Copyright Notice

DeepLearning.AI

Interpretability

Welcome

DeepLearning.AI

Explainable AI

Explainable AI

Responsible AI

- Development of AI is creating new opportunities to improve lives of people
- Also raises new questions about the best way to build the following into AI systems:

Fairness	**Explainability**	Privacy	Security
• Ensure working towards systems that are fair and inclusive to all users. • Explainability helps ensure fairness.	• Understanding how and why ML models make certain predictions. • Explainability helps ensure fairness.	Training models using sensitive data needs privacy preserving safeguards.	Identifying potential threats can help keep AI systems safe and secure.

Explainable Artificial Intelligence (XAI)

The field of XAI allow ML system to be more transparent, providing explanations of their decisions in some level of detail.

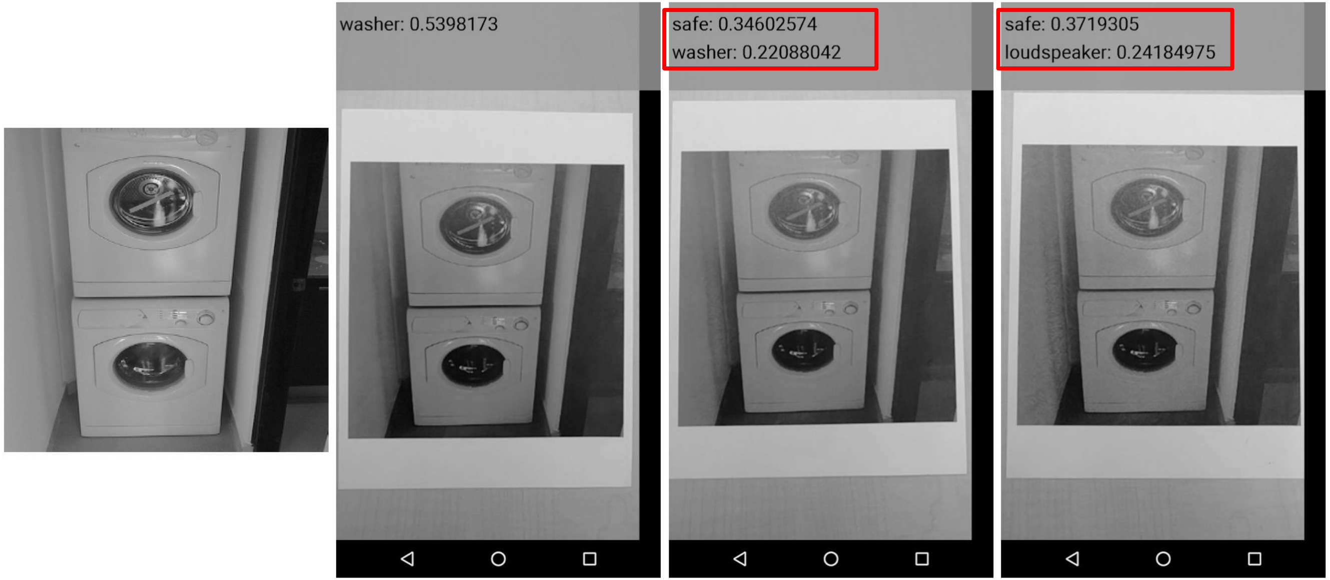
These explanations are important:

- To ensure algorithmic fairness.
- Identify potential bias and problems in training data.
- To ensure algorithms/models work as expected.

Need for Explainability in AI

1. Models with high sensitivity, including natural language networks, can generate wildly wrong results
2. Attacks
3. Fairness
4. Reputation and Branding
5. Legal and regulatory concerns
6. Customers and other stakeholders may question or challenge model decisions

Deep Neural Networks (DNNs) can be fooled



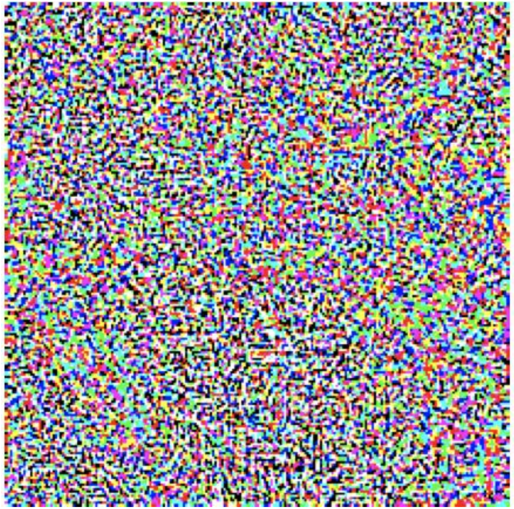
(a) Image from dataset (b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

DNNs can be fooled into misclassifying inputs with no resemblance to the true category.

Deep Neural Networks (DNNs) can be fooled

"Panda"
57.7 % confidence

+ε



"Nematode"
8.2 % confidence

"Gibbon"
99.3 % confidence

DeepLearning.AI

Interpretability

Model Interpretation Methods

What is interpretability?

"(Models) are interpretable if their operations can be understood by a human, either through introspection or through a produced explanation."

"Explanation and justification in machine learning: A survey"

