Transparency and Interpretability

ICS 491

Model Interpretability

Interpretability

Example: longevity predictor

$$Y = M_1X_1 + M_2X_2 + M_3X_3 + B$$

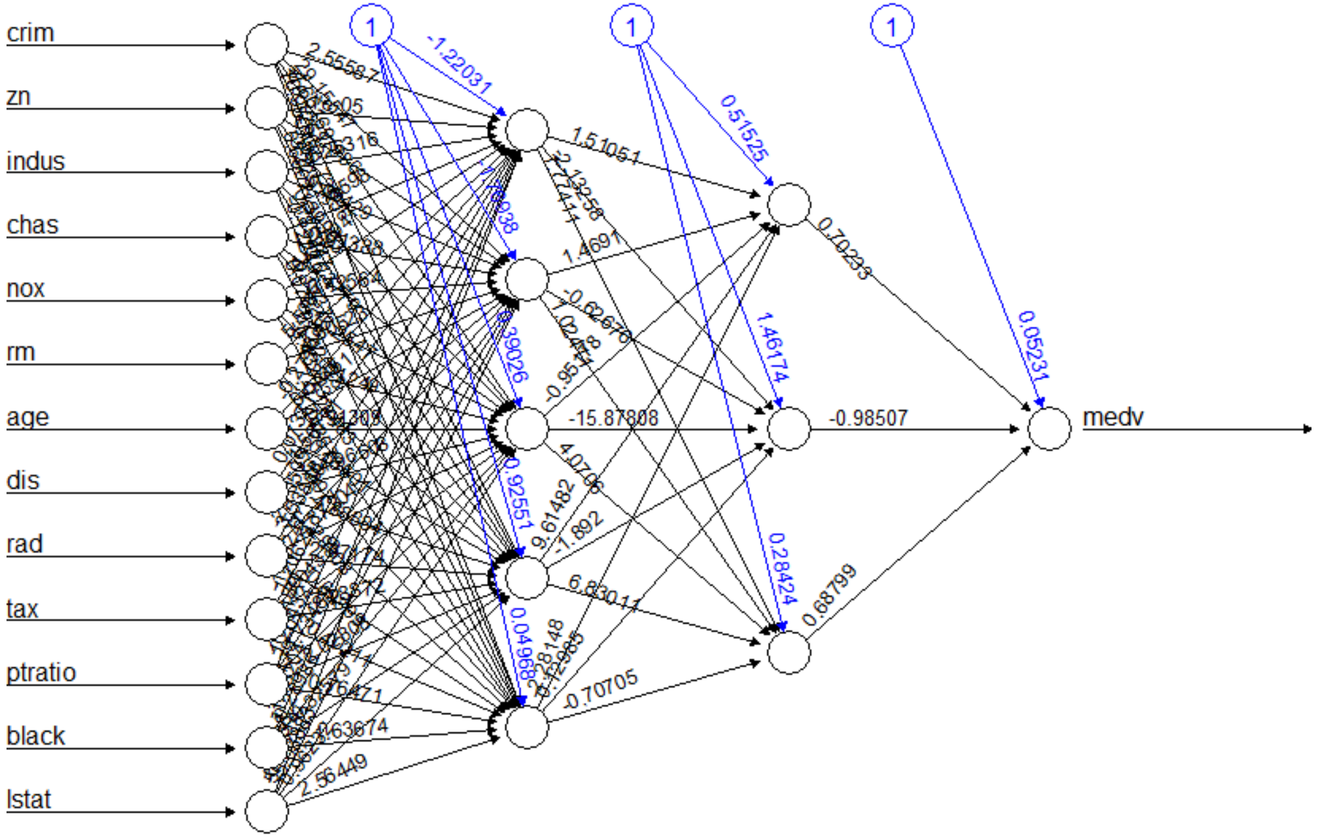
$$Y = 3X_1 + 9X_2 - 16X_3 + 2$$

X_1 is mean lifespan of immediate family members over the past 100 years who share your gender (proxy for genetics)

