Explainable AI

Agenda

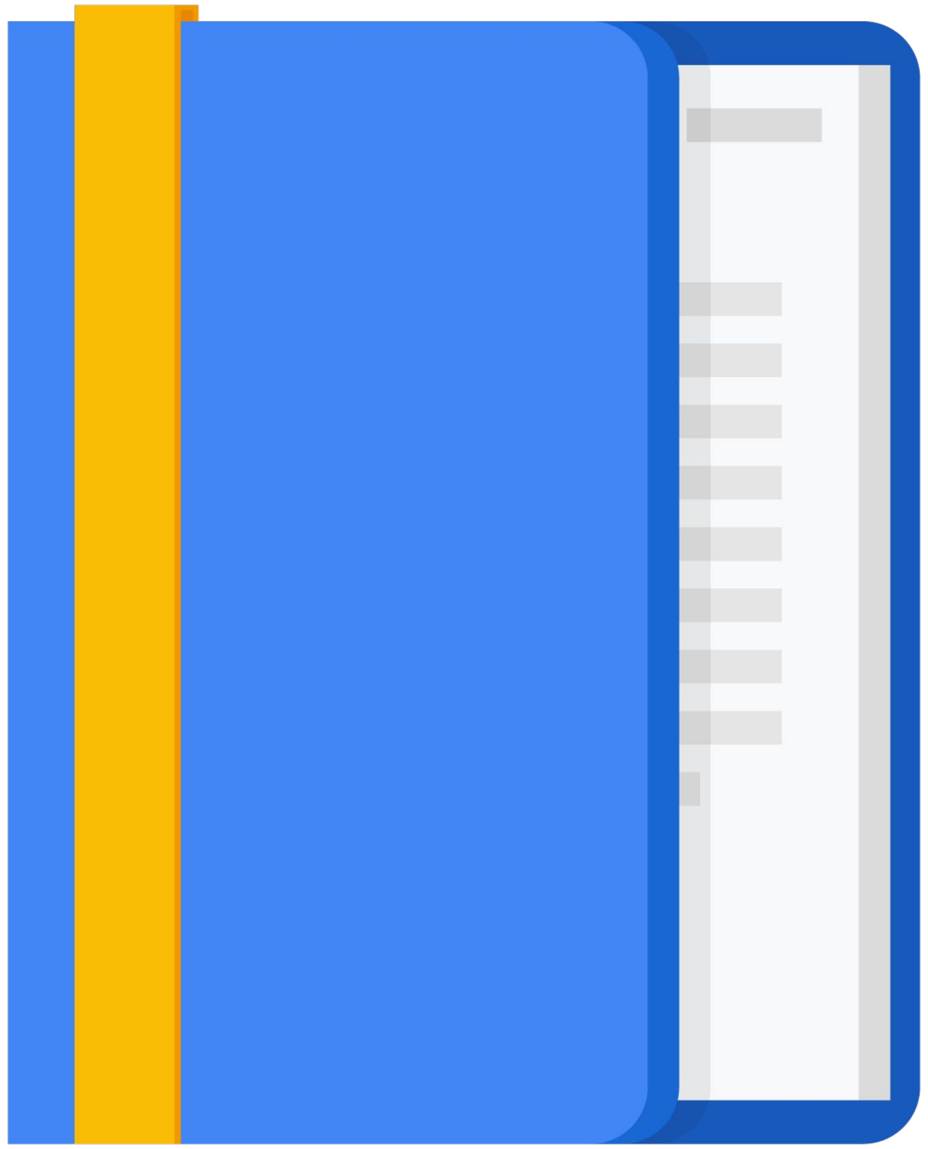
What is Explainable AI?

Interpretable ML methods

Deepdive: Integrated Gradients (IG)

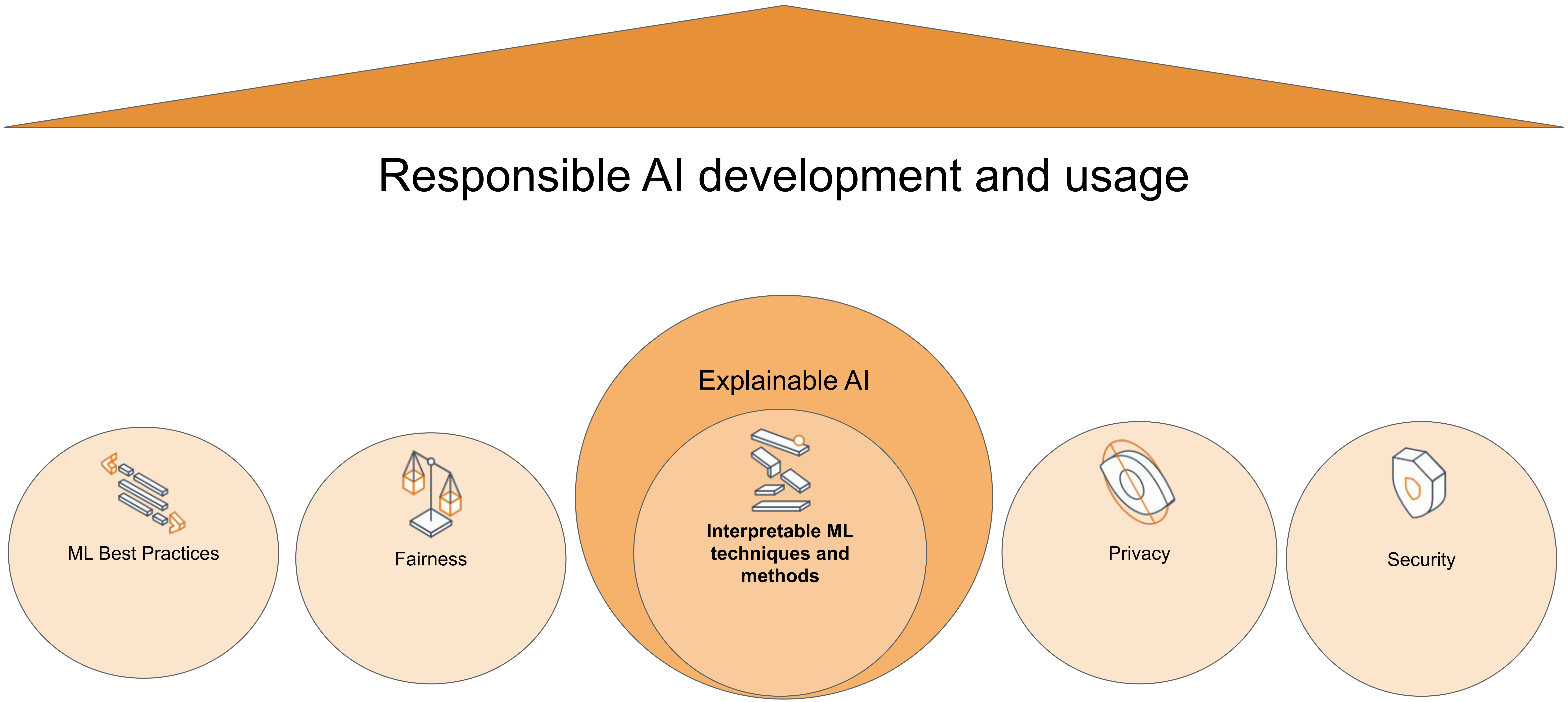
Picking baselines and future research directions

Explainable AI on Google Cloud



Understanding an image classifier with Explainable AI

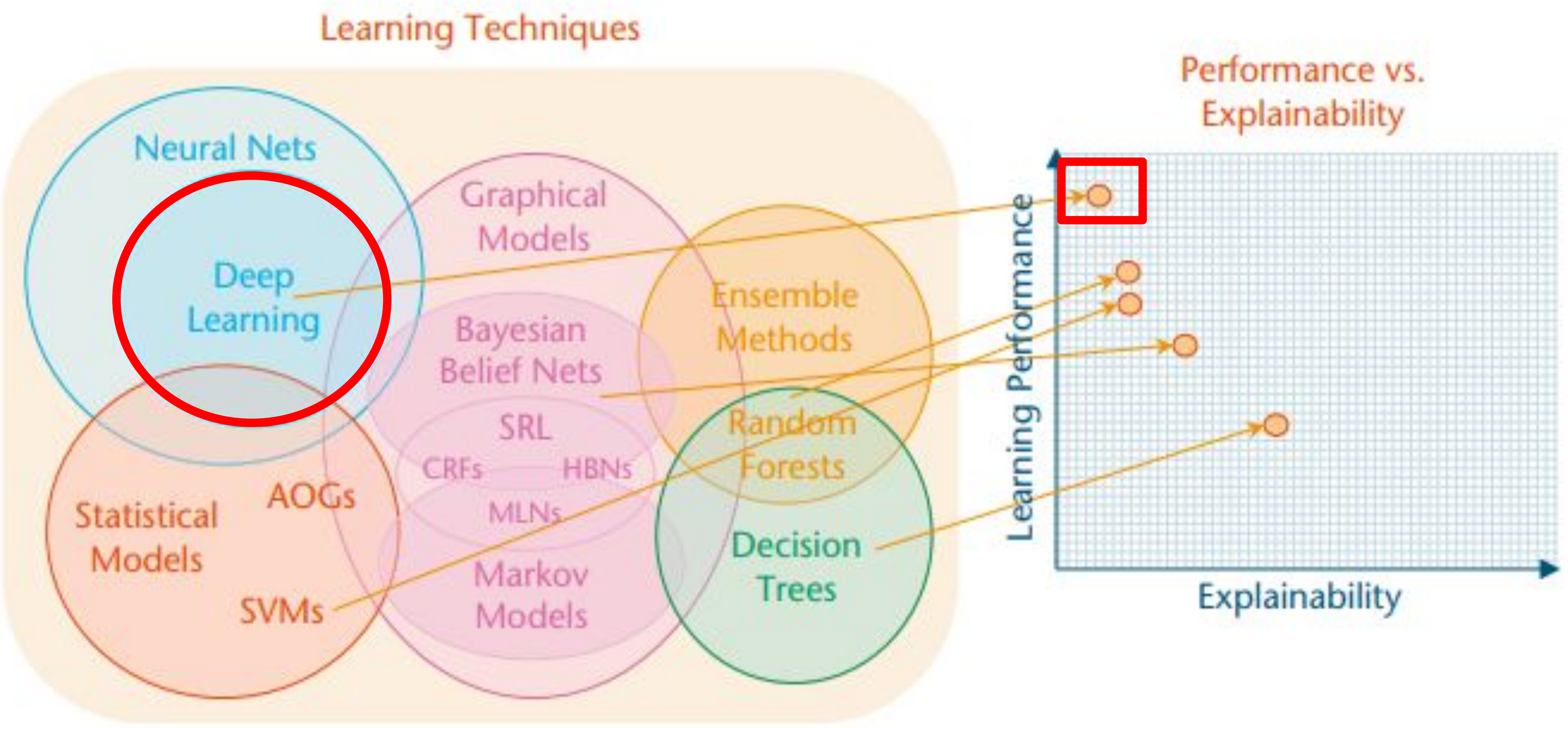
What is Explainable AI?