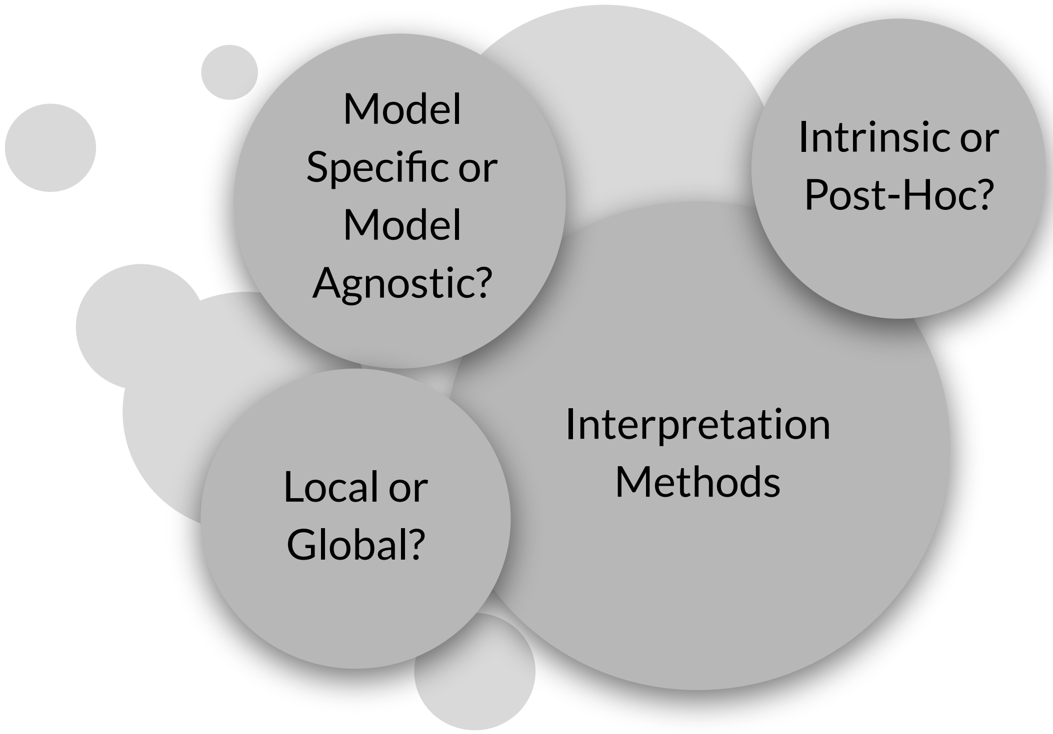
- O. Biran, C. Cotton

What are the requirements?

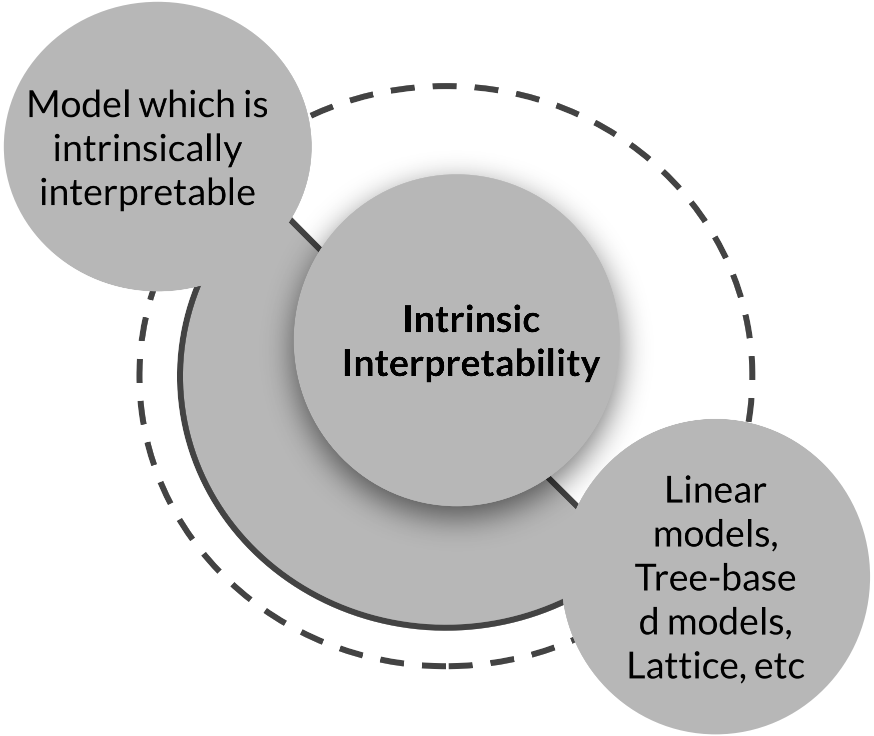
You should be able to query the model to understand:

- Why did the model behave in a certain way?
- How can we trust the predictions made by the model?
- What information can model provide to avoid prediction errors?

Categorizing Model Interpretation Methods



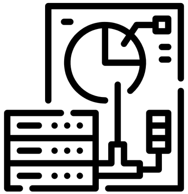
Intrinsic or Post-Hoc?



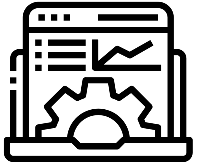
Intrinsic or Post-Hoc?

- Post-hoc methods treat models as black boxes
- Agnostic to model architecture
- Extracts relationships between features and model predictions, agnostic of model architecture
- Applied after training

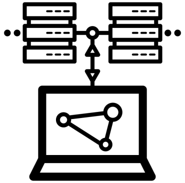
Types of results produced by Interpretation Methods



Feature Summary Statistics



Feature Summary Visualization



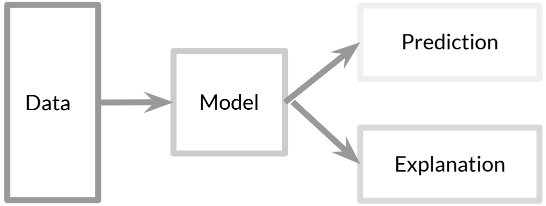
Model Internals

Data point

Model Specific or Model Agnostic

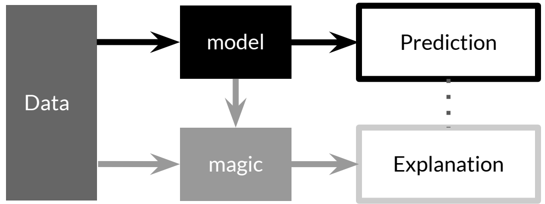
Model Specific

- These tools are limited to specific model classes
- Example: Interpretation of regression weights in linear models
- Intrinsically interpretable model techniques are model specific
- Tools designed for particular model architectures