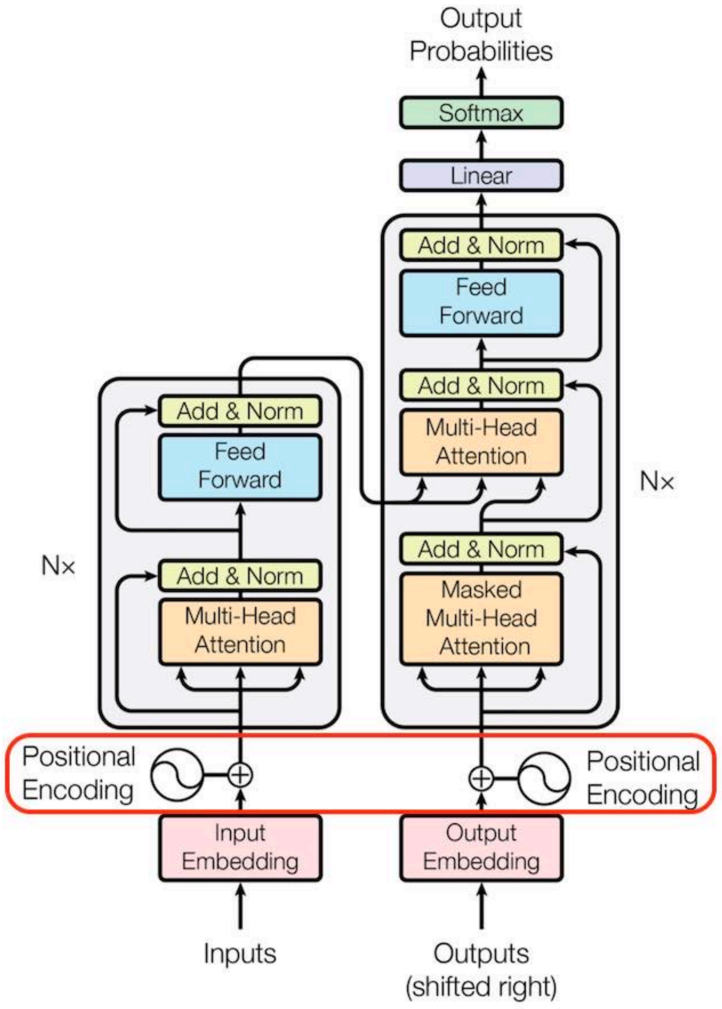
X_2 is hours of exercise per day

X_3 is mean saturated fat per day

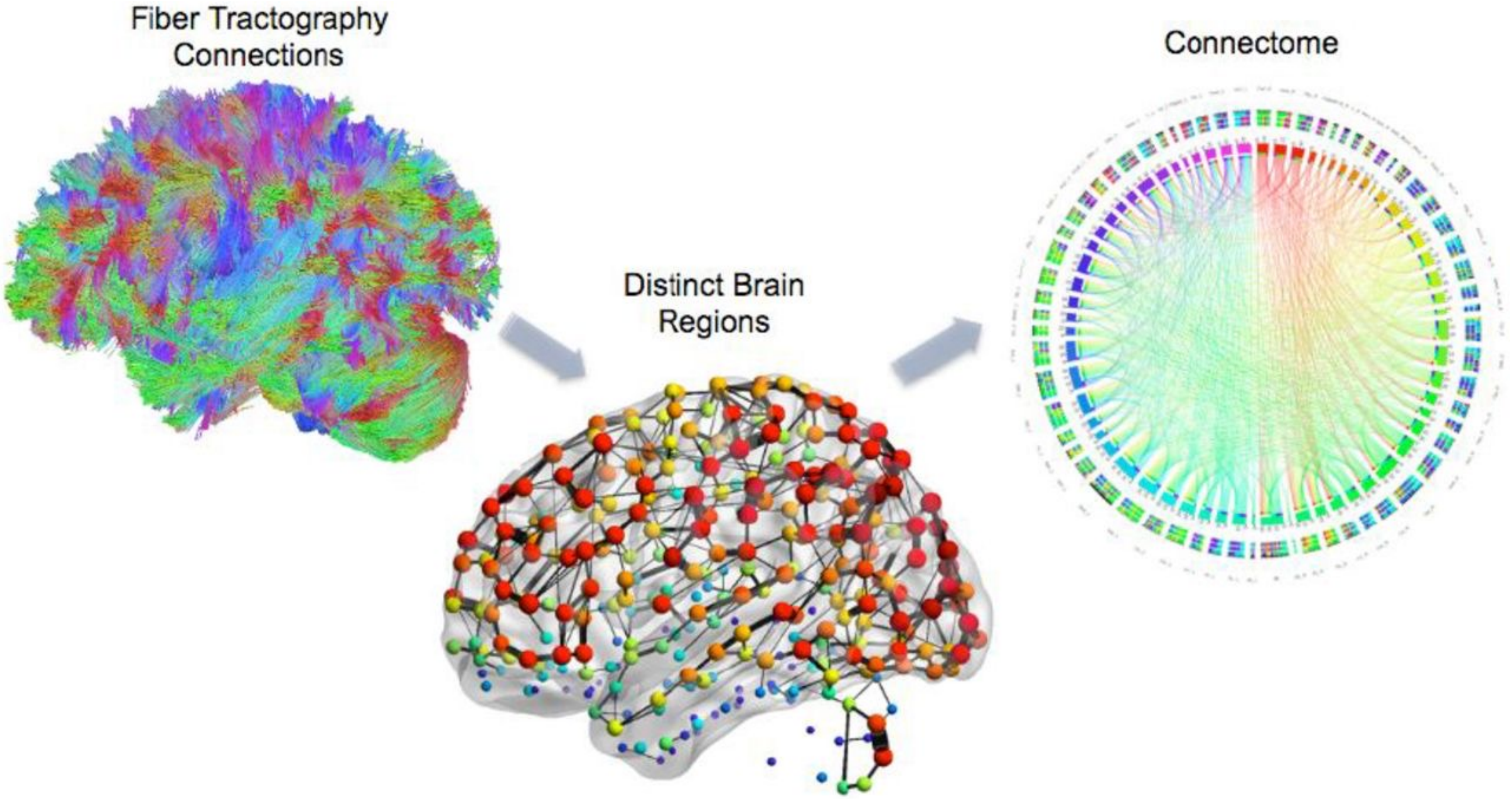
Interpretable?



Interpretable?



Interpretable?



Interpretability vs explainability

- Interpretable/explainable AI is a core topic in HAI worth 2 classes
- Many definitions out there, but for the purposes of this class, we will go with Amazon's distinction

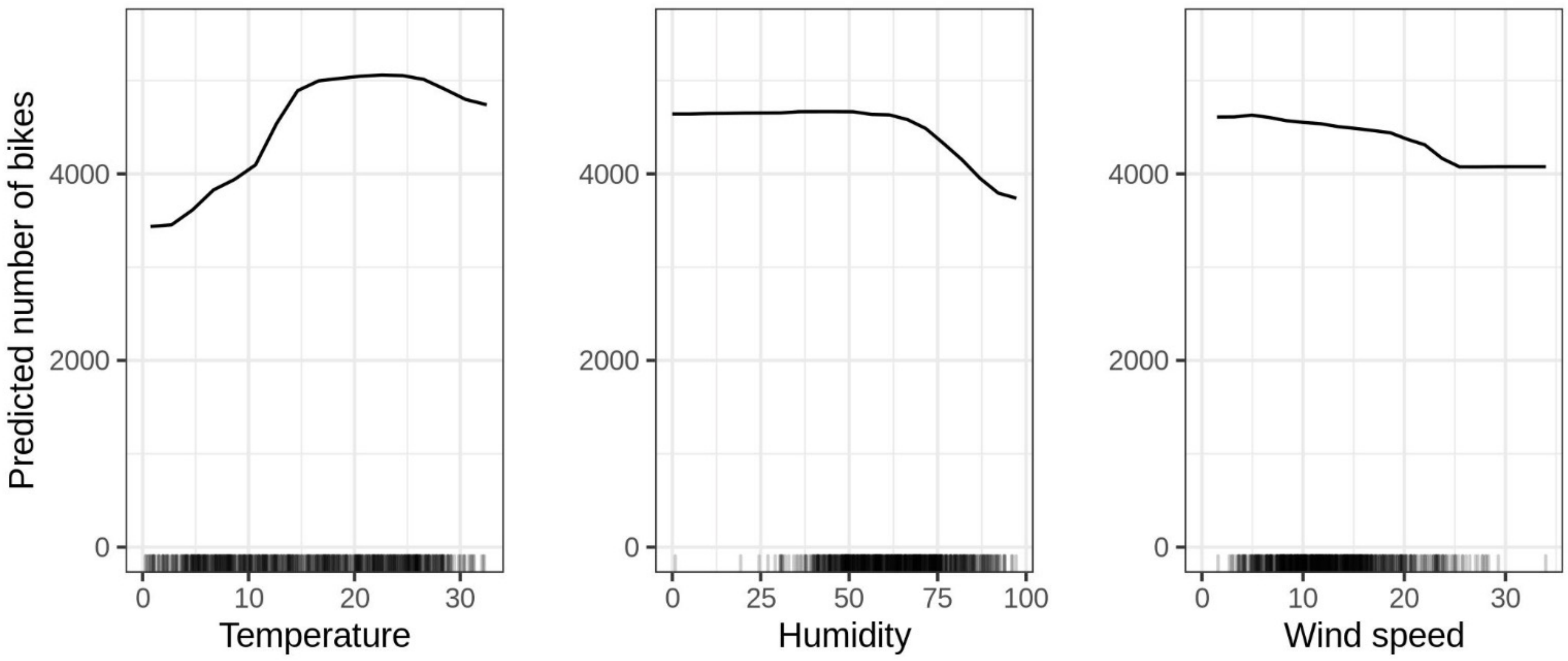
Interpretability — If a business wants high model transparency and wants to understand exactly why and how the model is generating predictions, they need to observe the inner mechanics of the AI/ML method. This leads to interpreting the model's weights and features to determine the given output. This is interpretability.

