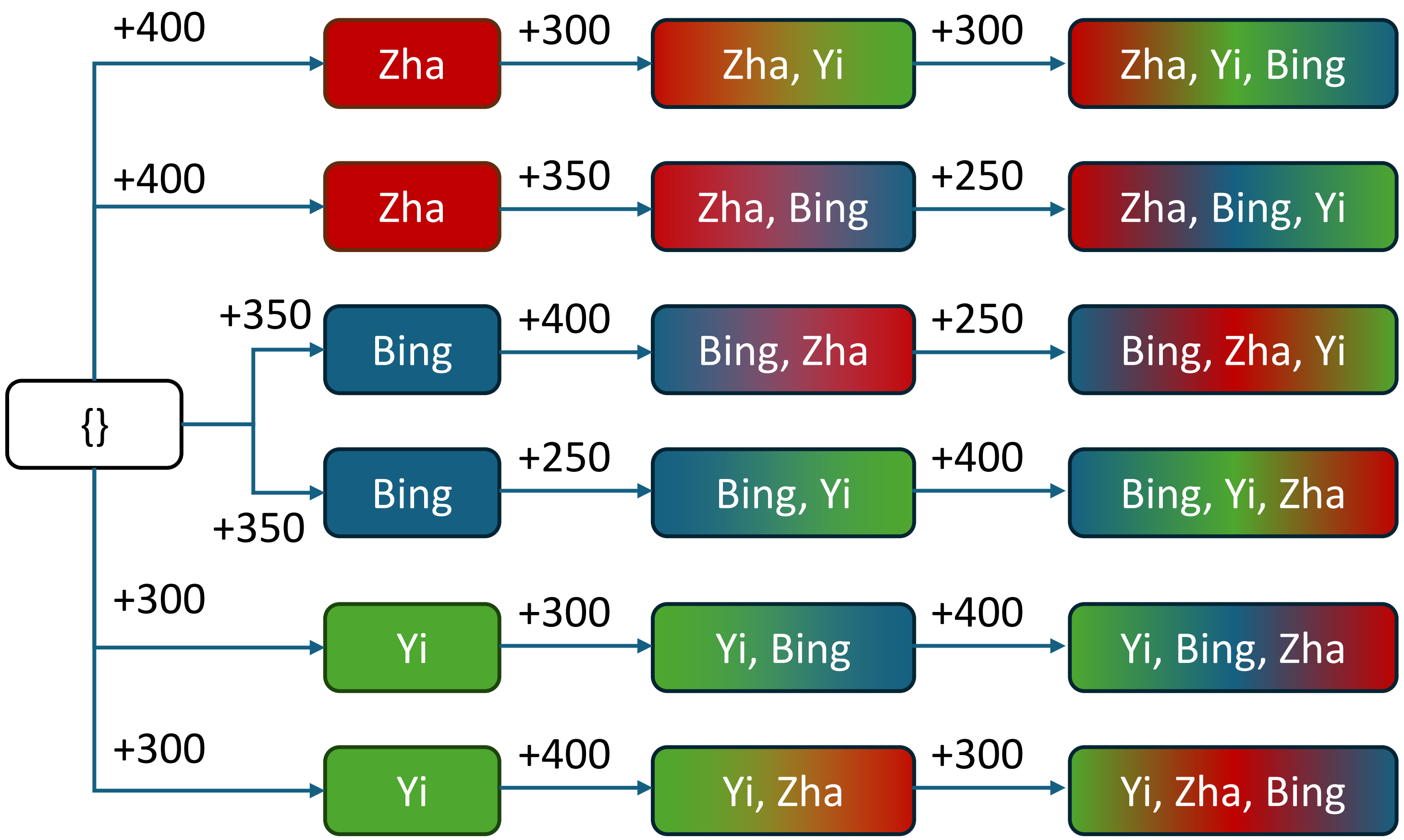
Tai, Yi

1000

Step 1: Do multiple trails in different team set up

	Team	Result
Case 1	{}	0
Case 2	{Zha}	400
Case 3	{Bing}	350
Case 4	{Yi}	300
Case 5	{Zha, Bing}	750
Case 6	{Zha, Yi}	700
Case 7	{Bing, Yi}	600
Case 8	{Zha, Bing, Yi}	1000

Step 2: Arrange trails as paths



Step 3: Calculate average contributions

(Average) Shapley Values

Ne, Zha

Avg(
Zha - {}
Zha - {}
Yi, Zha - Yi
Bing, Zha - Bing
Bing, Yi, Zha - Bing, Yi
Yi, Bing, Zha - Yi, Bing
) = 400

Ao, Bing

Avg(
Bing - {}
Bing - {}
Zha, Bing - Zha
Yi, Bing - Yi
Zha, Yi, Bing - Zha, Yi
Yi, Zha, Bing - Yi, Zha
) = 325

Tai, Yi

Avg(
Yi - {}
Yi - {}
Zha, Yi - Zha
Bing, Yi - Bing
Zha, Bing, Yi - Zha, Bing
Bing, Zha, Yi - Bing, Zha
) = 275

Additivity!