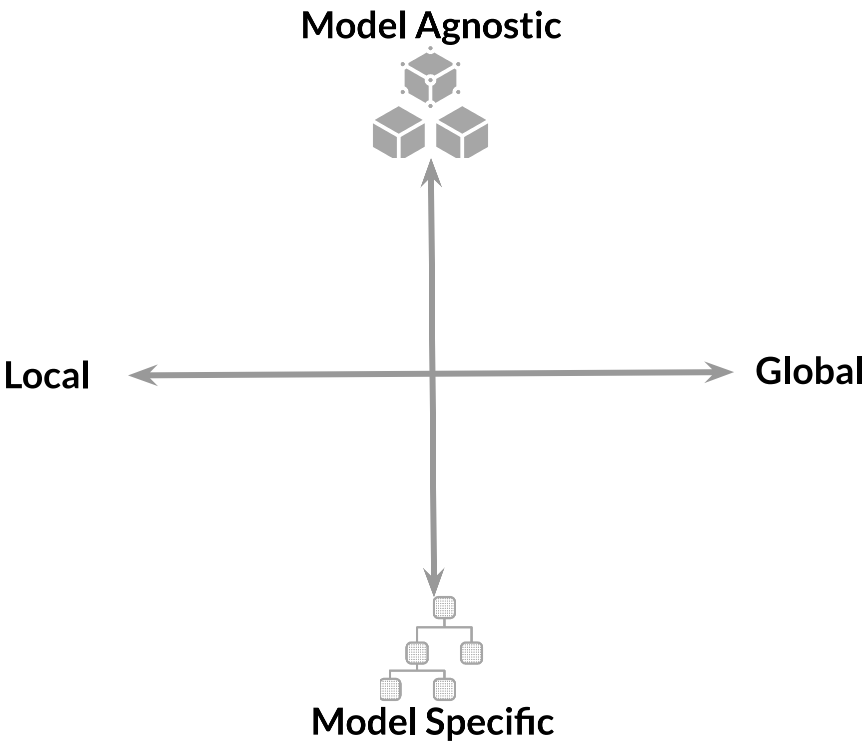


Model Agnostic

- Applied to any model after it is trained
- Do not have access to the internals of the model
- Work by analyzing feature input and output pairs

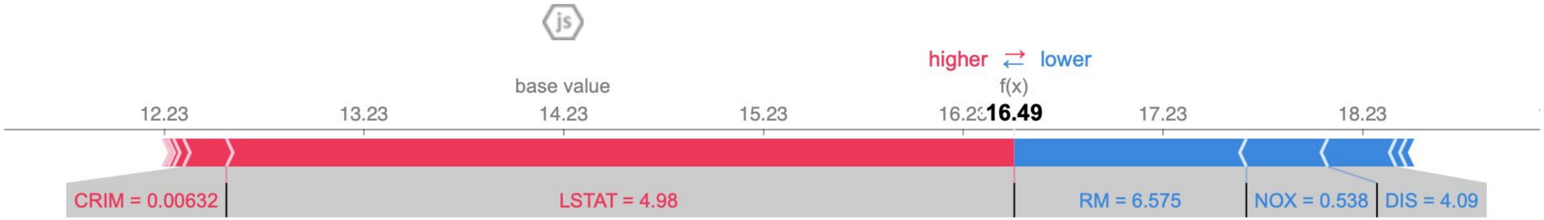


Interpretability of ML Models



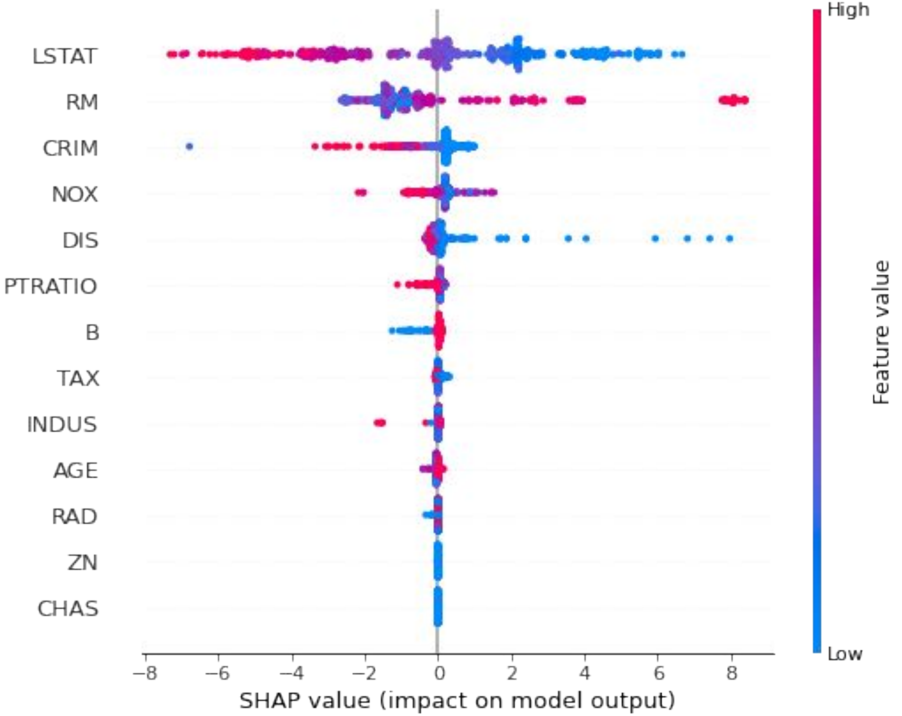
Local or Global?

- Local: interpretation method explains an individual prediction.
- Feature attribution is identification of relevant features as an explanation for a model.



Local or Global?

- Global: interpretation method explains entire model behaviour
- Feature attribution summary for the entire test data set



DeepLearning.AI

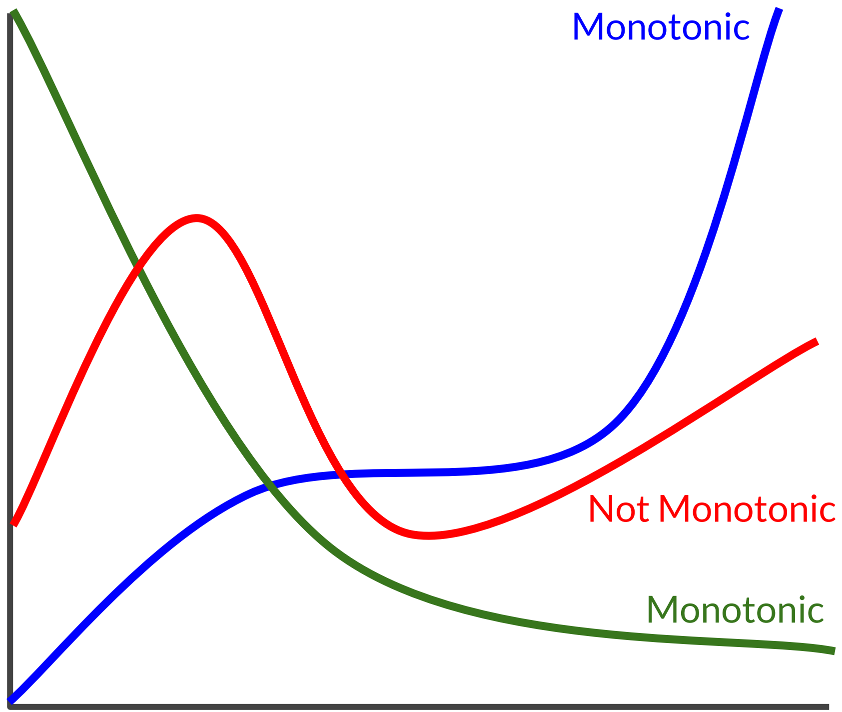
Interpretability

Intrinsically Interpretable Models

Intrinsically Interpretable Models

- How the model works is self evident
- Many classic models are highly interpretable
- Neural networks look like “black boxes”
- Newer architectures focus on designing for interpretability