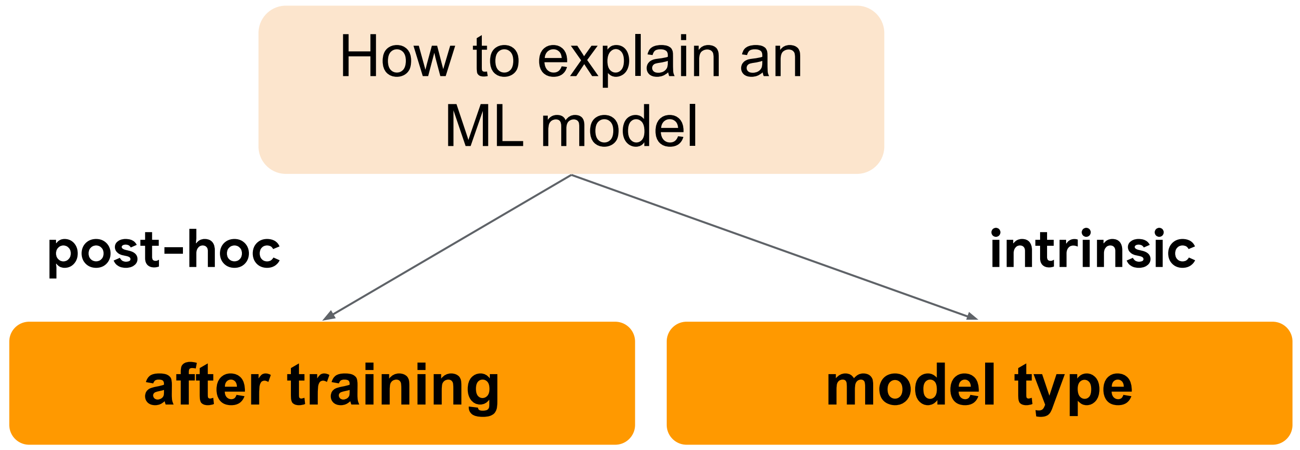
Understanding a model's behavior is critical to many tasks

- **Explain** predictions to support decision making processes
- **Debug** unexpected behavior from a model
- **Refine** modeling and data collection processes
- **Verify** that model behavior is acceptable
- **Present** the model's predictions to stakeholders

Complexity - Explainability tradeoff



Taxonomy: interpretable machine learning methods

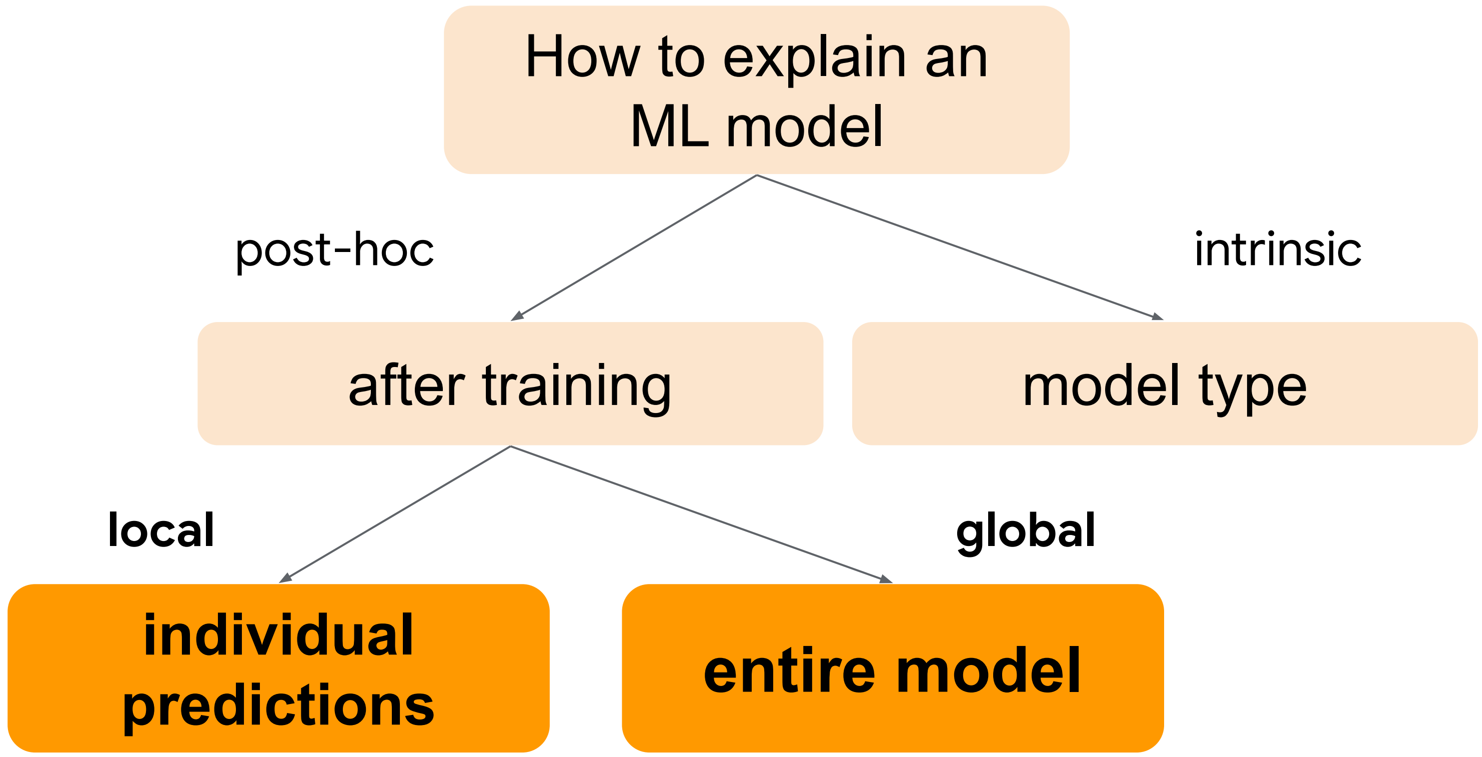


- ➔ Linear models, GLMs, GAMs
- ➔ Decision trees
- ➔ Bayesian Networks

- **Intrinsic**
  - Restricting the complexity of the machine learning model
  - Simple structure
    - e.g. decision trees, linear models

- **Post-hoc**
  - Applying methods that analyze trained models
    - e.g. Permutation feature importance
  - Can be applied to intrinsically interpretable models

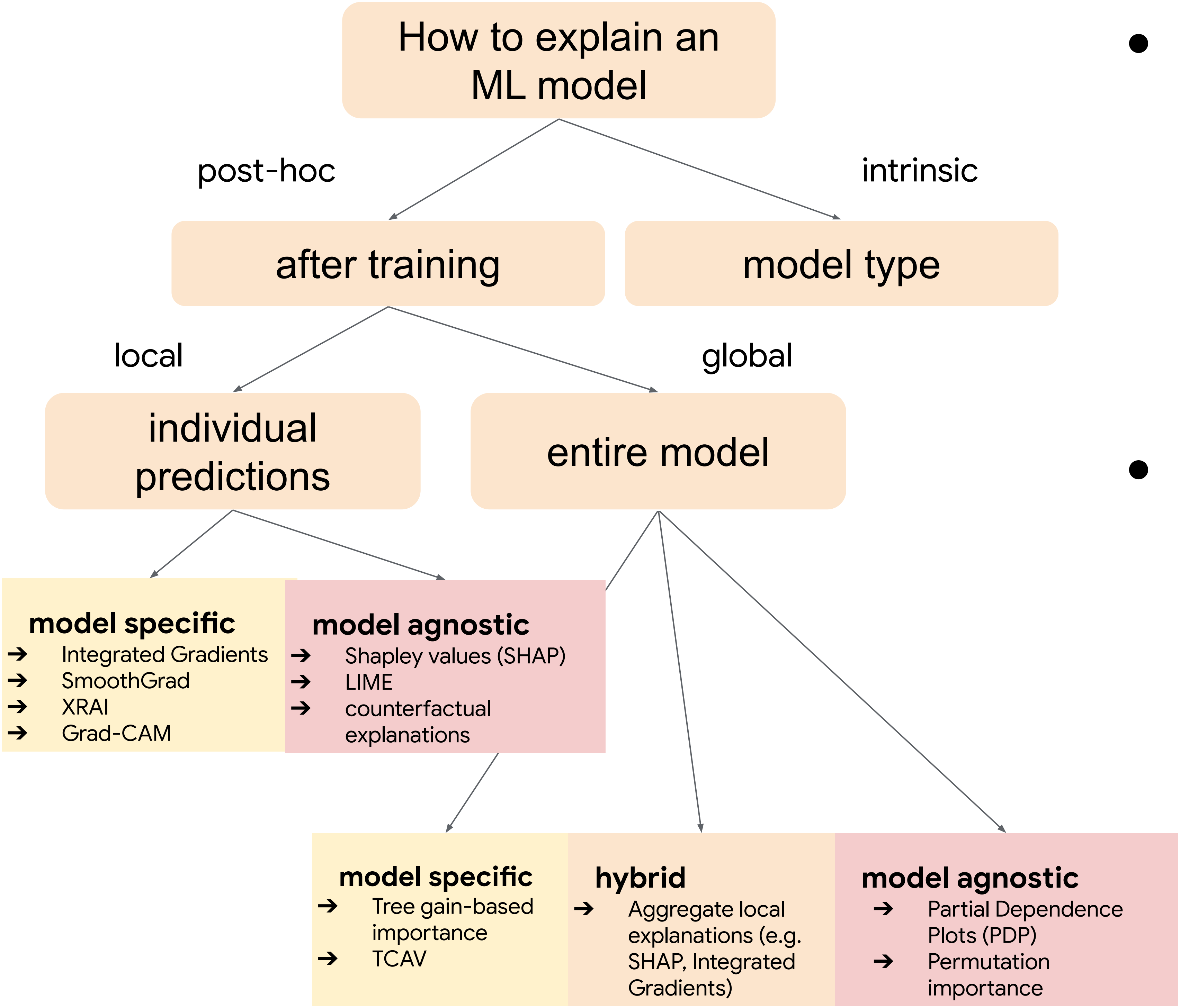
Taxonomy: interpretable machine learning methods



- **Local**
  - Interpretability of individual predictions or a small part of the model’s prediction space.
  - Higher precision but lower recall understanding of model behavior.

- **Global**
  - Aggregated, ranked contributions of input variables for the entire model’s prediction space.
  - Higher recall view of the entire model prediction space, but lower precision due to aggregations e.g. averages.

Taxonomy: interpretable machine learning methods

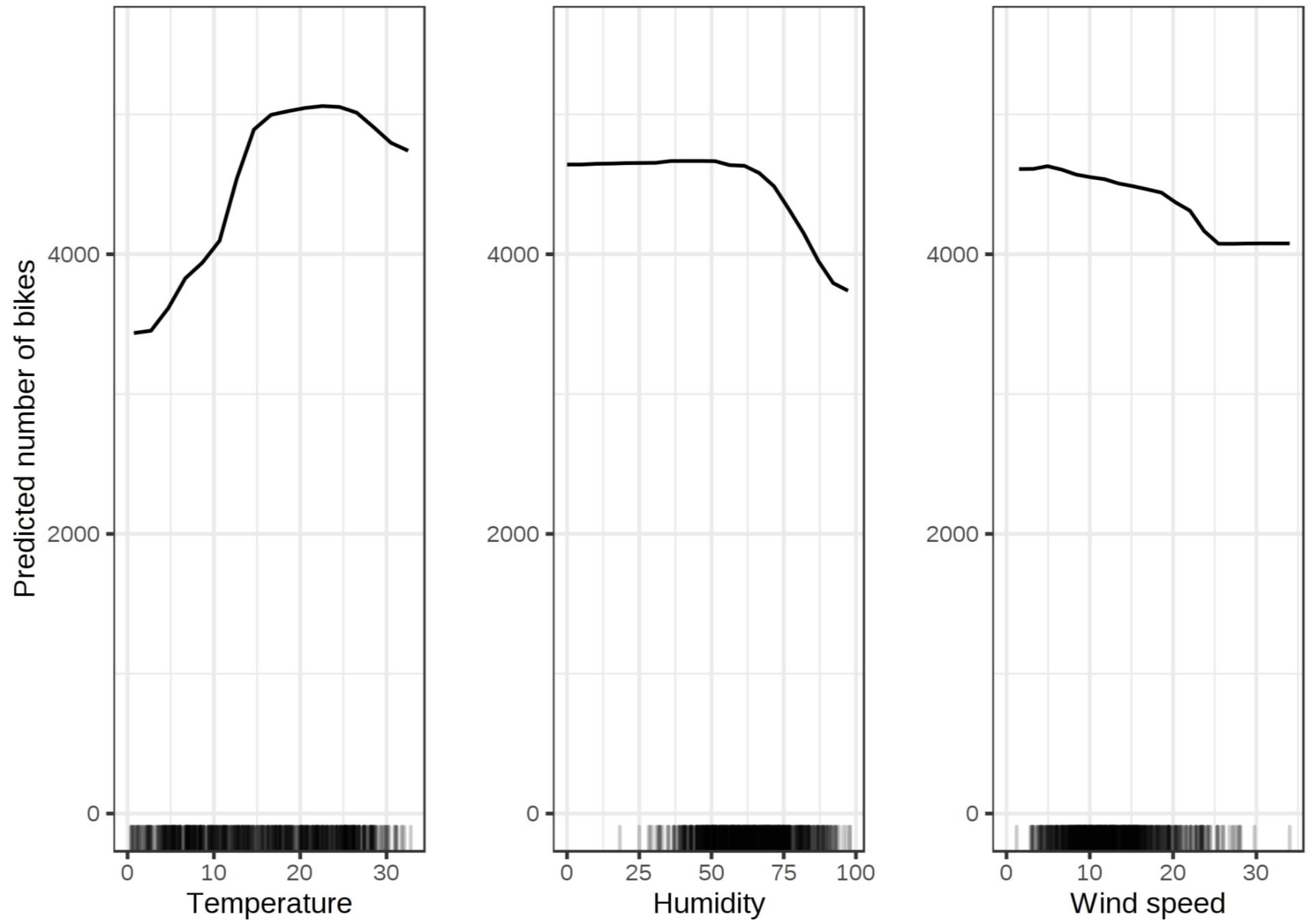


- **Model Specific**
  - Only works for specific models due to definition
    - e.g. neural network gradients, tree-based feature importances
- **Model Agnostic**
  - Portable across model definitions
    - e.g. Permutation feature importances, explainable surrogates

Questions?