What happens if we have 1000 people?

- Number of cases = 2 to the number of features (8=2^3)
- Approximation algorithms of Shapley values
  - Sampled Shapley
    - Approximate Shapley value by sampling (not calculating all the cases) permutations
  - Kernal SHAP
    - Slightly faster than Sampled Shapley, but it assumes the independence of features.
  - Tree SHAP
    - Faster approximation algorithm, but it can only be applied to Tree-based models. Hence this technique is model-specific

Interpretability tools

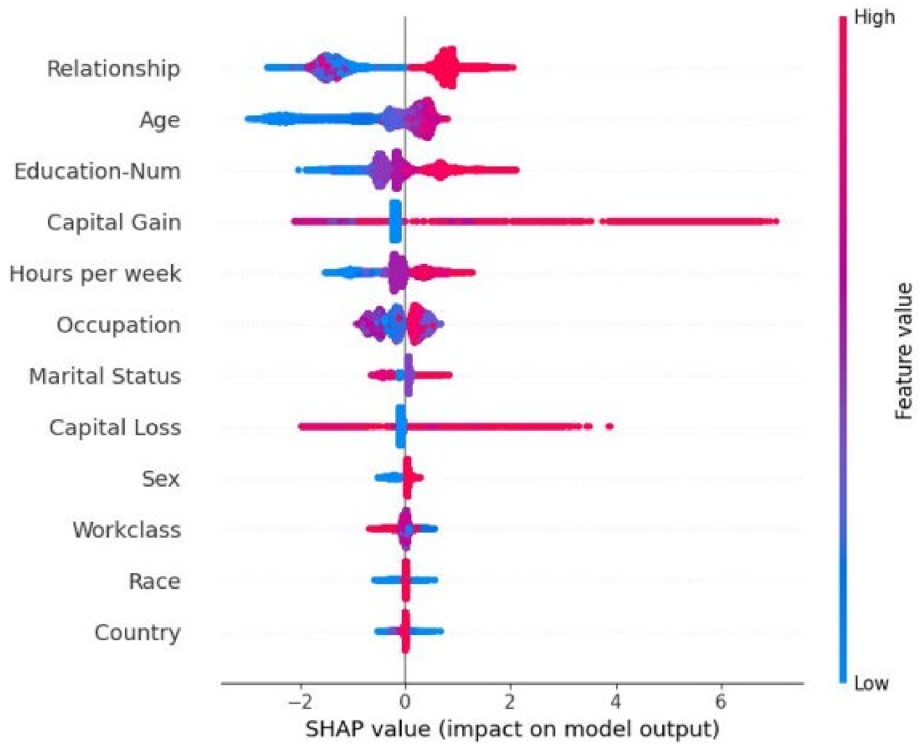
- SHAP Python library
- Learning Interpretability Tool (LIT)

SHAP Python Library

- It provides popular implementations of approximate Shapley values, including Sampled Shapley, Kernel SHAP, Tree SHAP, etc.
- It has limited applications to domains such as images.
- Code Sample:

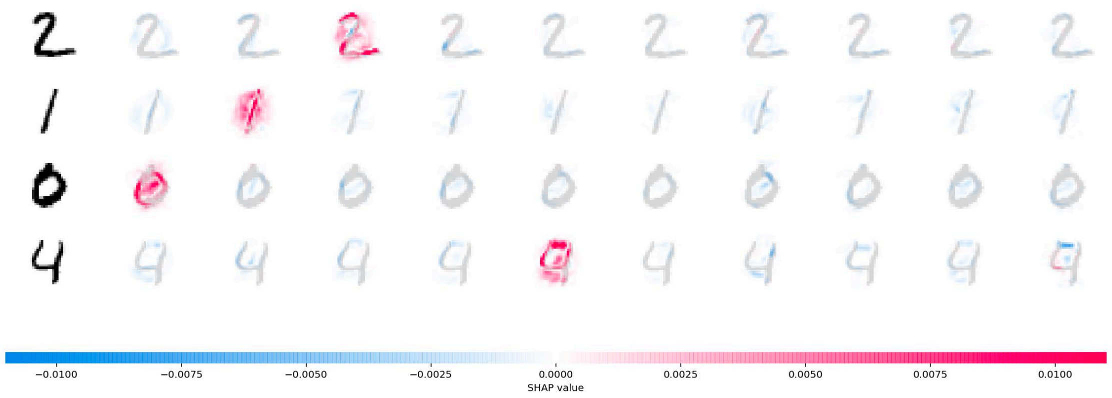
```
import shap
...
explainer =
shap.TreeExplainer(model)
shap_values =
explainer.shap_values(X)
shap.summary_plot(shap_values, X)
```