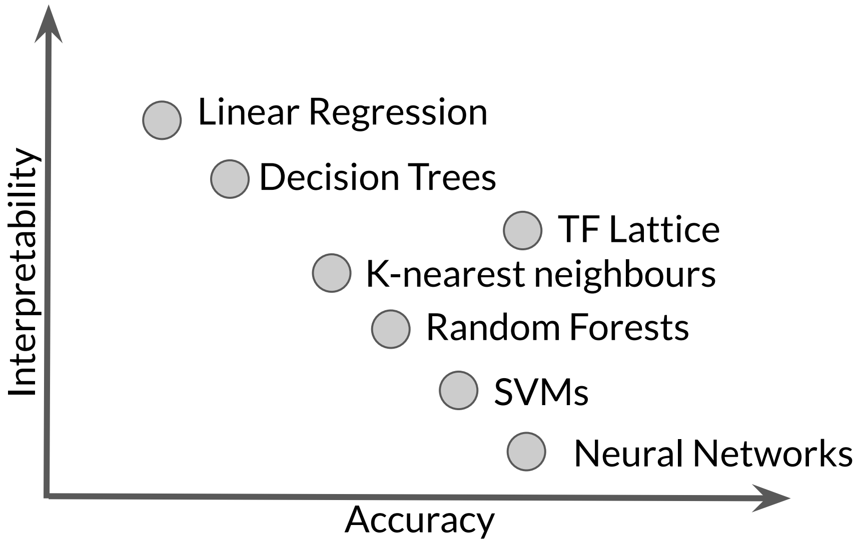
Monotonicity improves interpretability



Interpretable Models

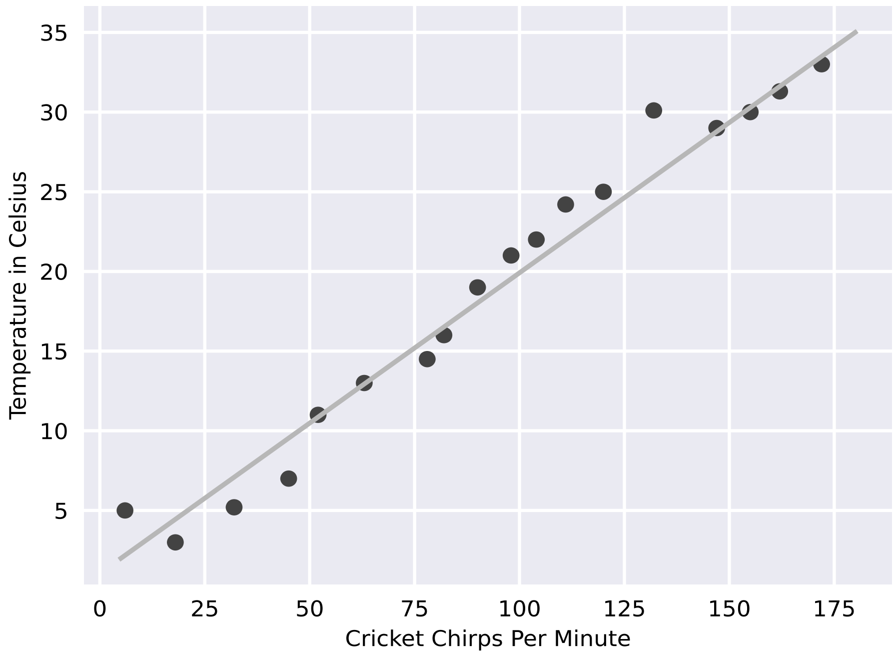
Algorithm	Linear	Monotonic	Feature Interaction	Task
Linear regression	Yes	Yes	No	regr
Logistic regression	No	Yes	No	class
Decision trees	No	Some	Yes	class, regr
RuleFit	Yes*	No	Yes	class, regr
K-nearest neighbors	No	No	No	class, regr
TF Lattice	Yes*	Yes	Yes	class, regr

Model Architecture Influence on Interpretability



Interpretability vs Accuracy Trade off

Classics: Linear Regression



Interpretation from Weights

Linear models have easy to understand interpretation from weights

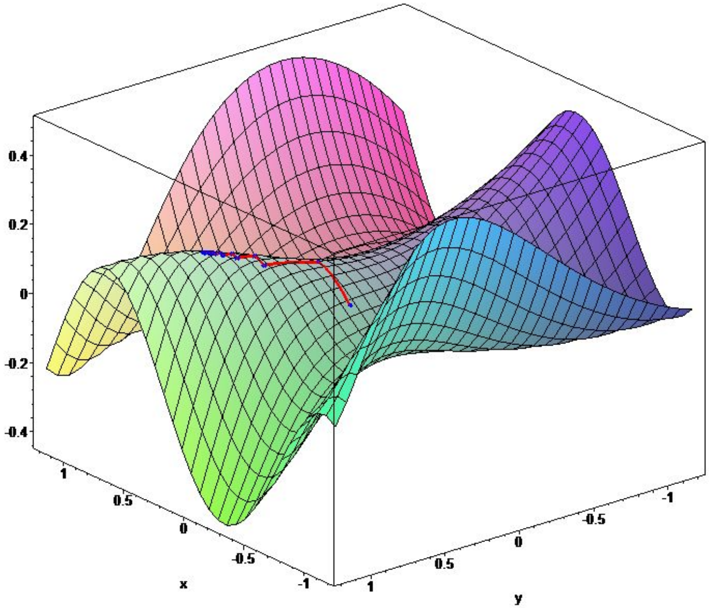
- Numerical features: Increase of one unit in a feature increases prediction by the value of corresponding weight.
- Binary features: Changing between 0 or 1 category changes the prediction by value of the feature's weight.
- Categorical features: one hot encoding affects only one weight.