Explainability — Explainability is how to take an ML model and explain the behavior in human terms. With complex models (for example, black boxes), you cannot fully understand how and why the inner mechanics impact the prediction. However, through model agnostic methods (for example, partial dependence plots, SHapley Additive exPlanations (SHAP) dependence plots, or surrogate models) you can discover meaning between input data attributions and model outputs, which enables you to explain the nature and behavior of the AI/ML model.

Related concepts

- Model trustworthiness
- AutoML / neural architecture search

Partial dependence plots

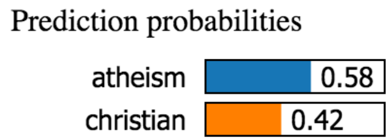


Christoph Molnar

Black-Box Explainability

Perturbation of Model inputs

- Create synthetic data with only part of the original attributes
  - “I love ICS691D! I have attended every class.” → “I ICS691D! I have attended every class.”
- Classify the synthetic data points
  - Sentiment(“I ICS691D! I have attended every class.”) = 0.53
- Measure the importance of each attribute by the performance of the models with and without the features
  - Sentiment(“I love ICS691D! I have attended every class”) = 0.9
  - 0.95 – 0.53 = 0.42, so the word “love” has much importance



https://homes.cs.washington.edu/~marcotcr/blog/lime/

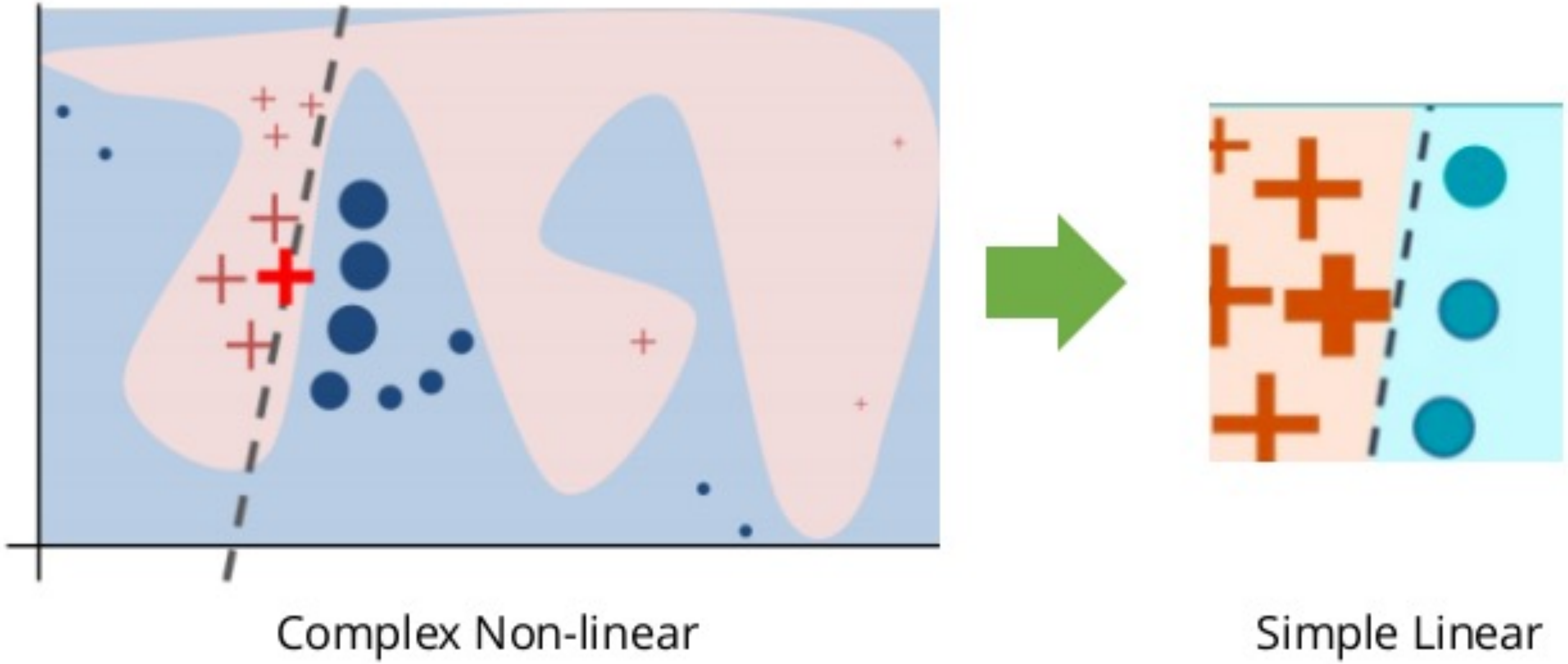
Text with highlighted words

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,

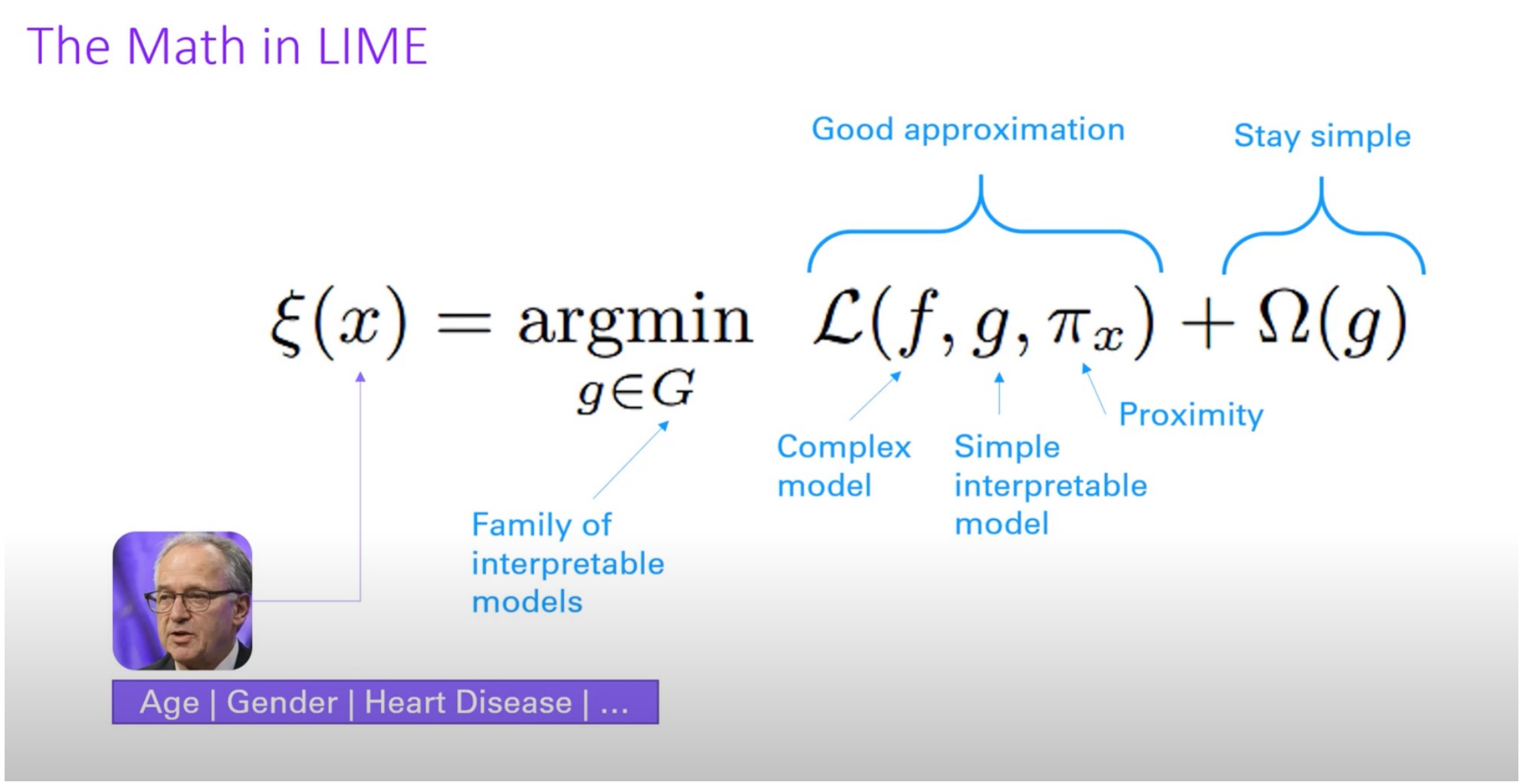
There have been some notes recently asking where to obtain the DARWIN fish.
This is the same question I have and I have not seen an answer on the
net. If anyone has a contact please post on the net or email me.

Surrogate models



https://deepandshallowml.files.wordpress.com/2019/11/lime_intuition_final.png

Local interpretable model-agnostic explanations (LIME): Perturbation + surrogation



DeepFindr YouTube channel

SHAP (Shapley additive explanations)

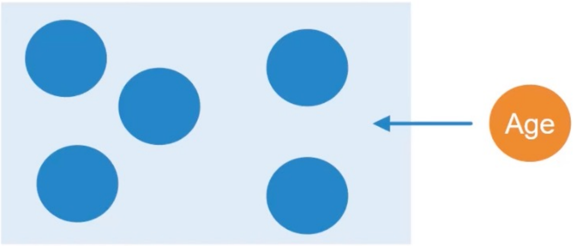
- Based on Shapley values from cooperative/coalitional game theory
- Set of players S
- v(S) is a function that maps coalition S to a real number ("worth" of S)
- Contribution of player i to the coalition S:

$$\varphi_i(v) = \frac{1}{\text{number of players}} \sum_{\text{coalitions excluding } i} \frac{\text{marginal contribution of } i \text{ to coalition}}{\text{number of coalitions excluding } i \text{ of this size}}$$

Can think of coalition S as all input features to a model, where each player i is an input feature

SHAP (Shapley additive explanations)

Calculating shapley values



Blackbox model

Input datapoint

Age $\phi_i(f, x) = \sum_{z' \subseteq x'} \underbrace{\frac{|z'|!(M - |z'| - 1)!}{M!}}_{\text{Weighting}} \underbrace{[f_x(z') - f_x(z' \setminus i)]}_{\text{Contribution}}$