Visualization Example



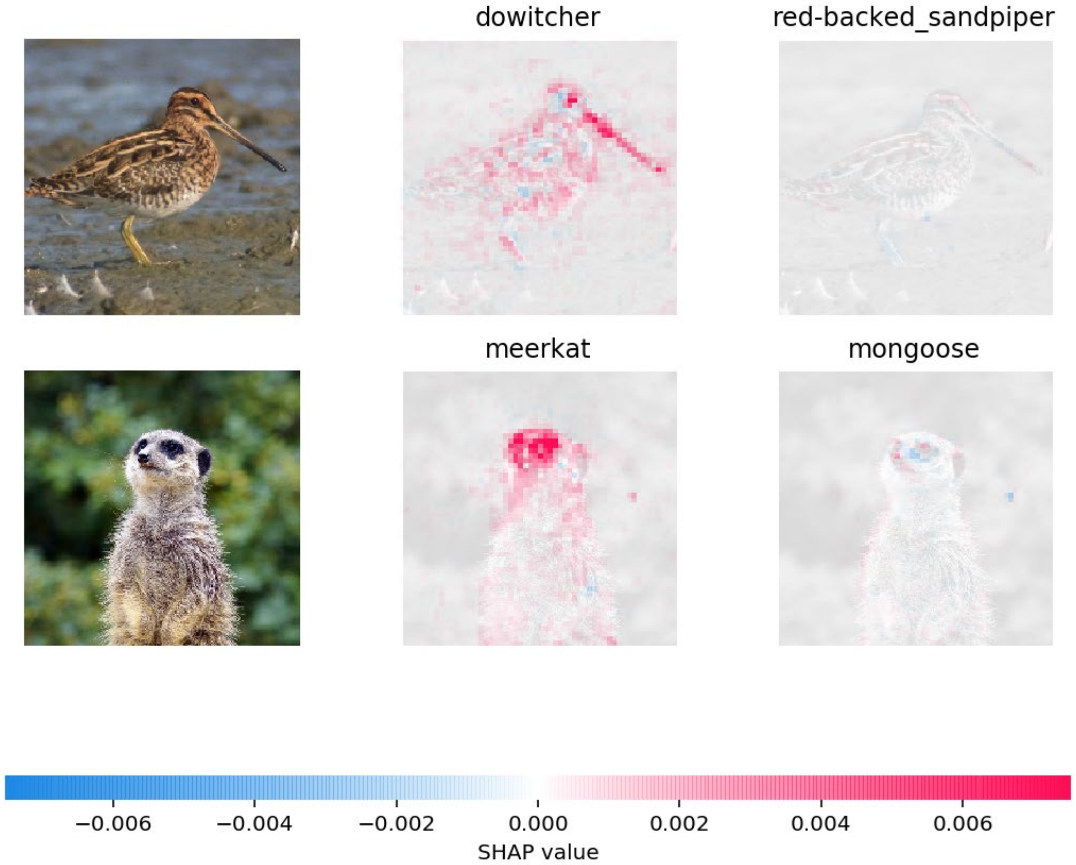
SHAP for images (Examples)