Feature Importance

- Relevance of a given feature to generate model results
- Calculation is model dependent
- Example: linear regression model, t-statistic

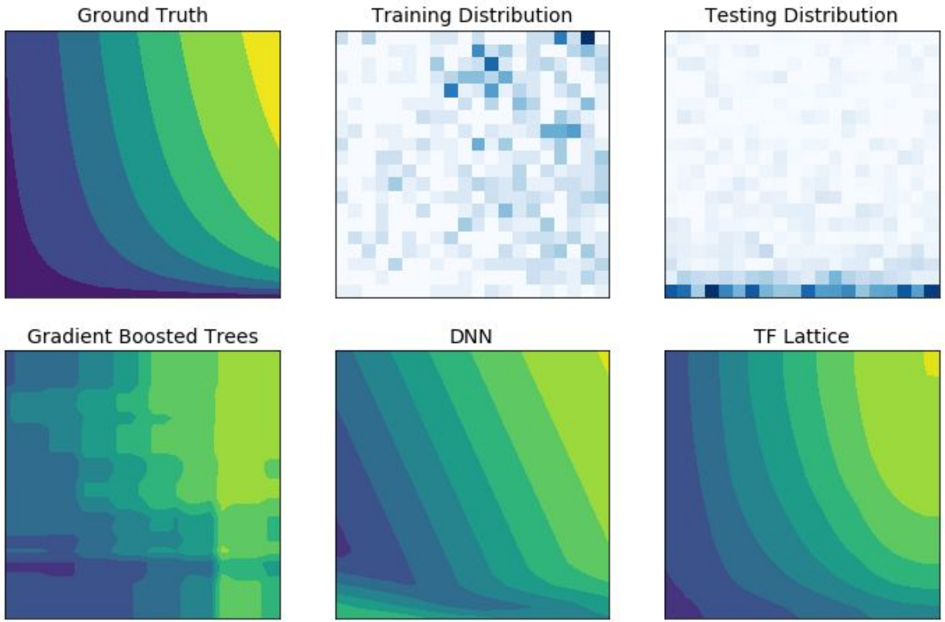
More advanced models: TensorFlow Lattice

- Overlaps a grid onto the feature space and learns values for the output at the vertices of the grid
- Linearly interpolates from the lattice values surrounding a point



More advanced models: TensorFlow Lattice

- Enables you to **inject domain knowledge** into the learning process through common-sense or policy-driven shape constraints
- Set constraints such as monotonicity, convexity, and how features interact



TensorFlow Lattice: Accuracy

Accuracy

- TensorFlow Lattice achieves accuracies comparable to neural networks
- TensorFlow Lattice provides greater interpretability

TensorFlow Lattice: Issues

Dimensionality

- The number of parameters of a lattice layer **increases exponentially** with the number of input features
- Very Rough Rule: Less than 20 features ok without ensembling

Understanding Model Predictions

Model Agnostic Methods

DeepLearning.AI

Model Agnostic Methods

These methods separate explanations from the machine learning model.

Desired characteristics:

- Model flexibility
- Explanation flexibility
- Representation flexibility

Model Agnostic Methods

Partial Dependence Plots

Individual Conditional Expectation

Accumulated Local Effects

Permutation Feature Importance

Permutation Feature Importance

Global Surrogate

Local Surrogate (LIME)

Shapley Values

SHAP

Understanding Model Predictions

Partial Dependence Plots

DeepLearning.AI

Partial Dependence Plots (PDP)

A partial dependence plot shows:

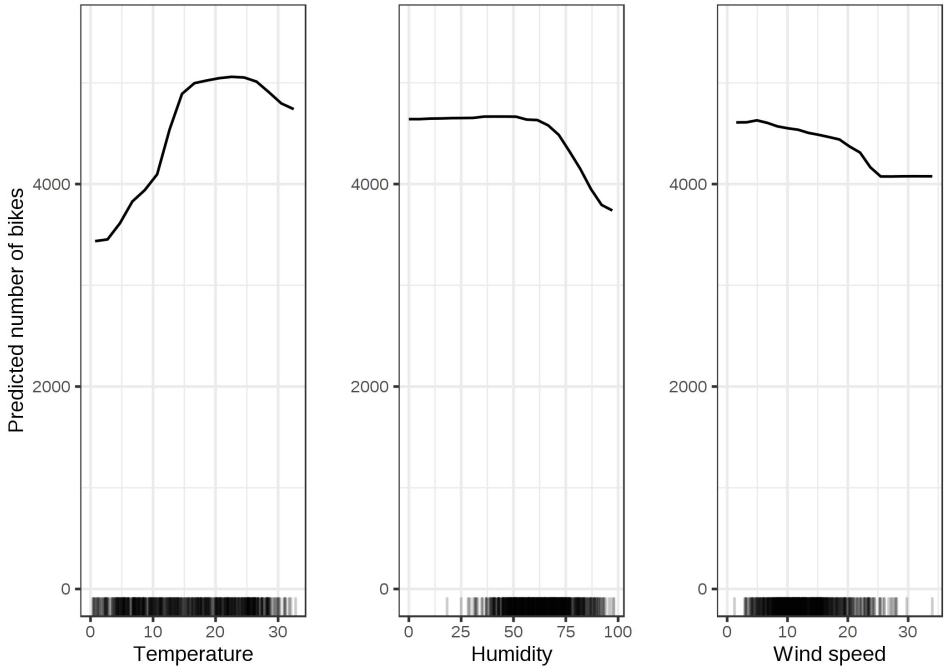
- The marginal effect one or two features have on the model result
- Whether the relationship between the targets and the feature is linear, monotonic, or more complex