Post-hoc, global, model agnostic: Partial Dependence Plots (PDPs)

- Shows the marginal effect one or two features have on the predicted outcome of a machine learning model
- Advantages
    - Very intuitive
    - Easy to implement
- Disadvantages
    - Realistic maximum number of features in a partial dependence function is two.
    - Some PDP do not show the feature distribution; Can be misleading
    - Assumption of independence



Post-hoc, global, model agnostic: Permutation Feature Importance

- Measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature.
- Advantages
  - Nice interpretation and easy implementation
  - Highly compressed global insight
  - No re-training needed
  - Takes into account all interactions with other features
- Disadvantages
  - When two features have interaction, affects both
  - When the permutation is repeated, the results might vary greatly

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

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

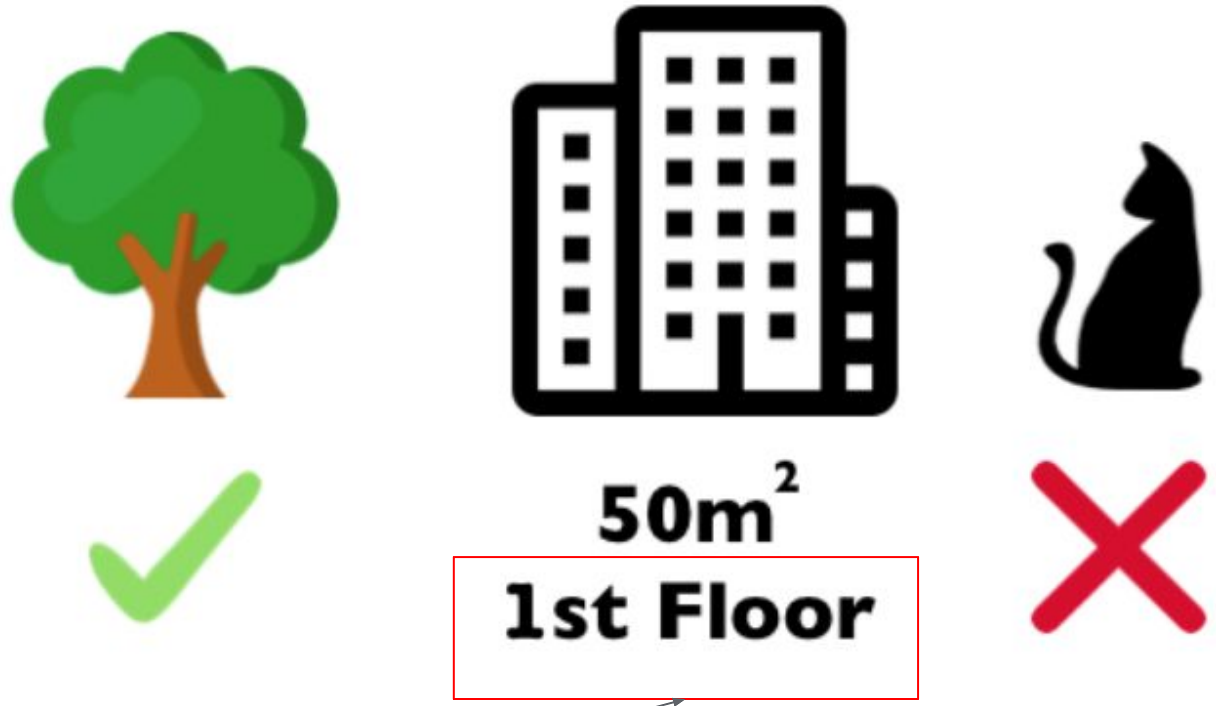
Shapley Values

- Shapley values come from an area of mathematics known as coalitional game theory.
- Precisely, the Shapley value of a feature is the average marginal contribution of a feature value across all possible coalitions.
- More intuitively, the feature values enter a room in random order. All feature values in the room participate in the game (= contribute to the prediction). The Shapley value of a feature value is the average change in the prediction that the coalition already in the room receives when the feature value joins them.

How much does each feature contribute?

How much does `cat_banned` contribute?

Consider coalitions of features



drawn randomly

Coalition

- `park_nearby`
- `size_50`

+

- `cat_banned`

How much does `cat_banned` contribute?

Consider coalitions of features

Coalition

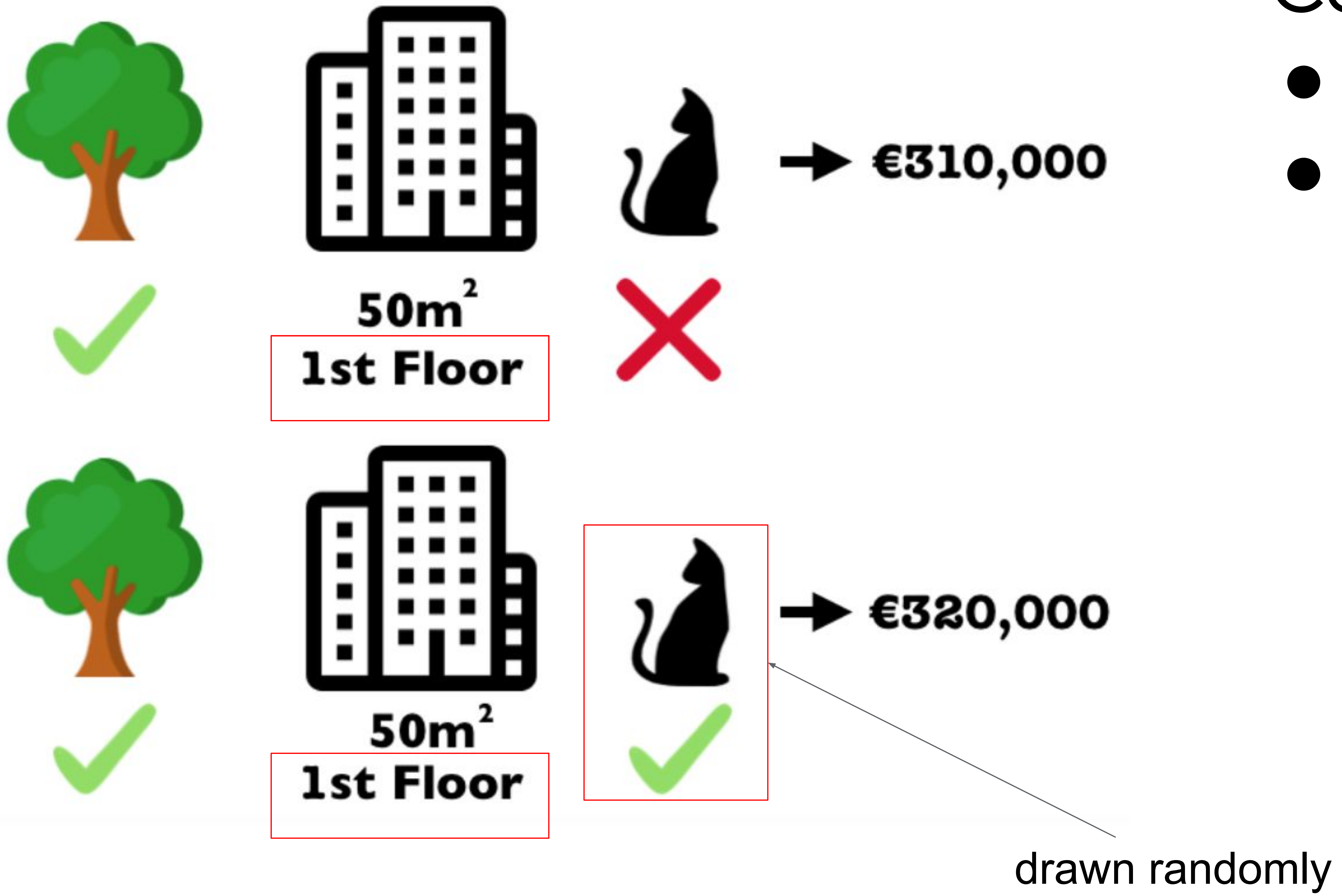
- `park_nearby`
- `size_50`

+

- `cat_banned`

How much does `cat_banned` contribute?

Consider coalitions of features

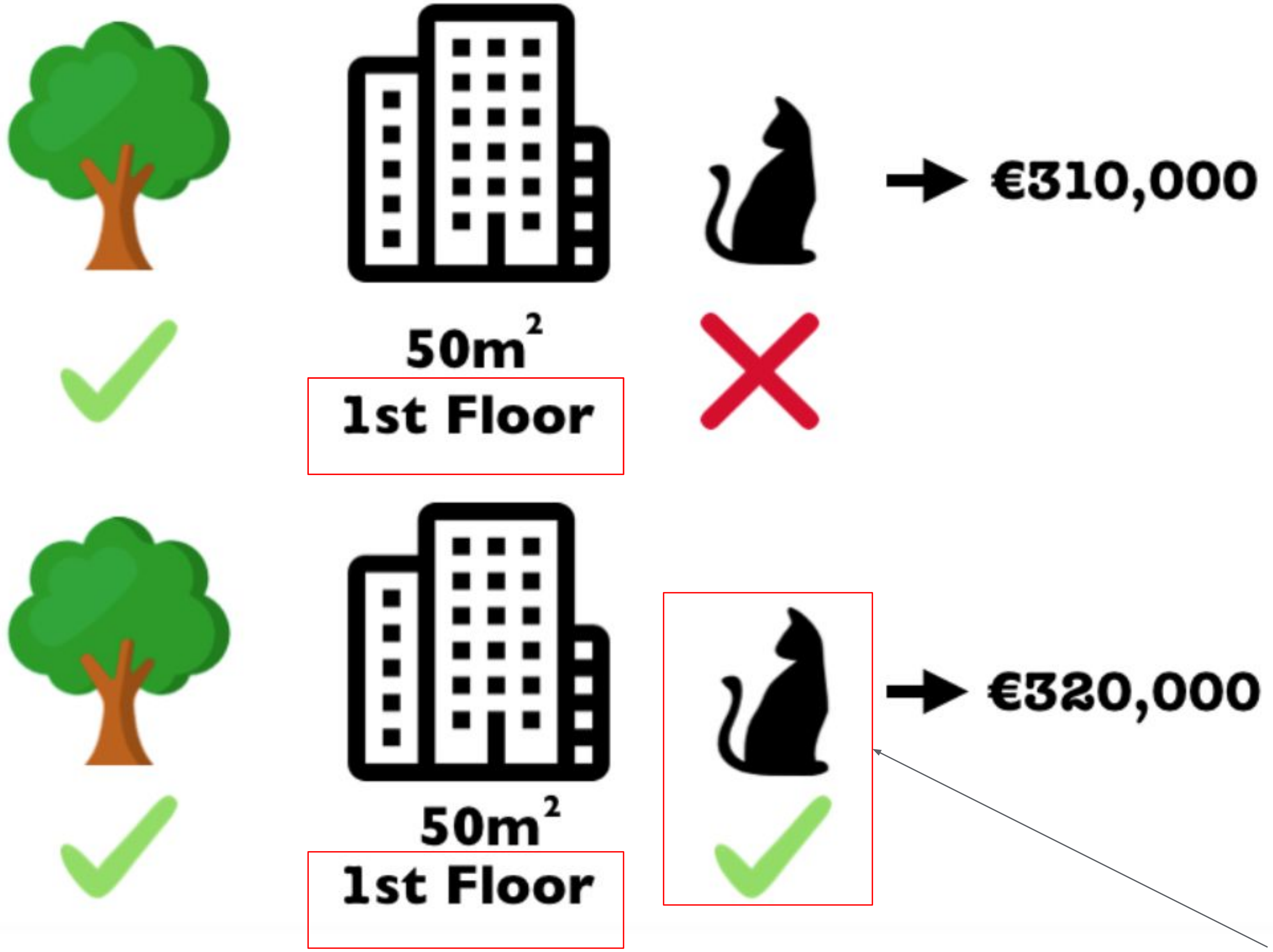


Coalition

- `park_nearby`
- `size_50`

How much does cat_banned contribute?