- DeepExplainer



SHAP applied to MNIST

GradientExplainer

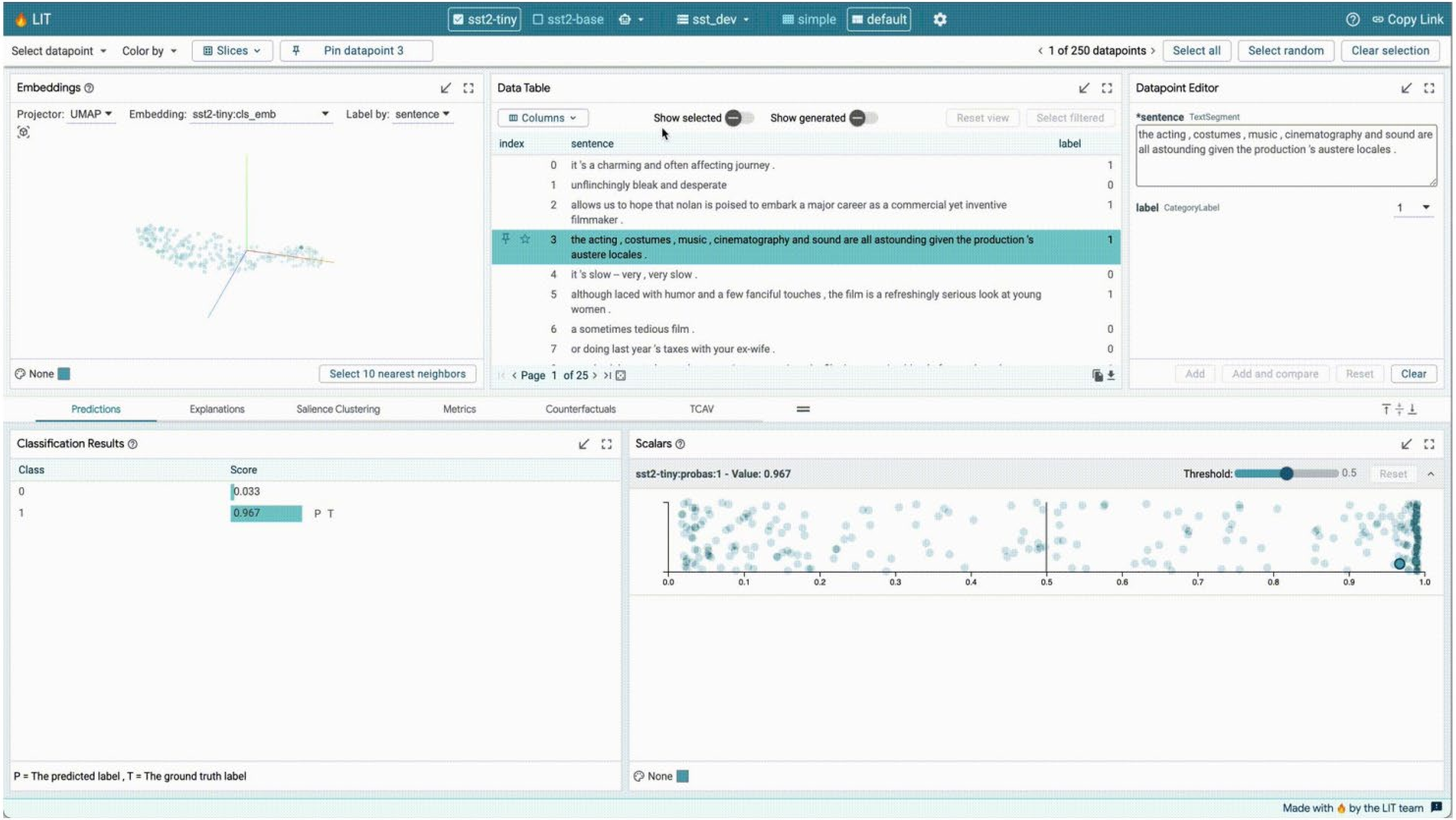


SHAP applied to ImageNet

https://github.com/shap/

Learning Interpretability Tool (LIT)

- It mainly supports Natural Language Processing (NLP) with some preliminary support for tabular and image data.
  - What kind of examples does my model perform poorly on?
  - Why did my model make this prediction? Does the model properly focus on important features, instead of obviously unimportant features like image background?
  - Does my model behave consistently if I change things like textual style, verb tense, or pronoun gender?
  - And does this method relate to counterfactual analysis in AI fairness and bias?



https://pair-code.github.io/lit/
https://github.com/PAIR-code/lit

Take-away messages

- Understood 4 interpretability techniques
  - Permutation Feature Importance
  - Partial Dependence Plots (PDPs)
  - Local Interpretable Model-Agnostic Explanations (LIME)
  - **Shapley Values**
    - SHAP (SHapley Additive exPlanations)
- Knew 2 interpretability tools
  - SHAP Python Library
  - Learning Interpretability Tool (LIT)

Readings for the Next Week

- None
- Optional
  - ❑ None

Feedback Survey

- One thing you learned or felt was valuable from today's class & reading
- Muddiest point: what, if anything, feels unclear, confusing or "muddy"
- https://www.wjx.cn/vm/hX0mIro.aspx