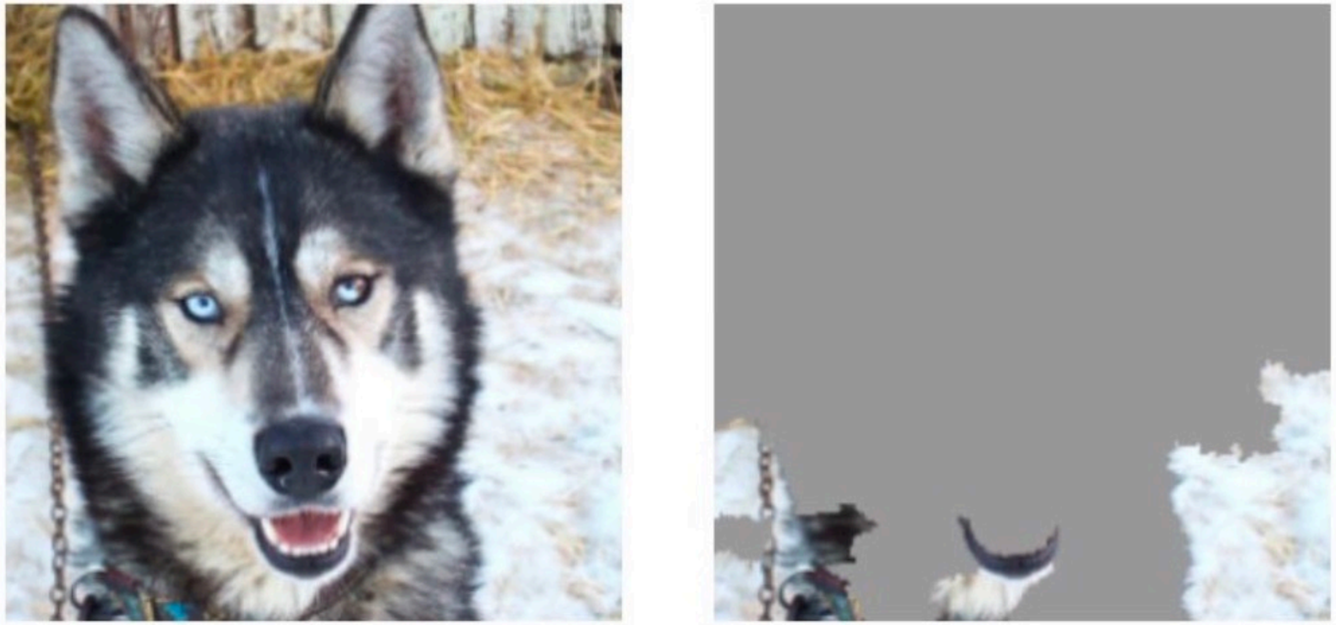
Shapley value for feature i

Subset

Simplified data input

DeepFindr YouTube channel

Saliency maps



(a) Husky classified as wolf (b) Explanation

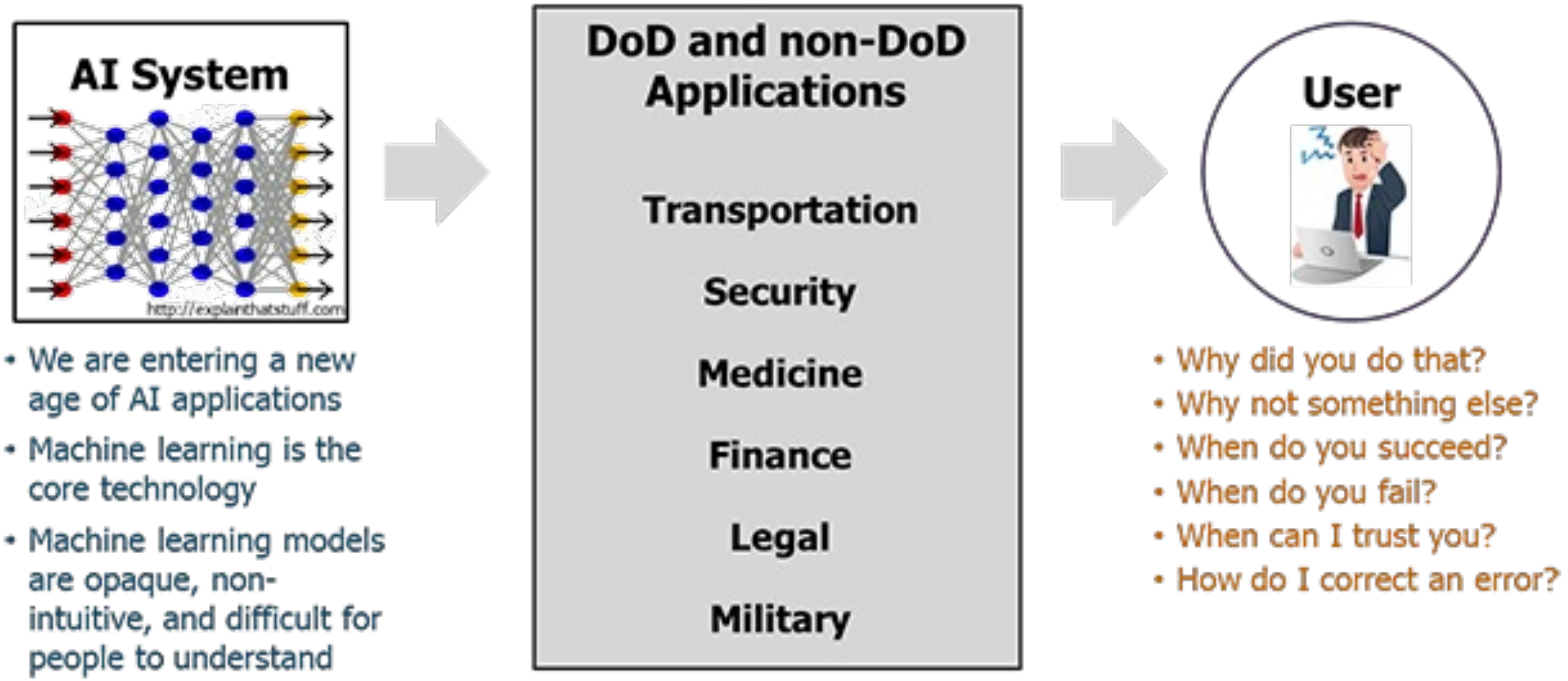
Ribeiro et al. Why should i trust you?" Explaining the predictions of any classifier. SIGKDD 2016.

How to construct a saliency map

Many approaches.

- **Occlusion- or perturbation-based:** Methods like SHAP and LIME manipulate parts of the image to generate explanations (model-agnostic).
- **Gradient-based:** Many methods compute the gradient of the prediction (or classification score) with respect to the input features. The gradient-based methods (of which there are many) mostly differ in how the gradient is computed.
  1. Perform a forward pass of the image of interest.
  2. Compute the gradient of class score of interest with respect to the input pixels.
  3. Visualize the gradients. You can either show the absolute values or highlight negative and positive contributions separately.

Research funding in XAI: DoD, NSF, NIH, …



Venture capital funding in XAI

FEATURE STORY

Kyndi Secures $20 Million in Funding Led by Intel Capital to Advance Industry's First Explainable AI Platform

Kenn So
Sep 23, 2019 · 8 min read · Member-only ·

Why explainable AI is exciting to VCs

What is driving the demand, how incumbents are responding, and how startups are already tackling explainability 2.0

Fiddler Labs Raises $10.2M in Series A Funding to Make AI Explainable in Every Enterprise

New VC fund Curiosity launches with first investment in explainable AI startup Deeploy

March 16, 2022
By Curiosity

Anthropic's quest for better, more explainable AI attracts $580M

Devin Coldewey @techcrunch / 6:58 AM HST • April 29, 2022   Comment

TECH · ARTIFICIAL INTELLIGENCE

Why investors are backing this former Facebook manager's 'explainable A.I.' startup

BY JONATHAN VANIAN
June 17, 2021 at 2:00 AM HST

XAI Products

Google Cloud Overview Solutions Products Pricing Resources Docs Support English Consol

Explainable AI

Features

Understand AI output with groundbreaking XAI tools, developed by Google Research and used to power AI at Google.

Feature attributions

A managed service for generating feature attributions. Supported methods include Samples Shapely, Integrated Gradients, and XRAI.

Integrated into Vertex AI services, including AutoML Tables and Vision, Vertex AI Prediction, Notebooks, Model Monitoring and BigQuery ML.

Learn more

Example-based Explanations (Preview)

Build better models with actionable explanations to mitigate data challenges.

A managed Approximate Nearest Neighbor Service for returning similar examples to new predictions or instances.