Consider coalitions of features



drawn randomly

Coalition

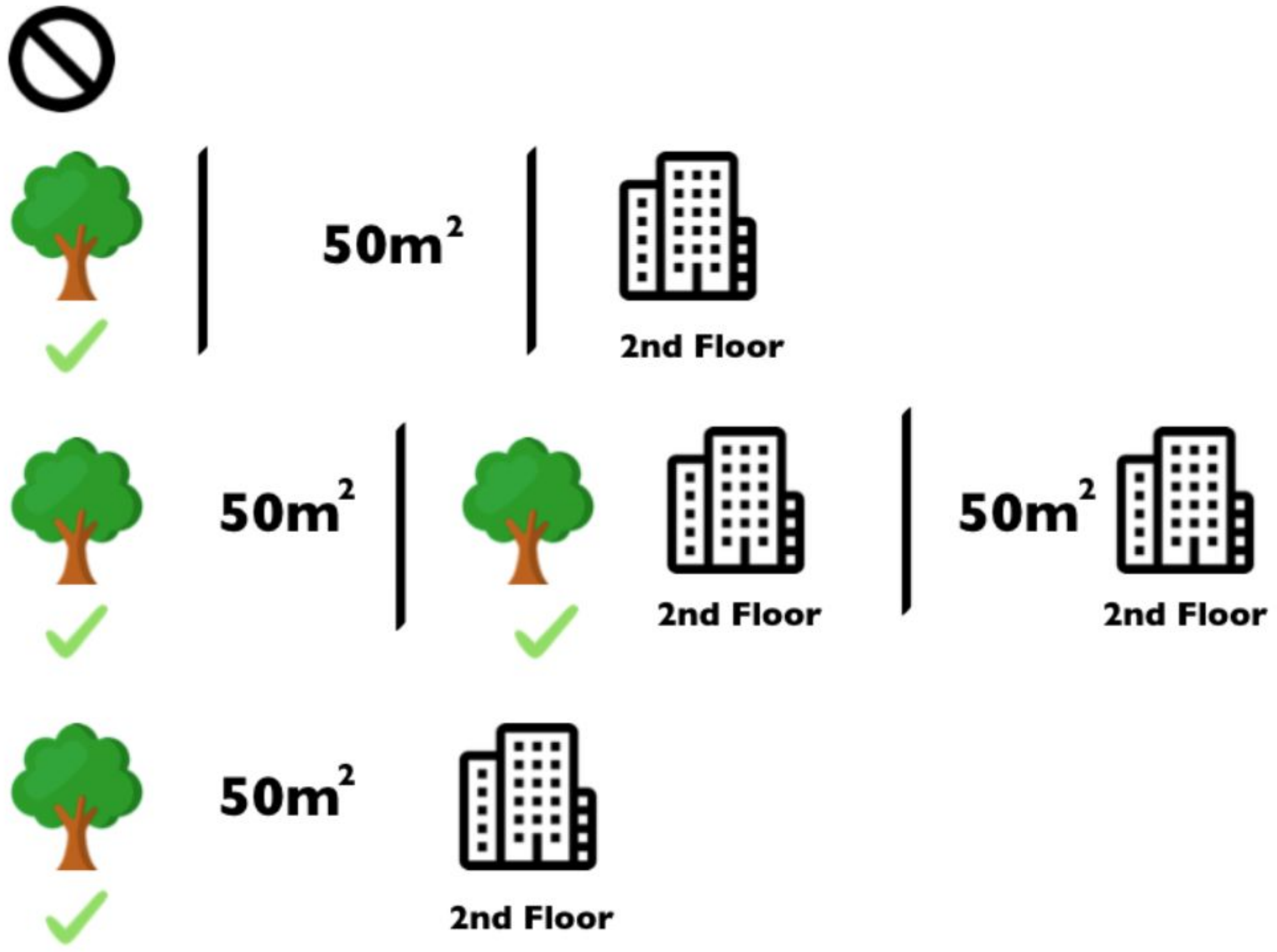
- park_nearby
- size_50

contribution of cat_banned is €10,000

How much does `cat_banned` contribute?

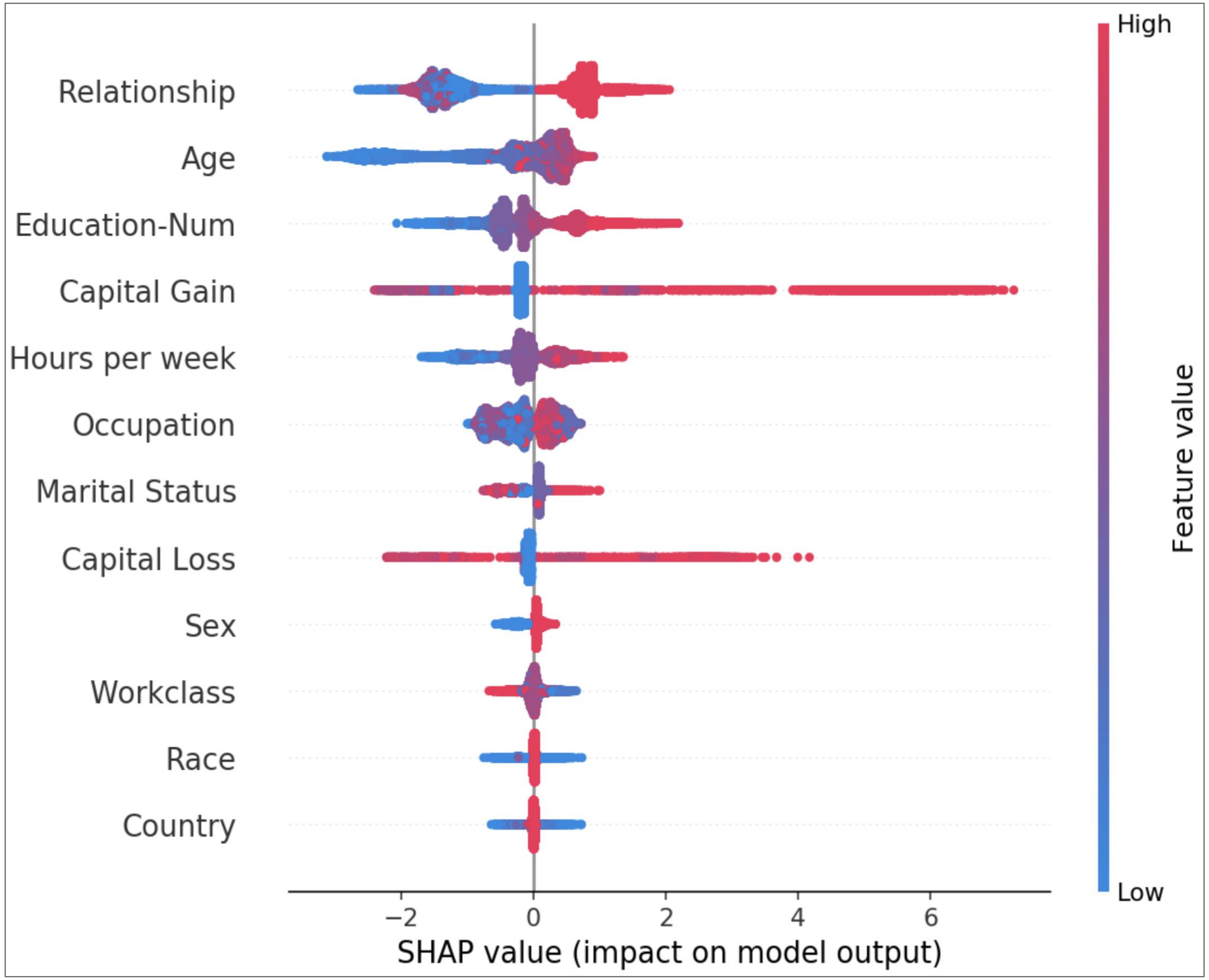
Consider *all* coalitions of features

- No feature values
- park-nearby
- size-50
- floor-2nd
- park-nearby+size-50
- park-nearby+floor-2nd
- size-50+floor-2nd
- park-nearby+size-50+floor-2nd



SHAP Values: Consistent Feature Attributions

- Exact computation of the Shapley value is computationally expensive because there are 2^k possible coalitions of the feature values and the "absence" of a feature has to be simulated by drawing random instances.
- SHAP library approximates the the Shapley values.
- A variation of this method called sampled Shapley is available in Explainable AI on Google Cloud.
- Paper, Github



Text and picture taken from Interpretable Machine Learning with XGBoost, Apr 17, 2018.

Agenda

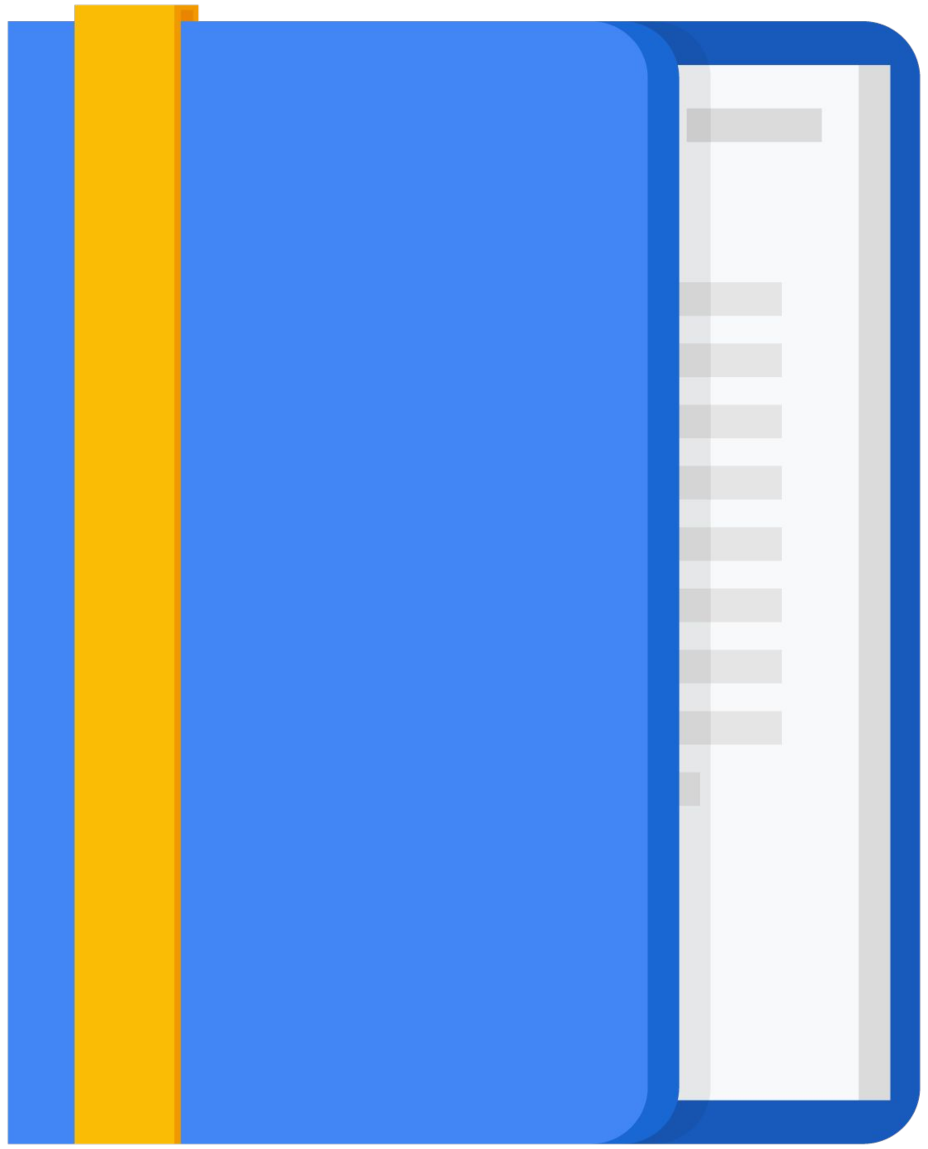
What is Explainable AI?