Partial Dependence Plots

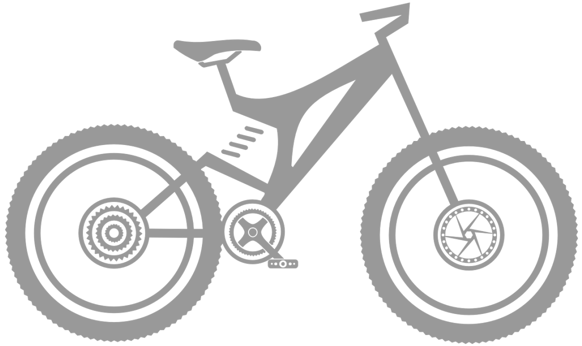
The partial function f_{xs} is estimated by calculating averages in the training data:

$$\hat{f}_{x_S}(x_S) = \frac{1}{n}\sum_{i=1}^{n}\hat{f}(x_S, x_C^{(i)})$$

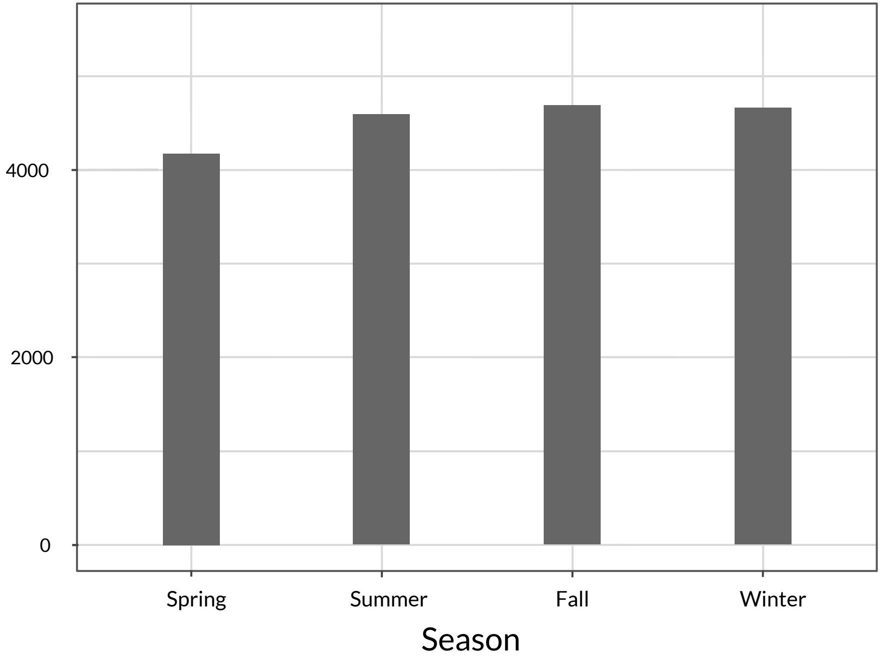
Partial Dependence Plots: Examples



PDP plots for a linear regression model trained on a bike rentals dataset to predict the number of bikes rented



PDP for Categorical Features



Advantages of PDP

- Computation is intuitive
- If the feature whose PDP is calculated has no feature correlations, PDP perfectly represents how feature influences the prediction on average
- Easy to implement

Disadvantages of PDP

- Realistic maximum number of features in PDP is 2
- PDP assumes that feature values have no interactions

DeepLearning.AI

Understanding Model Predictions

Permutation Feature Importance

Permutation Feature Importance

Feature importance measures the increase in prediction error after permuting the features

Feature is **important** if:

- Shuffling its values increases model error

Feature is **unimportant** if:

- Shuffling its values leaves model error unchanged

Permutation Feature Importance

- Estimate the original model error
- For each feature:
  - Permute the feature values in the data to break its association with the true outcome
  - Estimate error based on the predictions of the permuted data
  - Calculate permutation feature importance
  - Sort features by descending feature importance .

Advantages of Permutation Feature Importance

- Nice interpretation: Shows the increase in model error when the feature's information is destroyed.
- Provides global insight to model’s behaviour
- Does not require retraining of model

Disadvantages of Permutation Feature Importance

- It is unclear if testing or training data should be used for visualization
- Can be biased since it can create unlikely feature combinations in case of strongly correlated features
- You need access to the labeled data

Understanding Model Predictions

Shapley Values

Shapley Value

- The Shapley value is a method for assigning payouts to players depending on their contribution to the total
- Applying that to ML we define that:
  - Feature is a “player” in a game
  - Prediction is the “payout”
  - Shapley value tells us how the “payout” (feature contribution) can be distributed among features

Shapley Value: Example

Suppose you trained an ML model to predict apartment prices

You need to explain why the model predicts €300,000 for a certain apartment.

Average prediction of all apartments: €310,000.

Shapley Value

Term in Game Theory	Relation to ML	Relation to House Prices Example
Game	Prediction task for single instance of dataset	Prediction of house prices for a single instance
Gain	Actual prediction for instance - Average prediction for all instances	Prediction for house price (€300,000) - Average Prediction(€310,000) = -€10,000
Players	Feature values that contribute to prediction	‘Park=nearby’, ‘cat=banned’, ‘area=50m^2’, ‘floor=2nd’

Shapley Value

Goal :
Explain the difference between the actual prediction (€300,000) and the average prediction (€310,000): a difference of -€10,000.

Feature	Contribution
'park-nearby'	€30,000
size-50	€10,000
floor-2nd	€0
cat-banned	-€50,000
Total: -€10,000 (Final prediction - Average Prediction)	

One possible explanation

Advantages of Shapley Values

Based on solid theoretical foundation.
Satisfies Efficiency, Symmetry, Dummy, and Additivity properties

Value is fairly distributed among all features

Enables contrastive explanations

Disadvantages of Shapley Values

- Computationally expensive
- Can be easily misinterpreted
- Always uses all the features, so not good for explanations of only a few features.
- No prediction model. Can’t be used for “what if” hypothesis testing.
- Does not work well when features are correlated

DeepLearning.AI

Understanding Model Predictions

SHAP (SHapley Additive exPlanations)