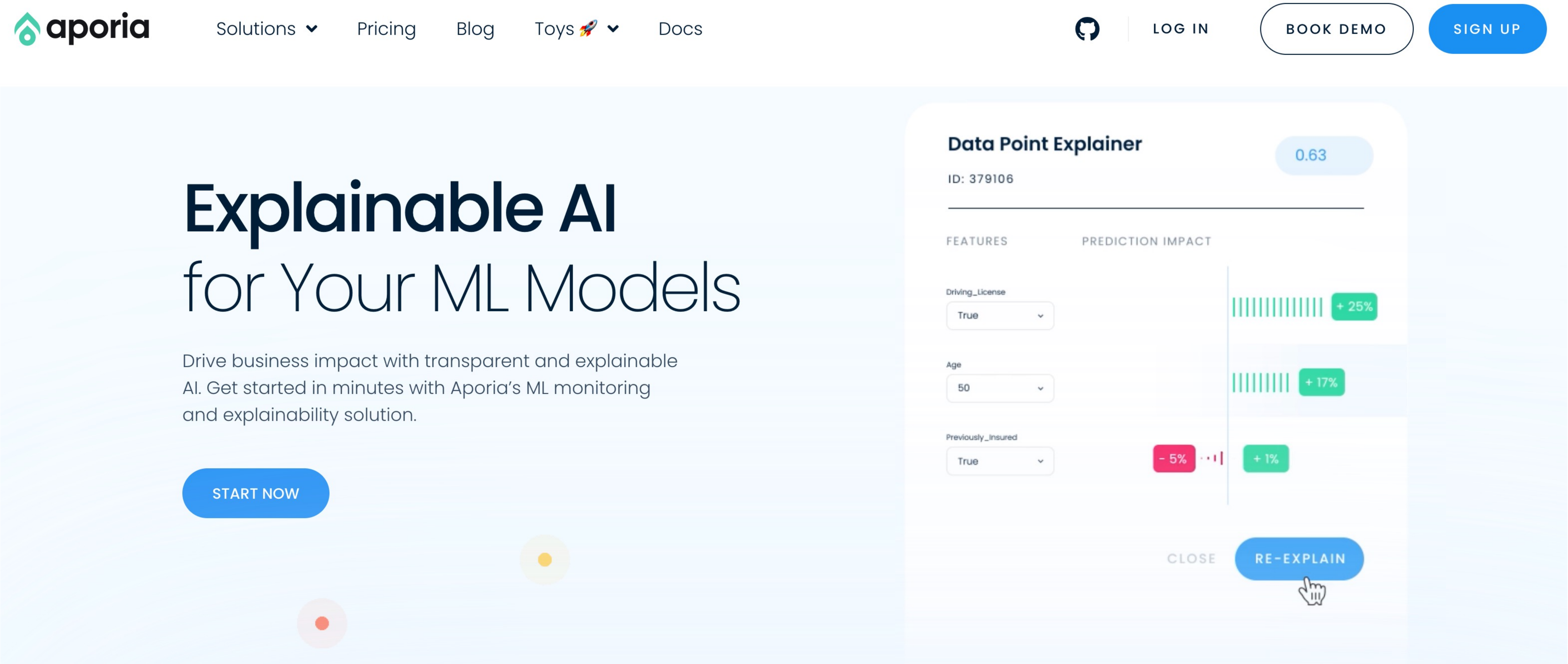
Learn more

Model analysis

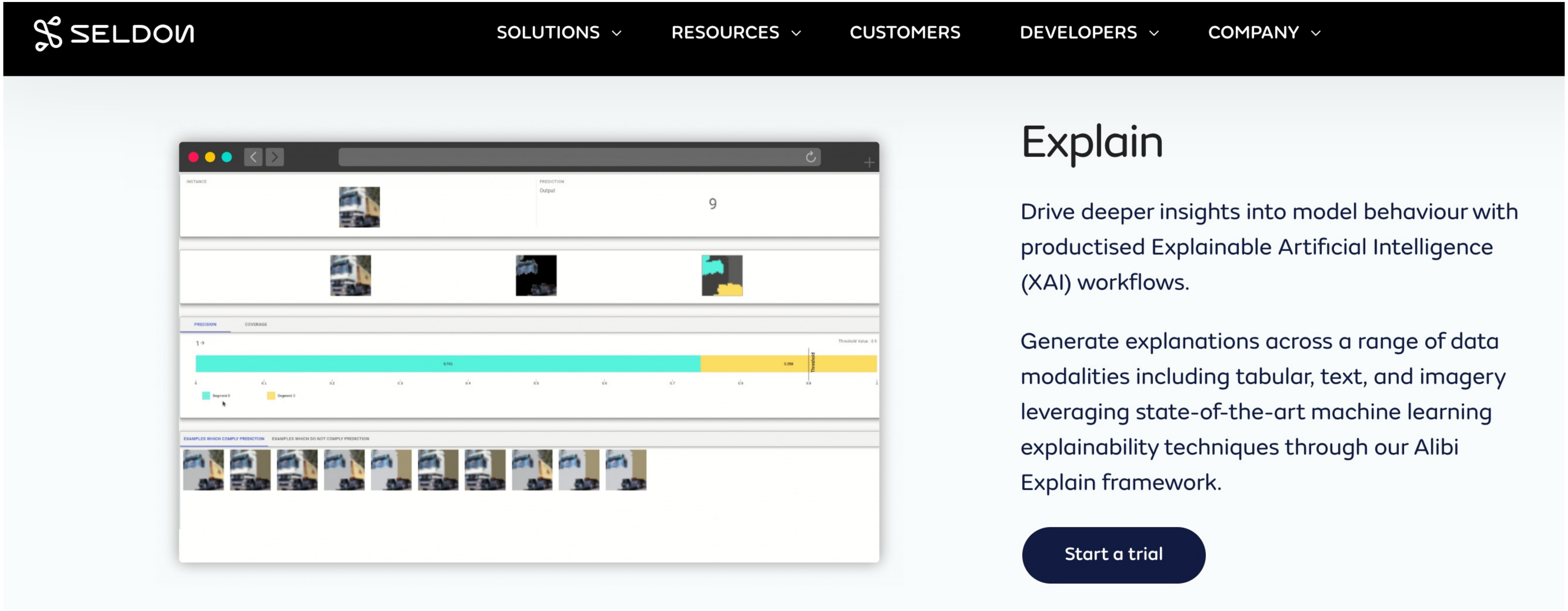
An advanced model analysis toolkit to help you better understand models.

Take action in Vertex AI to inspect models through an interactive dashboard with the integrated What-If Tool.

XAI Products



XAI Products



XAI Products

DataRobot

Platform ⌄ Solutions ⌄ Success ⌄ Customers ⌄ Resources ⌄ Partners ⌄ Company ⌄ Start for Free

Wiki search

Topics Alphabetic

− Artificial Intelligence
- AI Engineer
- Artificial Intelligence (AI)
- Artificial Intelligence Wiki
- Cognitive Computing
- Explainable AI
- Natural Language Processing

+ Data
+ Data Science
+ Features
+ Machine Learning
+ Modeling
+ Predictions

- Prediction Explanation: Highlights the features variables that impact each model's decision outcome for each record and the magnitude of different features for each.