Interpretable ML methods

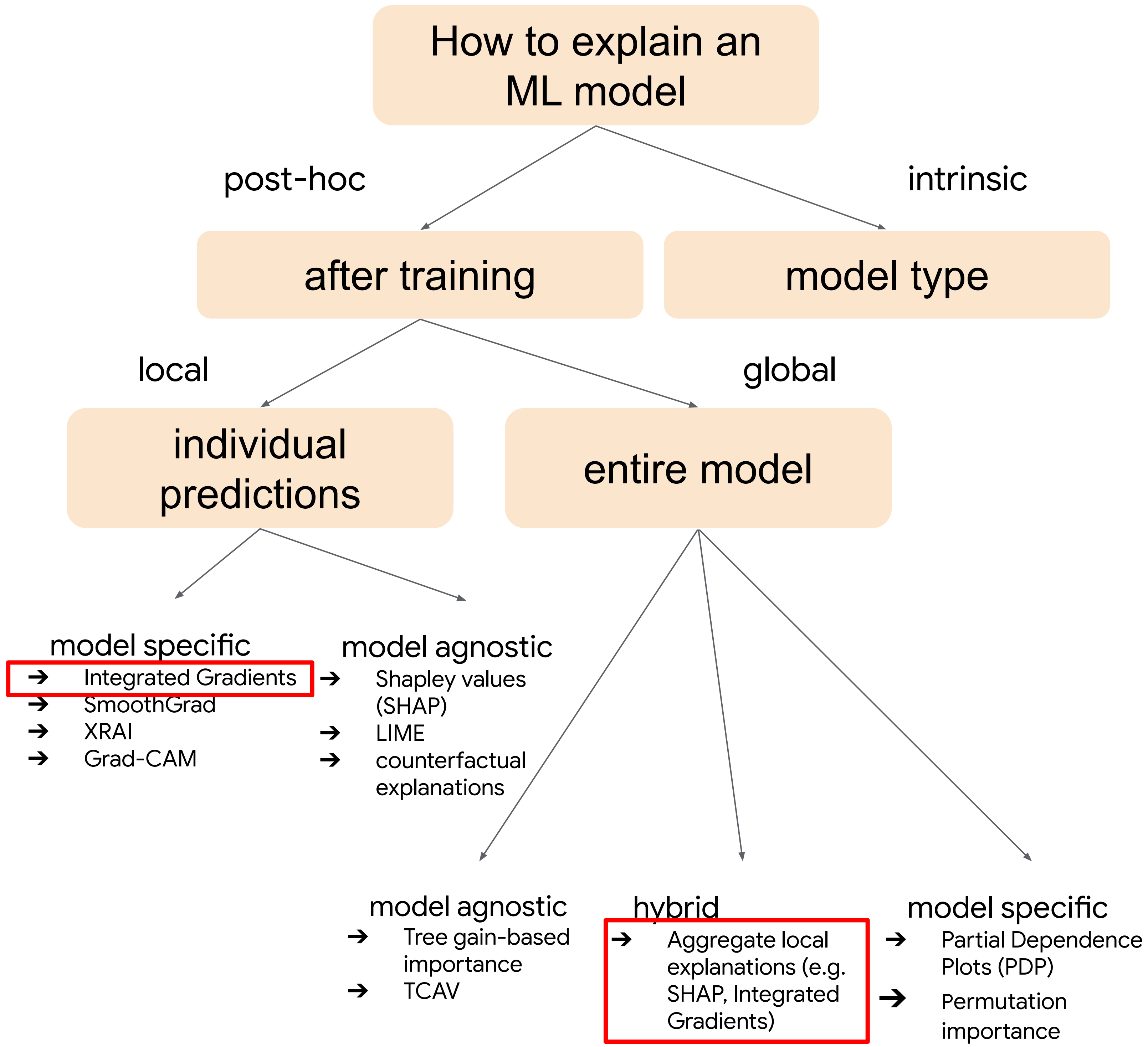
Deepdive: Integrated Gradients (IG)

Picking baselines and future research directions

Explainable AI on Google Cloud



IG in the interpretable ML method taxonomy



Gradient-based Attribution

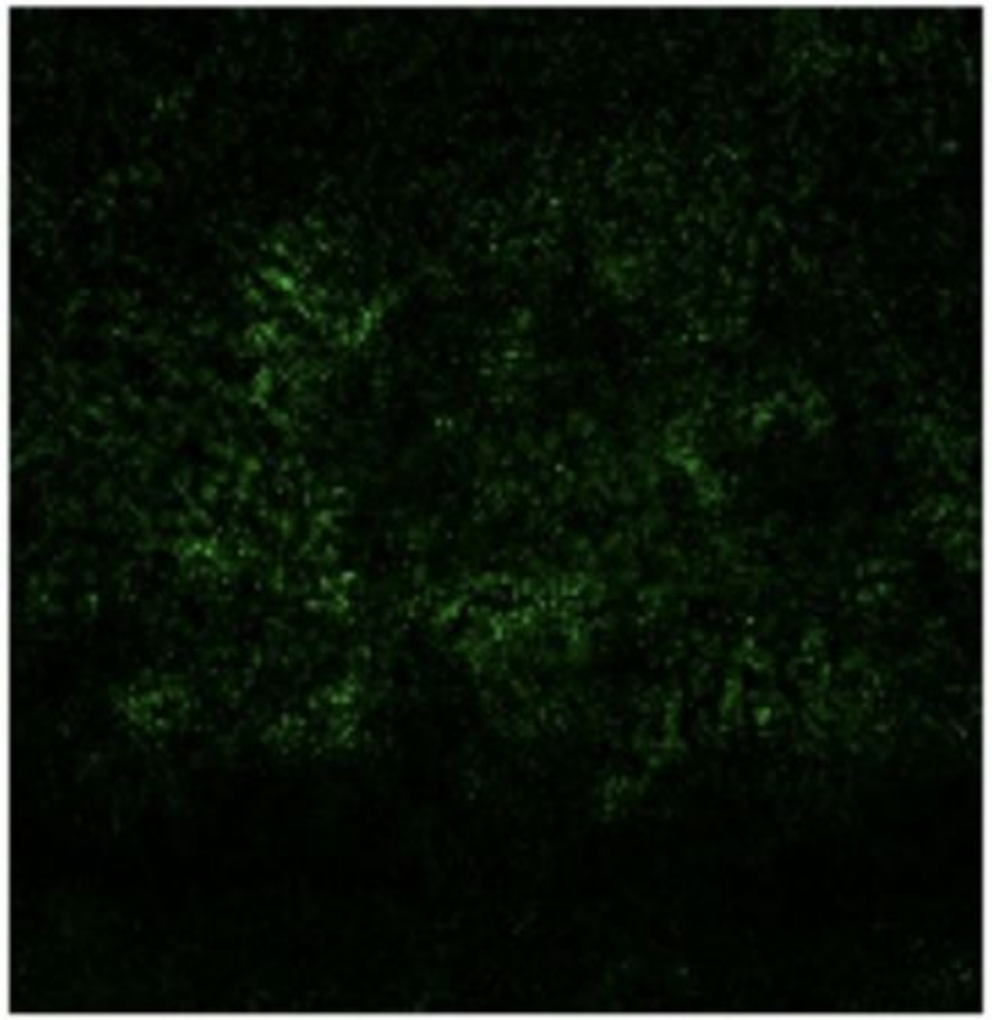
Create attribution using gradient of the output wrt each base input feature

$$\text{attribution for feature } x_i = x_i \frac{\partial y}{\partial x_i}$$

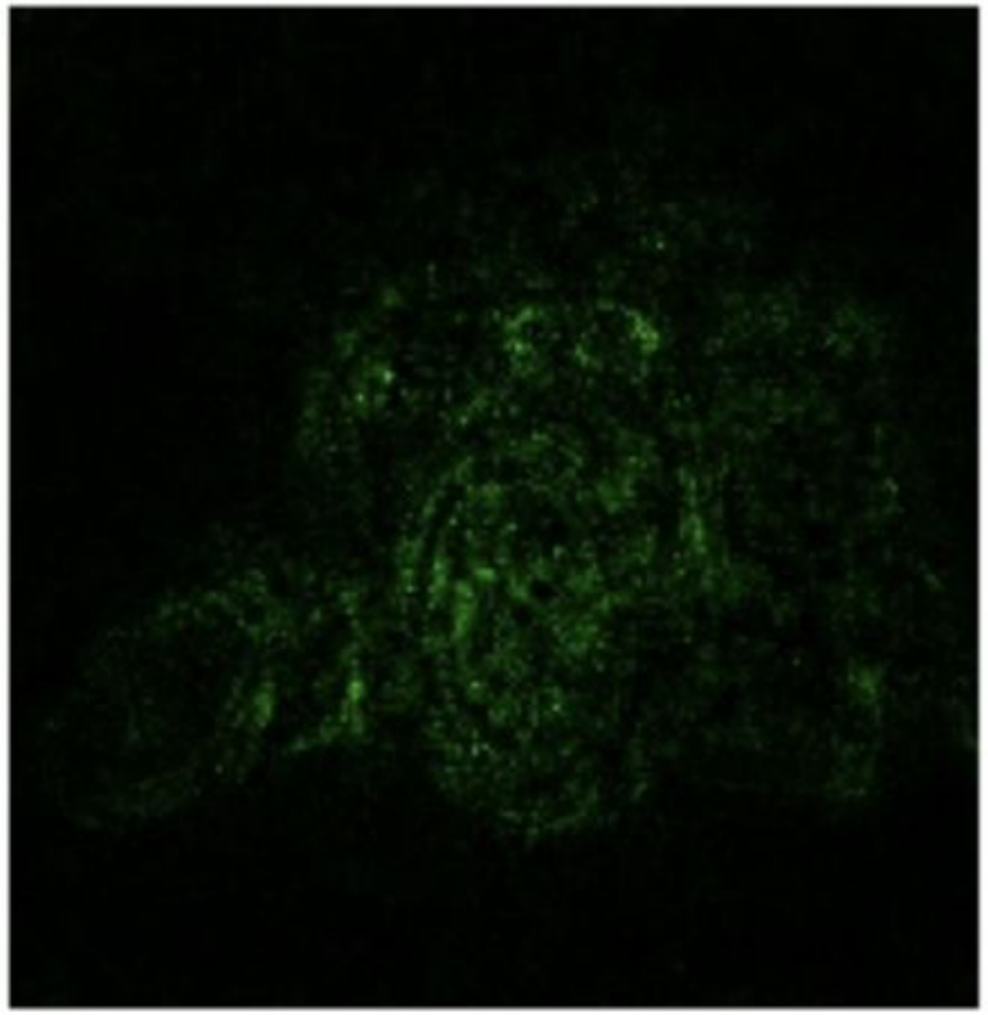
- same as feature weights for linear models
- 1st order approximation for non-linear models
- use (normalized) attribution as mask/window over image

Why not just gradients? Saturation

Original Image
(Input)