SHAP

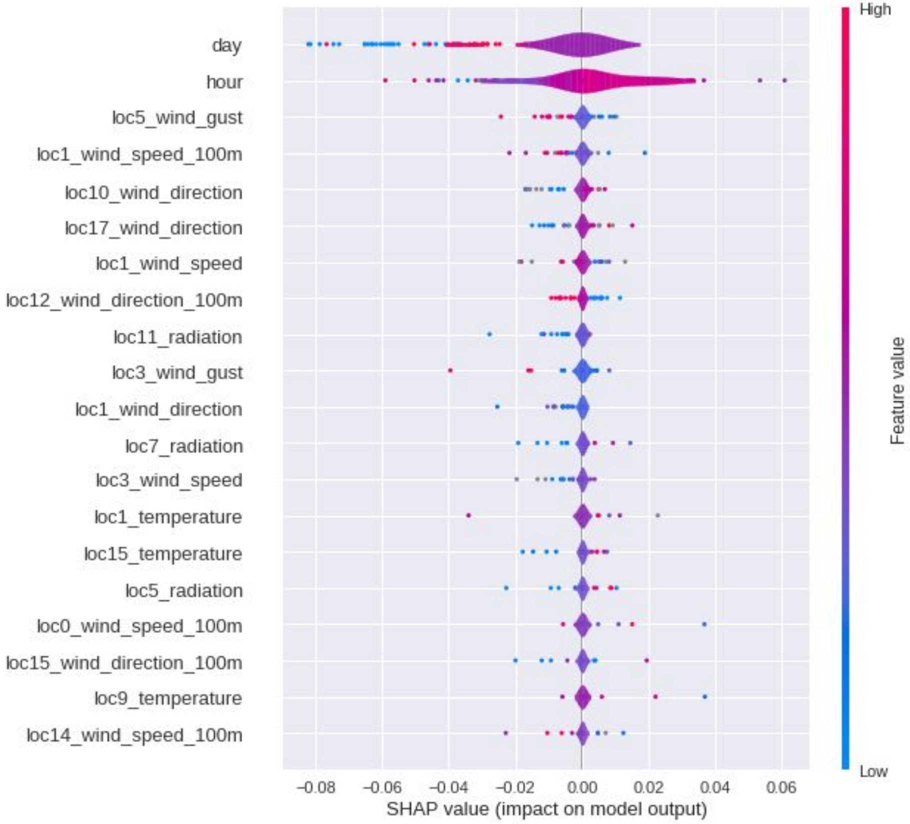
- SHAP (SHapley Additive exPlanations) is a framework for Shapley Values which assigns each feature an importance value for a particular prediction

- Includes extensions for:
    - TreeExplainer: high-speed exact algorithm for tree ensembles
    - DeepExplainer: high-speed approximation algorithm for SHAP values in deep learning models
    - GradientExplainer: combines ideas from Integrated Gradients, SHAP, and SmoothGrad into a single expected value equation
    - KernelExplainer: uses a specially-weighted local linear regression to estimate SHAP values for any model

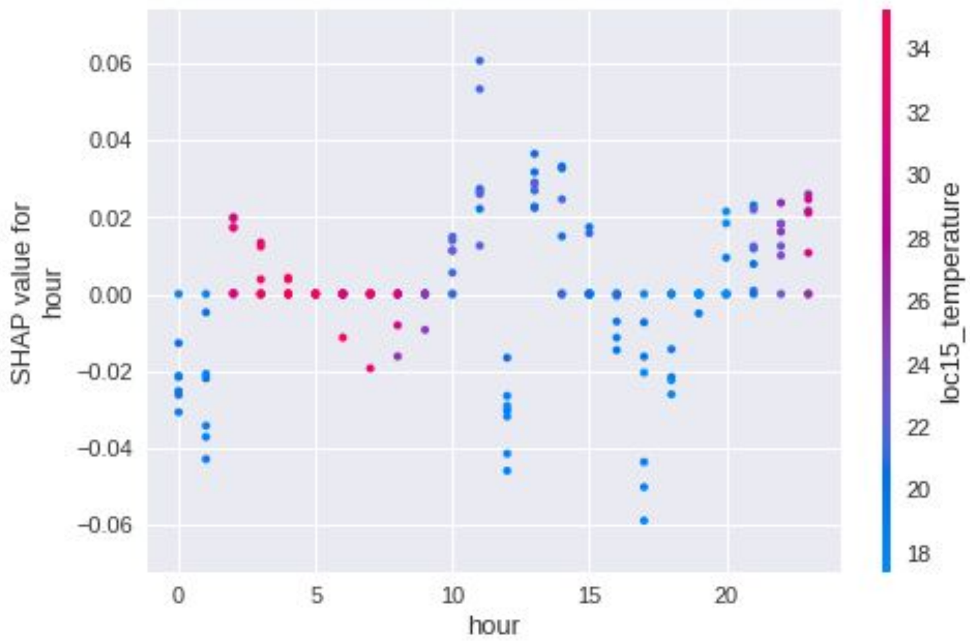
SHAP Explanation Force Plots

- Shapley Values can be visualized as forces
- Prediction starts from the baseline (Average of all predictions)
- Each feature value is a force that increases (red) or decreases (blue) the prediction

SHAP Summary Plot



SHAP Dependence Plot with Interaction



DeepLearning.AI

Understanding Model Predictions

Testing Concept Activation Vectors

Testing Concept Activation Vectors (TCAV)

Concept Activation Vectors (CAVs)

- A neural network's internal state in terms of human-friendly concepts
- Defined using examples which show the concept

Example Concepts

Understanding Model Predictions

LIME

DeepLearning.AI

Local Interpretable Model-agnostic Explanations (LIME)

- Implements local surrogate models - interpretable models that are used to explain individual predictions
- Using data points close to the individual prediction, LIME trains an interpretable model to approximate the predictions of the real model
- The new interpretable model is then used to interpret the real result

Understanding Model Predictions

DeepLearning.AI

AI Explanations

Google Cloud AI Explanations for AI Platform

Explain why an individual data point received that prediction

Debug odd behavior from a model

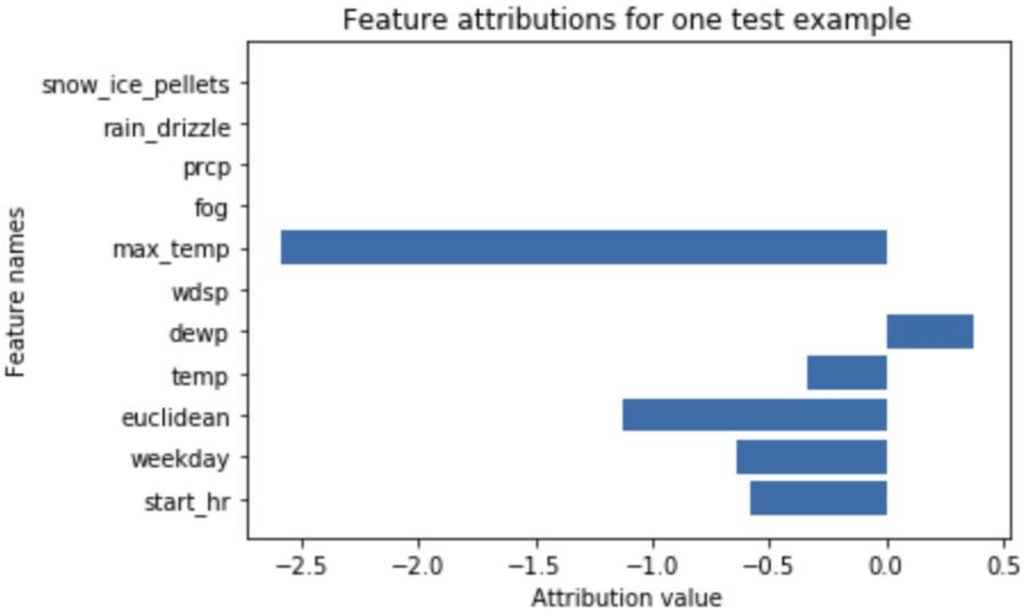
Refine a model or data collection process