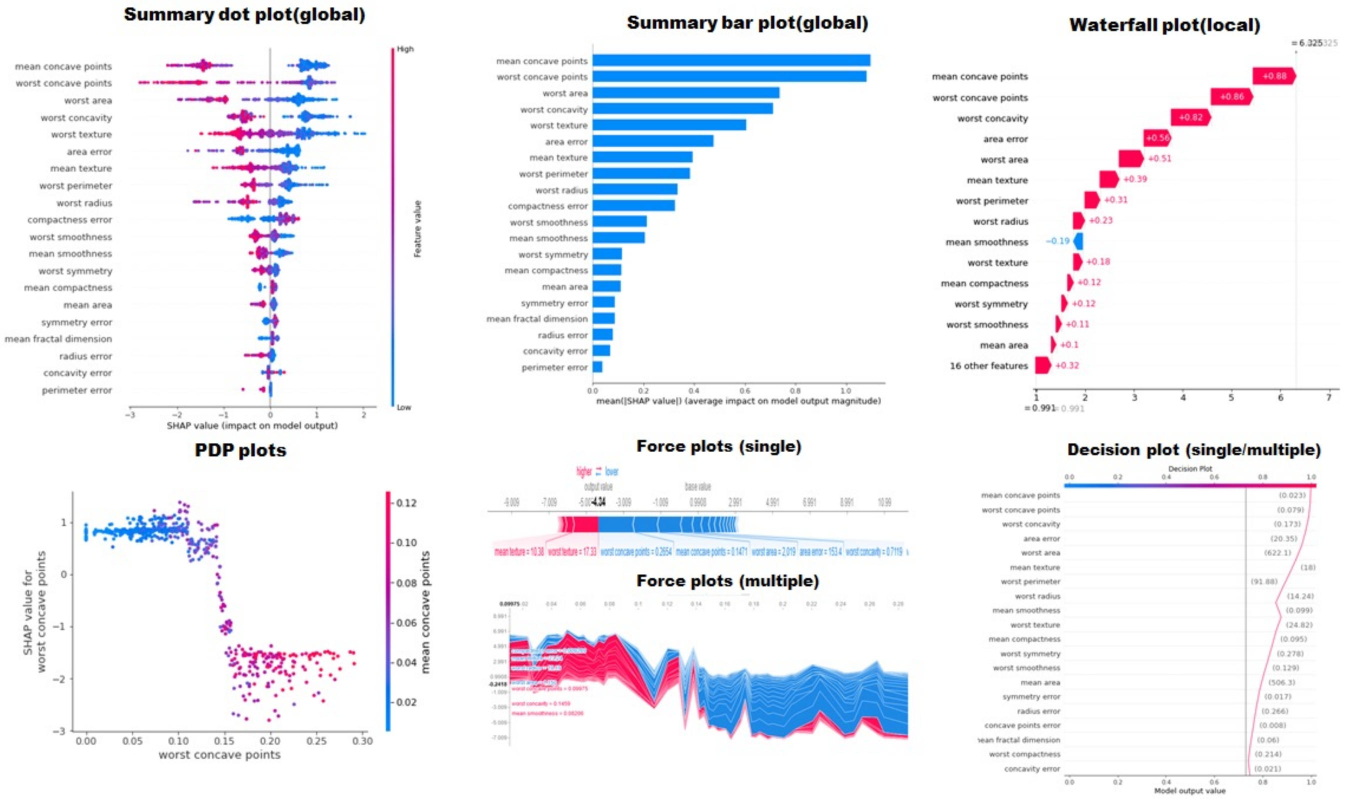
DataRobot automates several standard data processing steps within each model blueprint and makes all these transformations transparent. This ensures that AI models are not locked in a black box, a common problem that can arise when organizations turn to third-party technology suppliers to address their AI solutions. Our products are designed to help your organization build trustworthy AI models for a wide array of use cases and to promote the democratization of data science and machine learning tools.