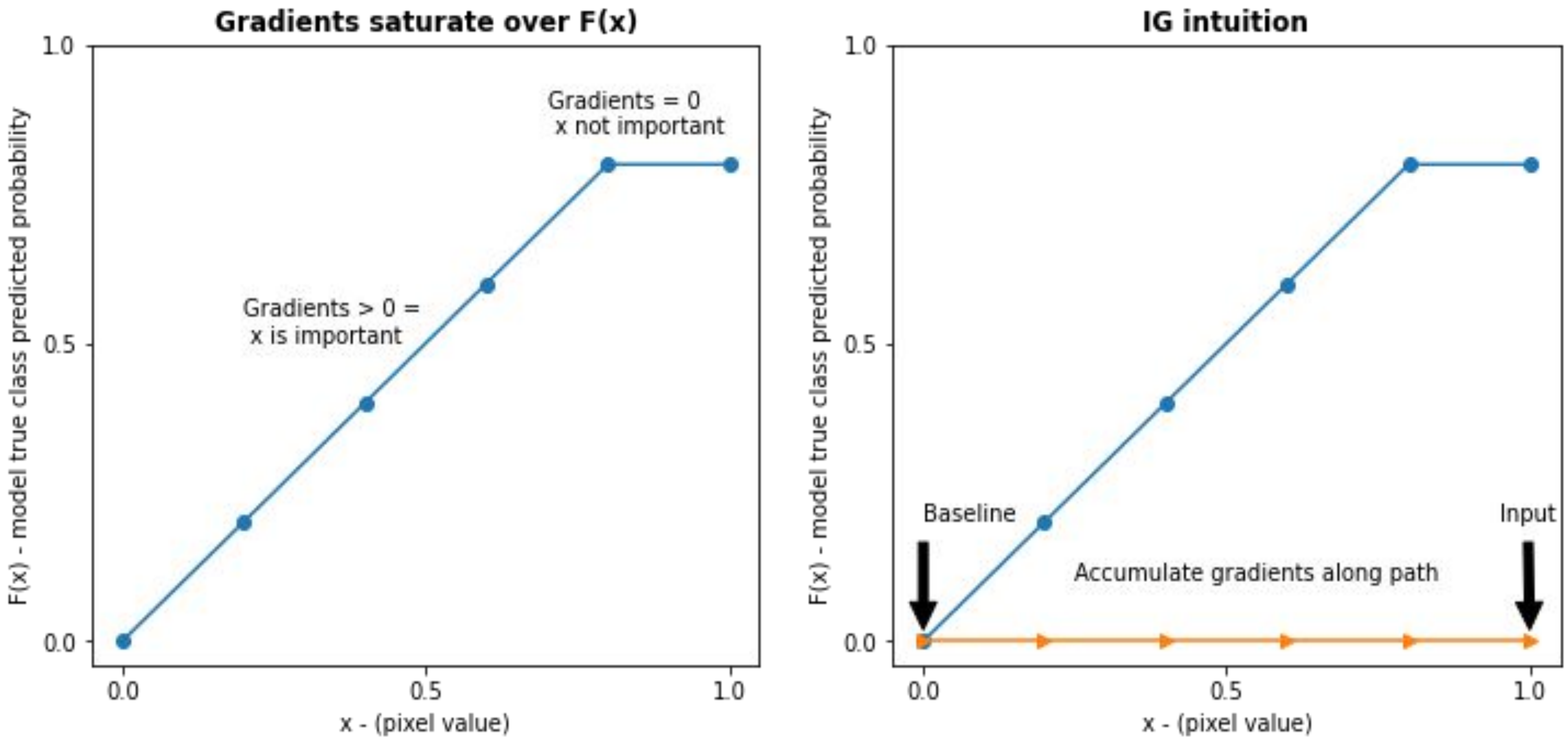


Vanilla
Gradients

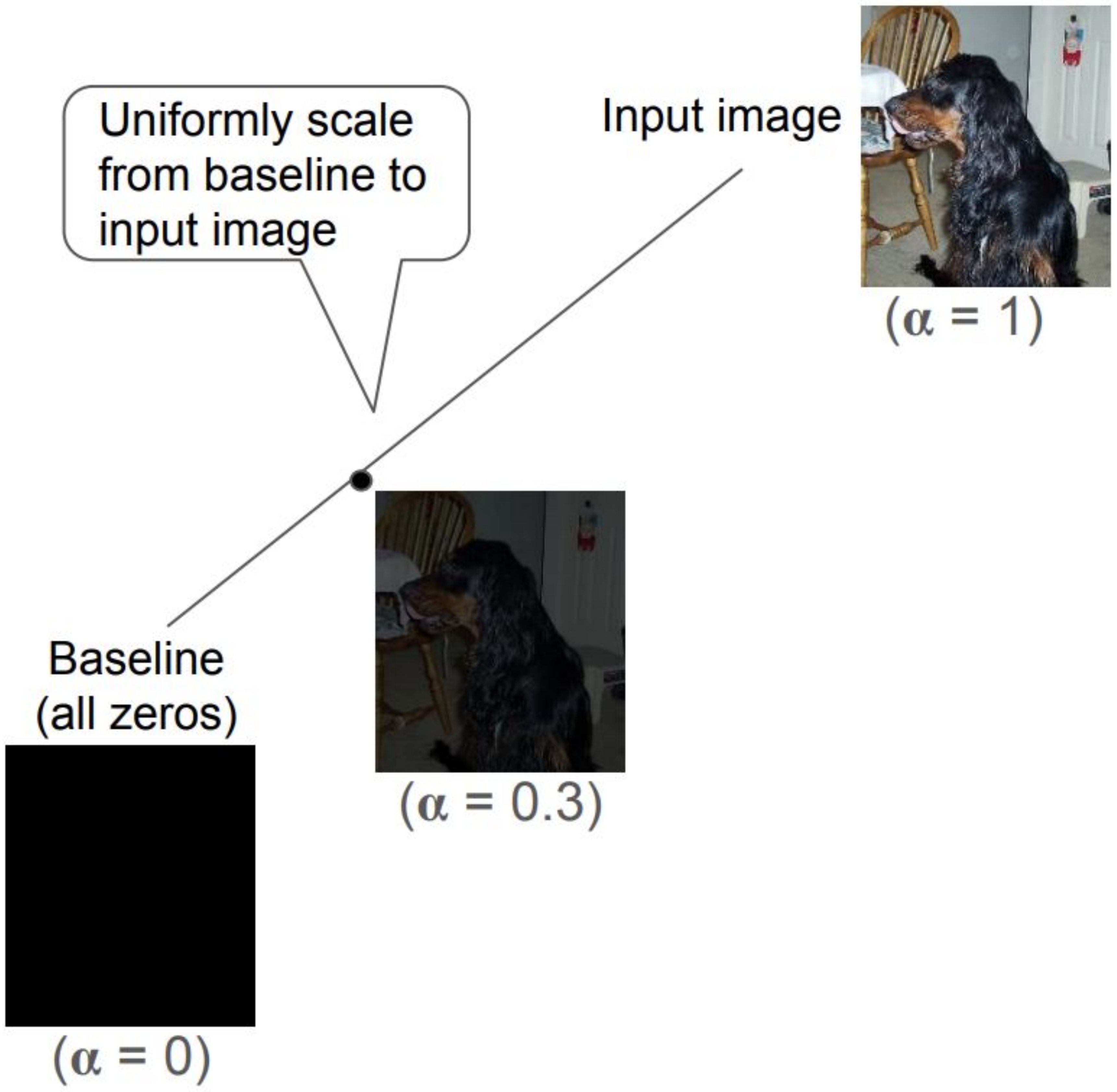


Integrated
Gradients

Intuition: how IG solves the gradient saturation problem



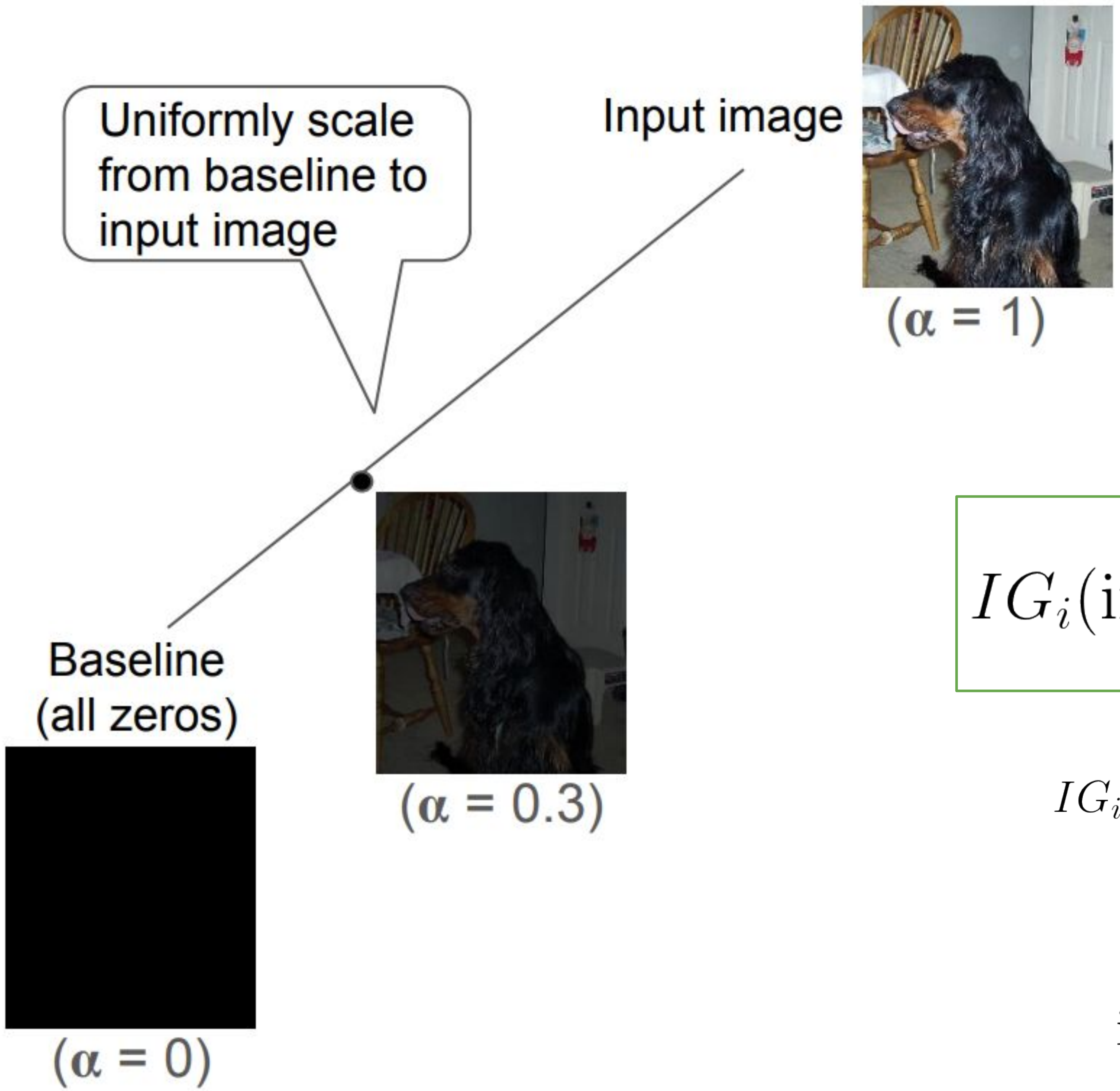
Integrated Gradients



Construct a sequence of images interpolating from baseline (black) to the actual image

Average the gradients across these images

Integrated Gradients



Construct a sequence of images interpolating from baseline (black) to the actual image

Average the gradients across these images

$$IG_i(\text{image}) = \text{image}_i \int_0^1 \nabla F_i(\alpha \cdot \text{image})d\alpha$$

$IG_i(\text{image})$ is the integrated gradient wrt the *i*th pixel i.e. the attribution for the *i*th pixel