Verify that the model's behavior is acceptable

Present the gist of the model

AI Explanations: Feature Attributions

```
Predicted duration: 11.1651134 minutes
Actual duration: 10.0 minutes
```



Tabular Data Example

AI Explanations: Feature Attributions

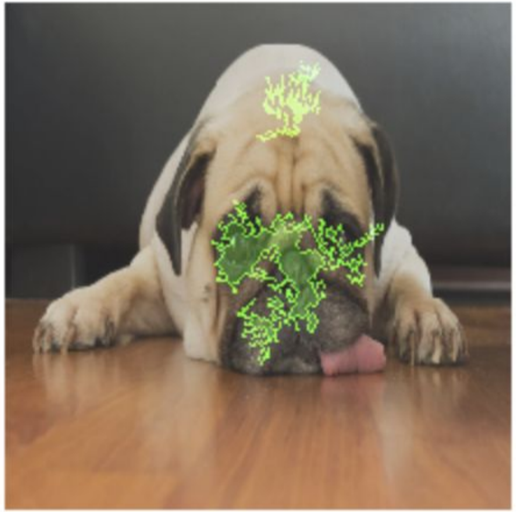
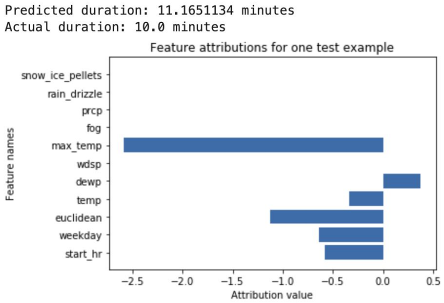
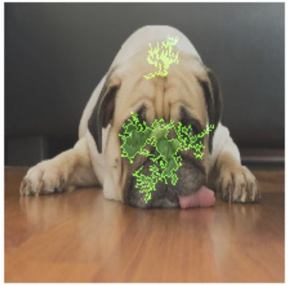


Image Data Examples

AI Explanations: Feature Attribution Methods





AI Explanations: Integrated Gradients

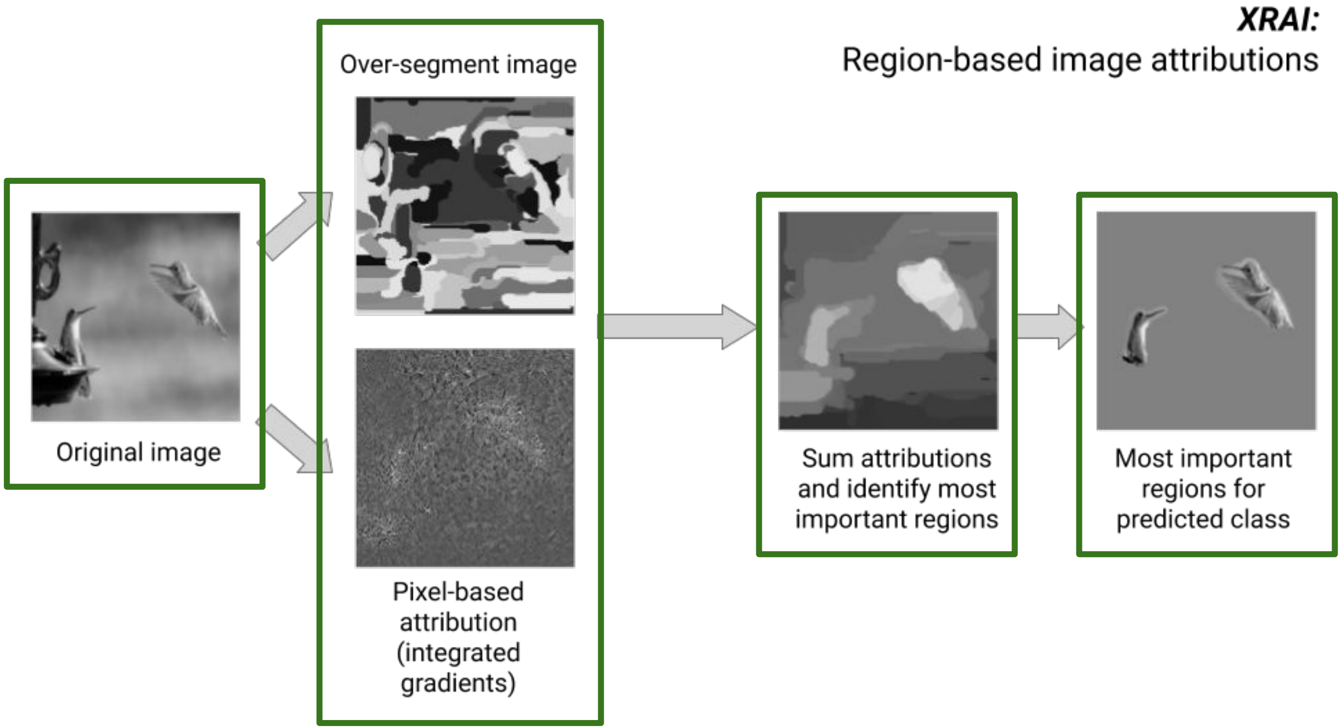
A gradients-based method to efficiently compute feature attributions with the same axiomatic properties as Shapley values

AI Explanations: XRAI (eXplanation with Ranked Area Integrals)

XRAI assesses overlapping regions of the image to create a saliency map

- Highlights relevant regions of the image rather than pixels
- Aggregates the pixel-level attribution within each segment and ranks the segments

AI Explanations: XRAI (eXplanation with Ranked Area Integrals)



DeepLearning.AI