So many startup ideas left untapped

XAI in Python: SHAP

SHAP GitHub

XAI in Python: SHAP

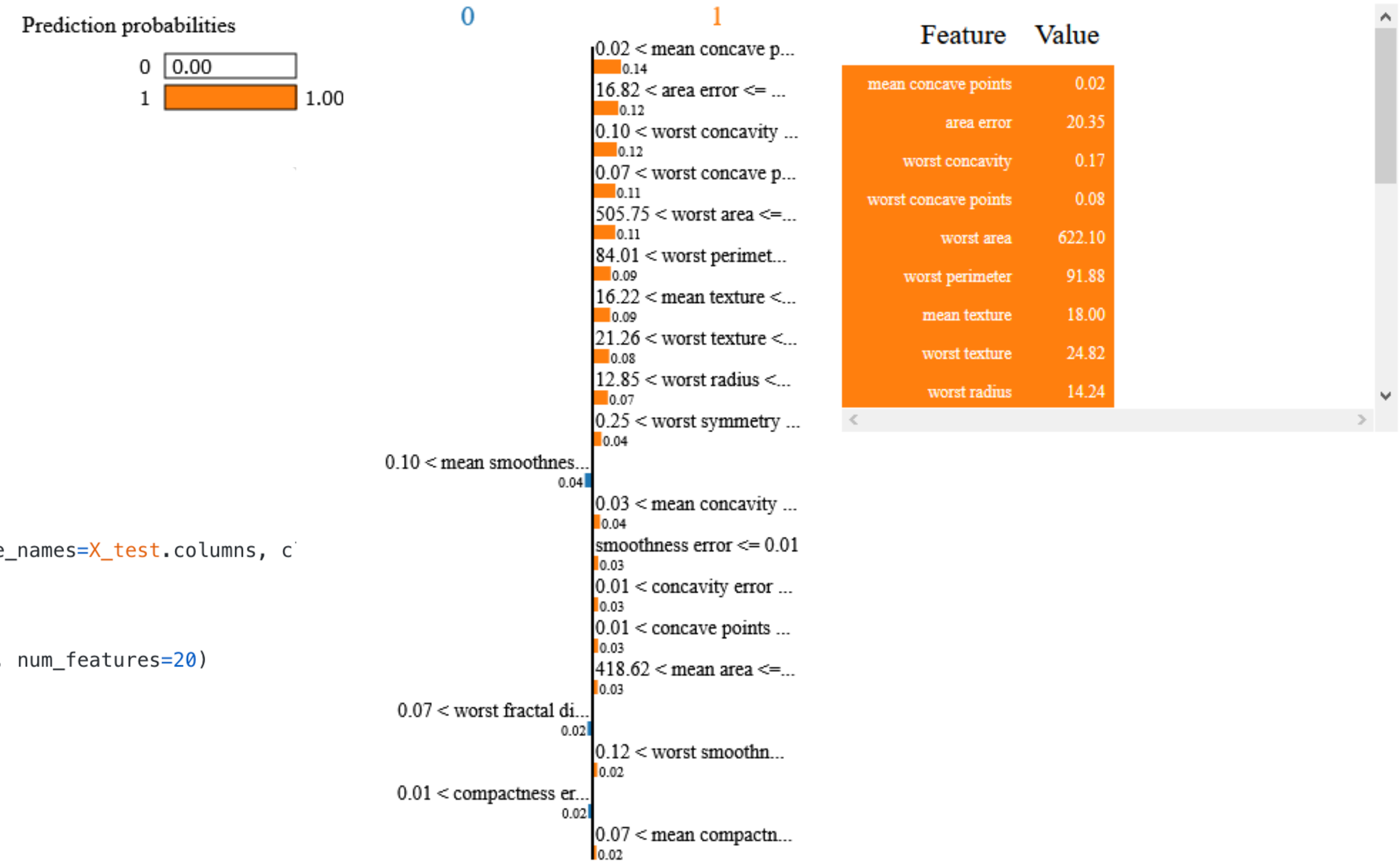


https://towardsdatascience.com/explainable-ai-xai-a-guide-to-7-packages-in-python-to-explain-your-models-932967f0634b

```
# Generate the Tree explainer and SHAP values
explainer = shap.TreeExplainer(xgb_mod)
shap_values = explainer.shap_values(X)
expected_value = explainer.expected_value

############## visualizations #############
# Generate summary dot plot
shap.summary_plot(shap_values, X,title="SHAP summary plot")

# Generate summary bar plot
shap.summary_plot(shap_values, X,plot_type="bar")

# Generate waterfall plot
shap.plots._waterfall.waterfall_legacy(expected_value, shap_values[79], features=X.loc[79,

# Generate dependence plot
shap.dependence_plot("worst concave points", shap_values, X, interaction_index="mean conca

# Generate multiple dependence plots
for name in X_train.columns:
     shap.dependence_plot(name, shap_values, X)
shap.dependence_plot("worst concave points", shap_values, X, interaction_index="mean conca

# Generate force plot - Multiple rows
shap.force_plot(explainer.expected_value, shap_values[:100,:], X.iloc[:100,:])

# Generate force plot - Single
shap.force_plot(explainer.expected_value, shap_values[0,:], X.iloc[0,:])

# Generate Decision plot
shap.decision_plot(expected_value, shap_values[79],link='logit' ,features=X.loc[79,:], fea
```

XAI In Python: LIME

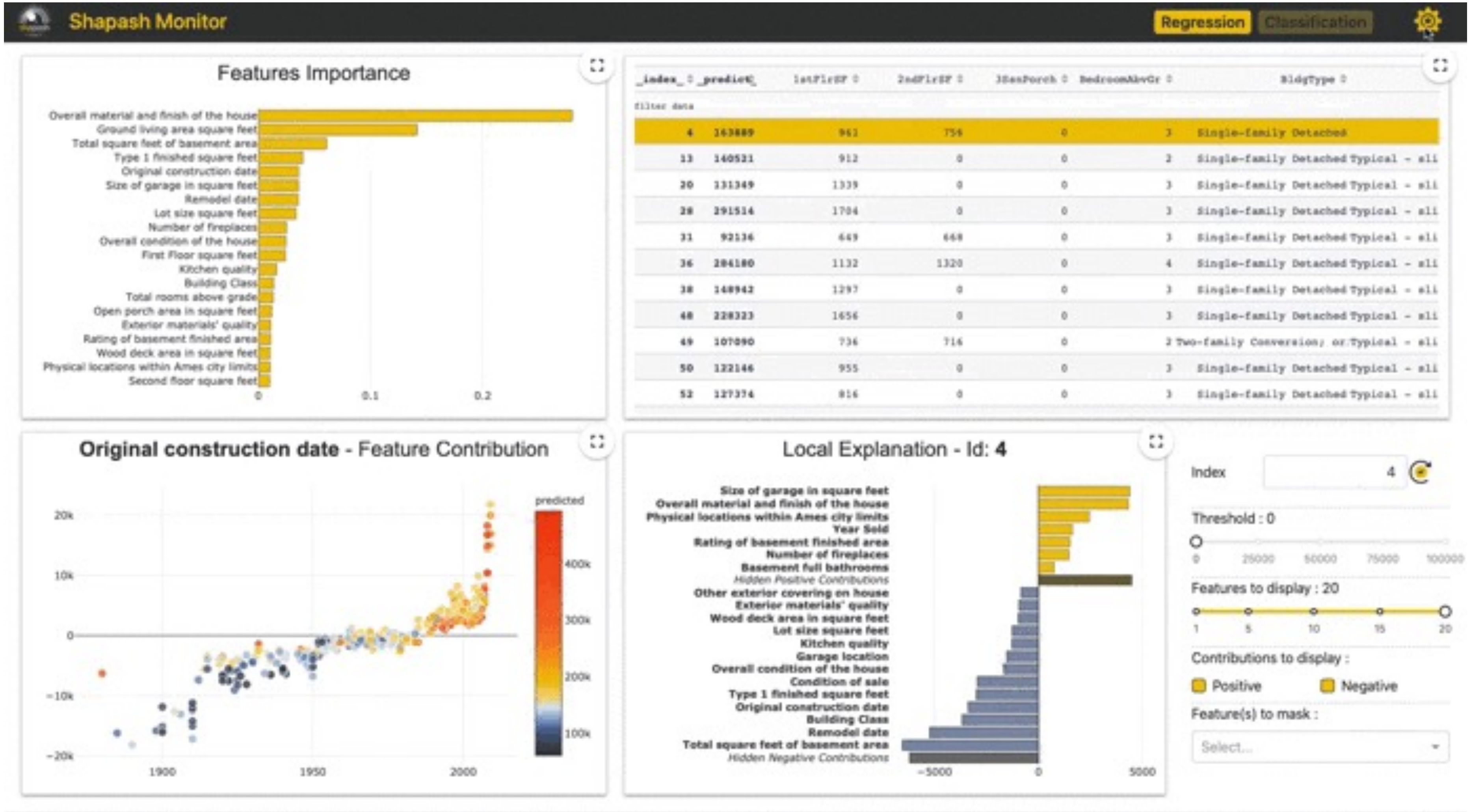
```
Intercept 0.3519728312975602
Prediction_local [1.4287122]