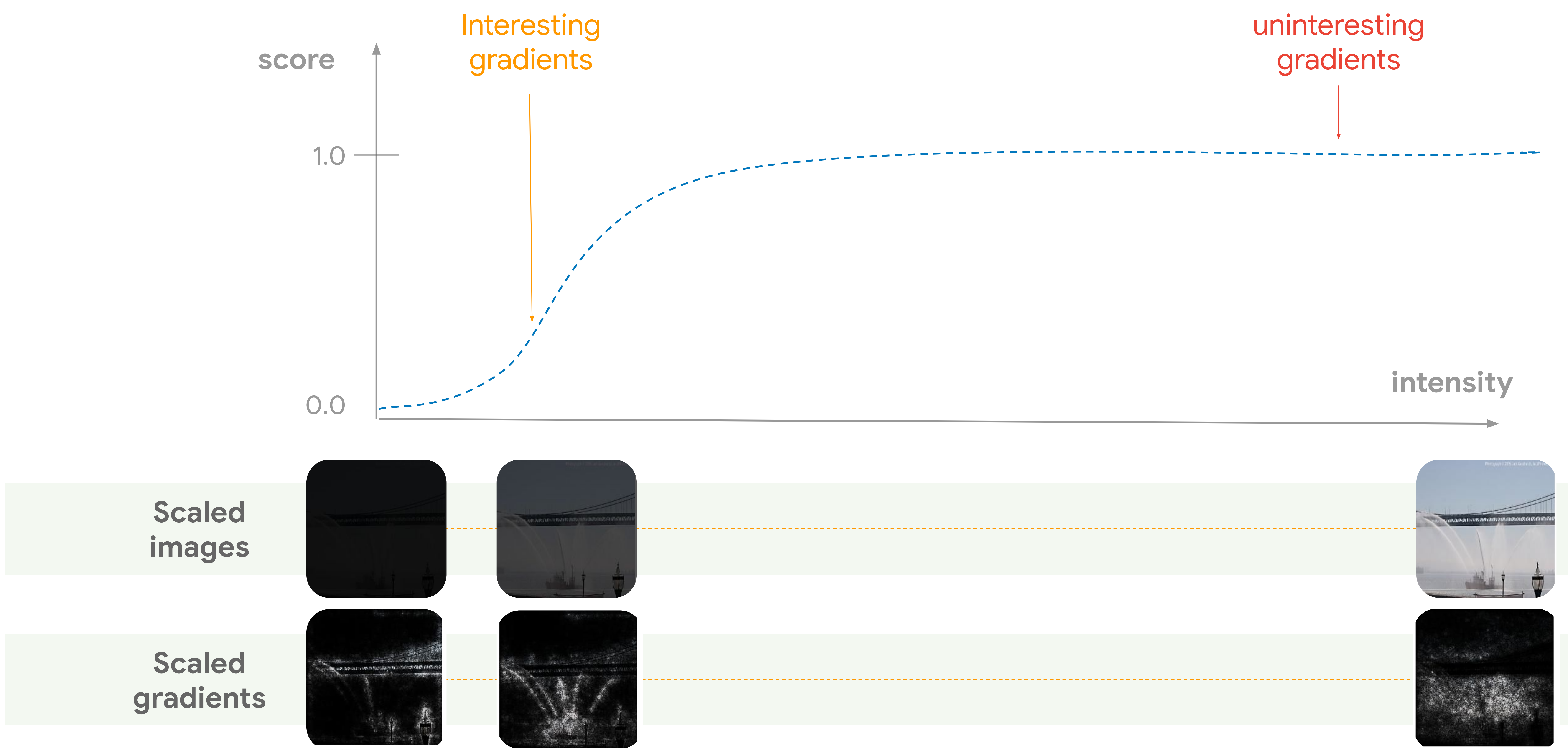
F is the prediction function for the label

image_i is the intensity of the *i*th pixel

An example of Integrated Gradients

The label for this image was “fire boat”.

Let’s see what integrated gradients will tell us.


score
Interesting
gradients
uninteresting
gradients
1.0
0.0
intensity
Scaled
images
Scaled
gradients

Image

Explanation

Questions?

Agenda

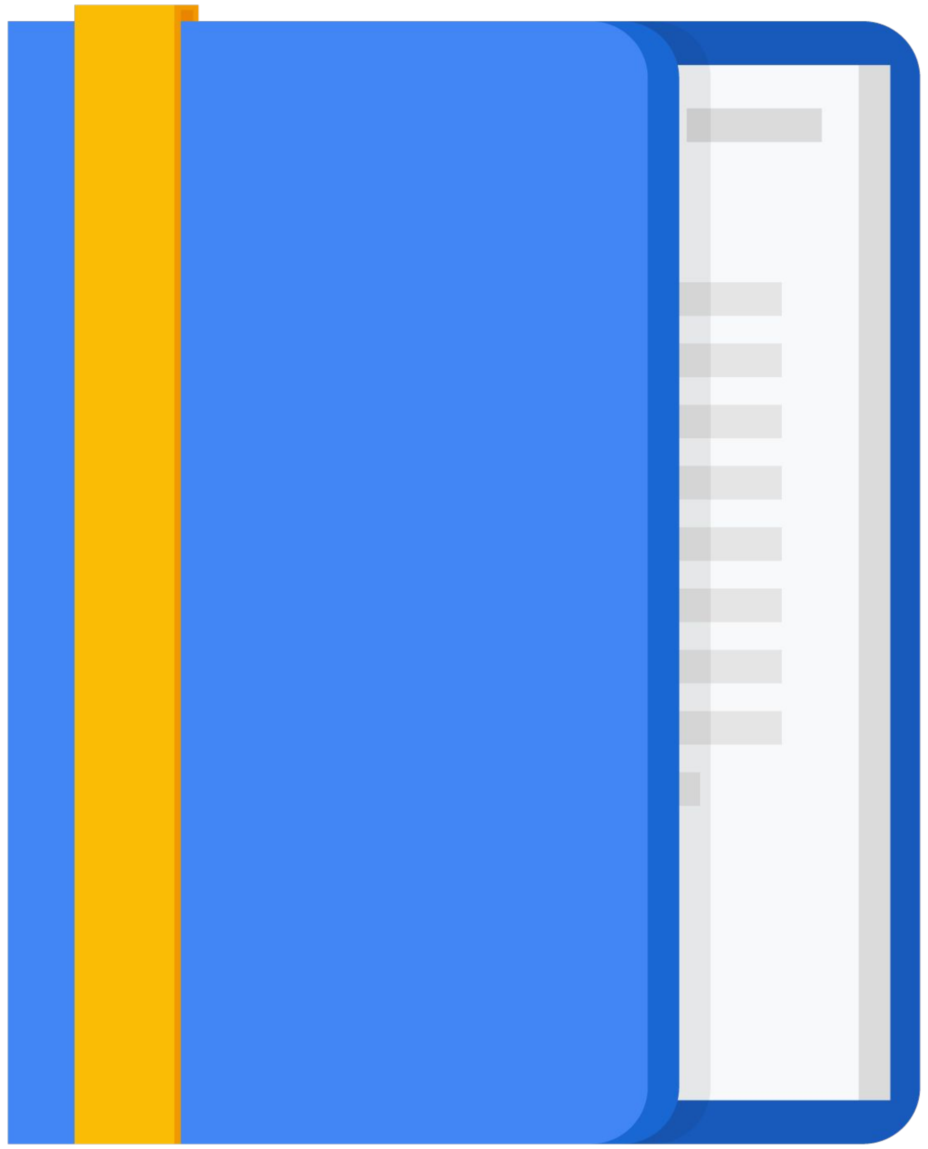
What is Explainable AI?

Interpretable ML methods

Deepdive: Integrated Gradients (IG)

Picking baselines and future research directions

Explainable AI on Google Cloud



IG has a baseline selection problem

- Integrated Gradients requires a baseline image.
- The default choice for the baseline is an informative black image.
- What are the implications of such choice?

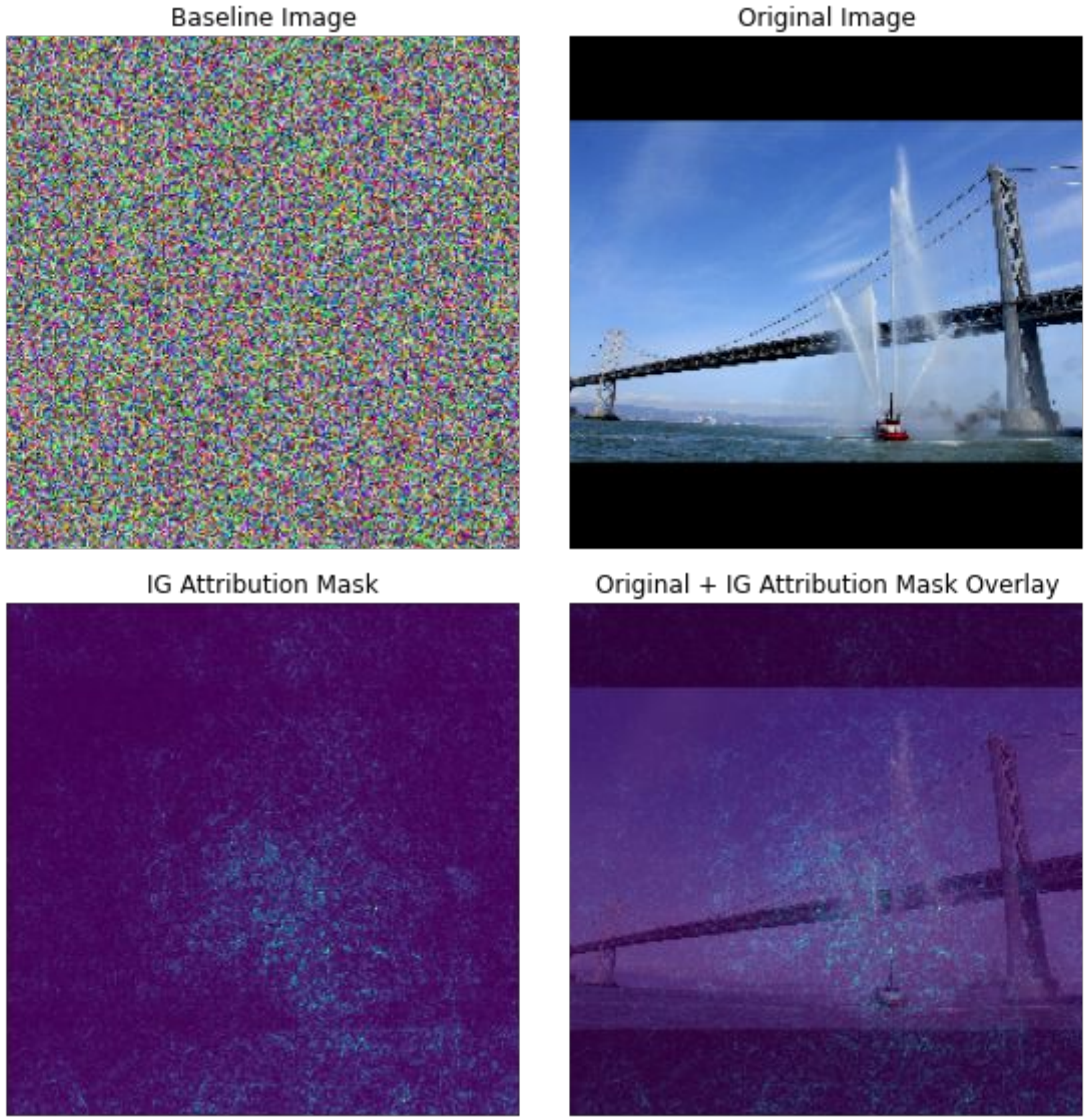
$$\text{IntegratedGrads}_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} \, d\alpha$$