Right: 0.9997322
```




```python
# Utilizing our same xgb_mod model object created above
# Import pacakages
import lime
import lime.lime_tabular
import numpy as np
import xgboost

############## create explainer ###########
# we use the dataframes splits created above for SHAP
explainer =
lime.lime_tabular.LimeTabularExplainer(X_test.to_numpy(), feature_names=X_test.columns, c

############## visualizations #############
exp = explainer.explain_instance(X_np[79], xgb_mod.predict_proba, num_features=20)
exp.show_in_notebook(show_table=True)
```

https://towardsdatascience.com/explainable-ai-xai-a-guide-to-7-packages-in-python-to-explain-your-models-932967f0634b

XAI in Python: shapash



Shapash

XAI in Python: Shapash

```
############## launch the app #############
# create explainer
xpl = SmartExplainer()
xpl.compile(
    x=X_test,
    model=xgb_mod
)
#Creating Application
app = xpl.run_app(title_story='Breast Cancer')

############## visualizations #############
# feature importance based on SHAP
xpl.plot.features_importance()

# contributions plot
xpl.plot.contribution_plot("worst concave points")

# Local explanation
xpl.plot.local_plot(index=79)

# compare plot
xpl.plot.compare_plot(index=[X_test.index[79], X_test.index[80]])

# Interactive interactions widget
xpl.plot.top_interactions_plot(nb_top_interactions=5)
```

XAI in Python

Many, many other APIs in Python.

- SHAP
- LIME
- SHAPASH
- ELI5
- Explainable Boosting Machines (EBM)
- Dalex
- ExplainerDashboard
- Alibi
- Skater
- ExplainX.ai
- …