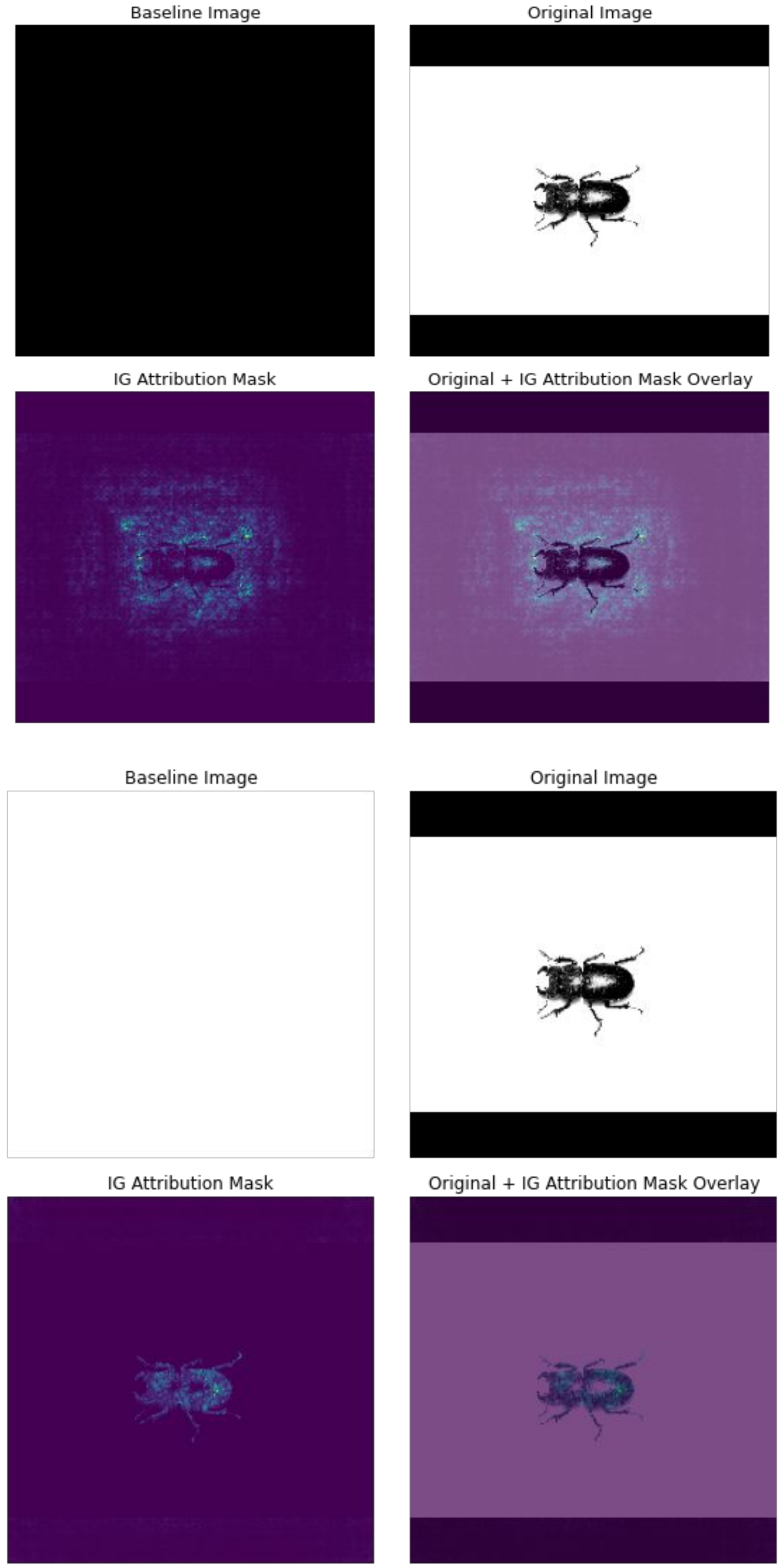
input image pixel

base image pixel
(usually eq. 0)

Black pixels never get any attribution !!!

How to select a good baseline?

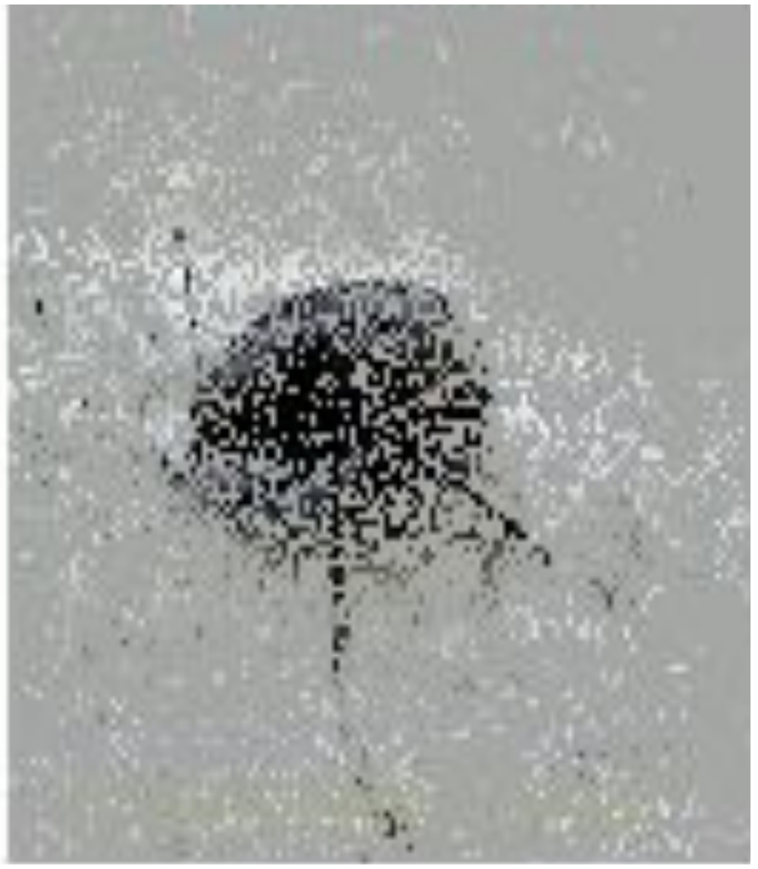




XRAI (better attributions through regions): improving upon IG

Original image

Integrated Gradients

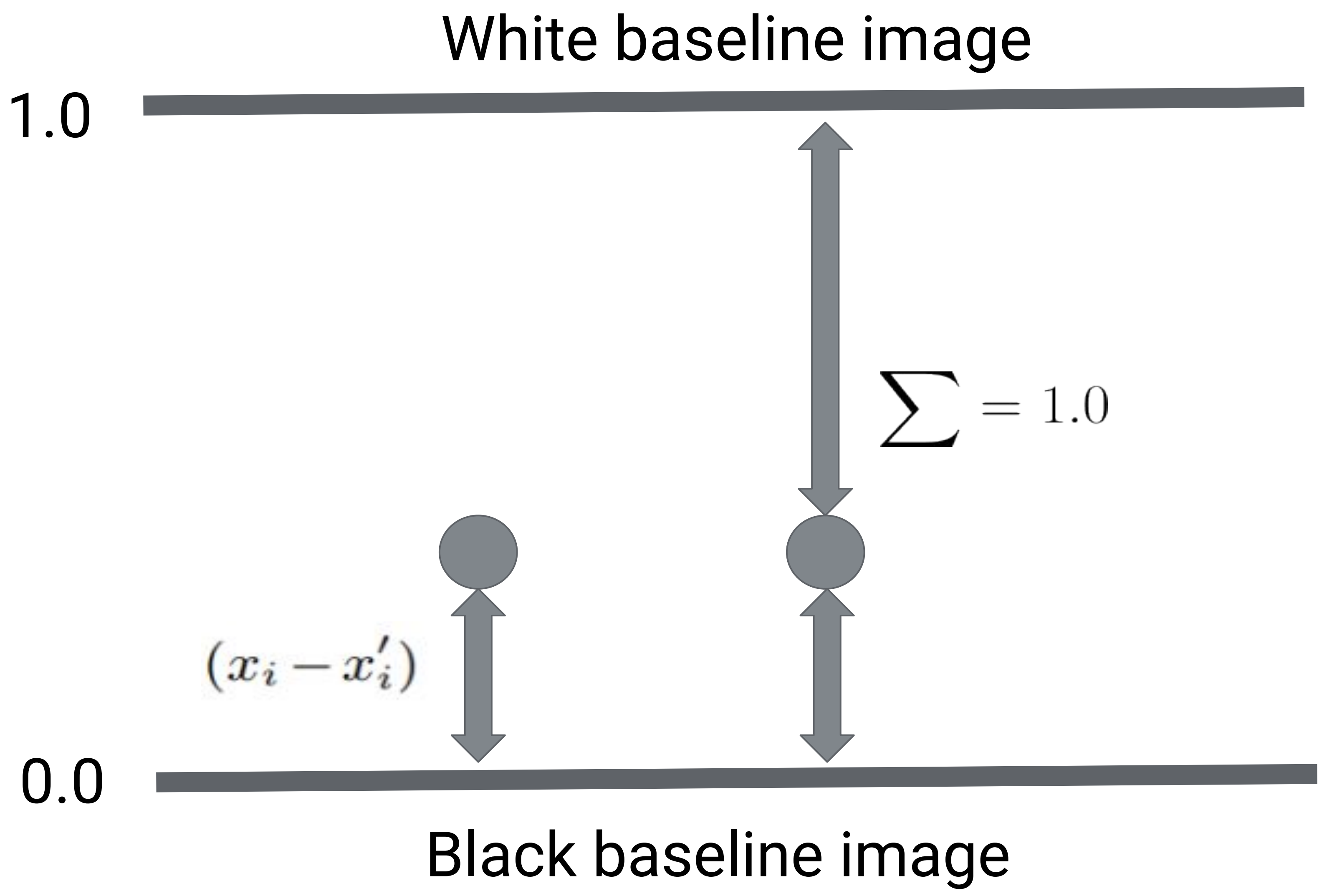


XRAI

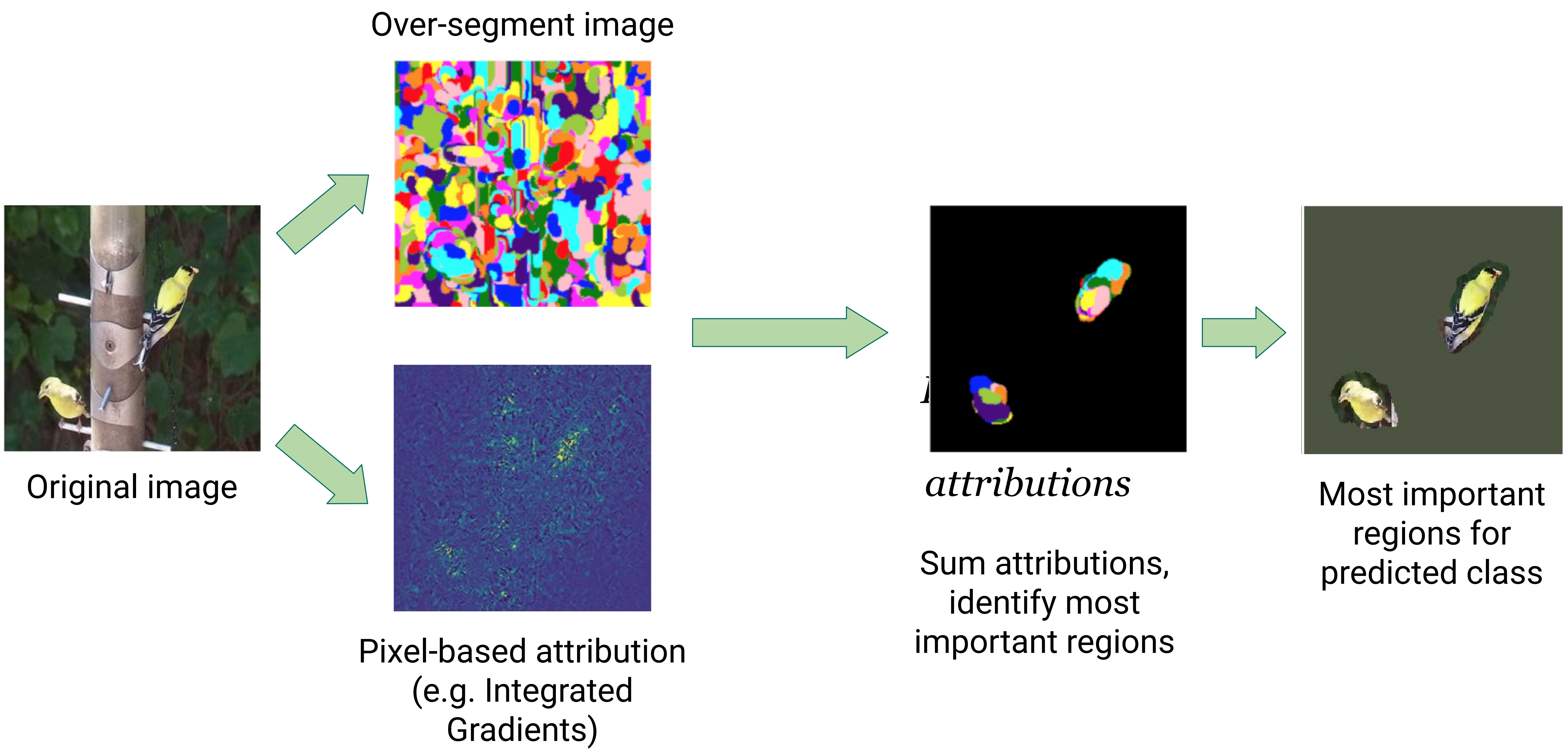
XRAI Uses Black AND White baselines

- Use two baseline images: completely black and completely white.

$$\text{IntegratedGrads}_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} \, d\alpha$$



XRAI: Region-based image attributions



Predictions on the future of Explainable AI

1. Will become a standard component of ML pipelines
2. Model agnostic interpretability methods will be the focus
3. Will converge with causal inference to improve ML reliability
4. Explanations will add uncertainty estimates to improve interpretation

Agenda

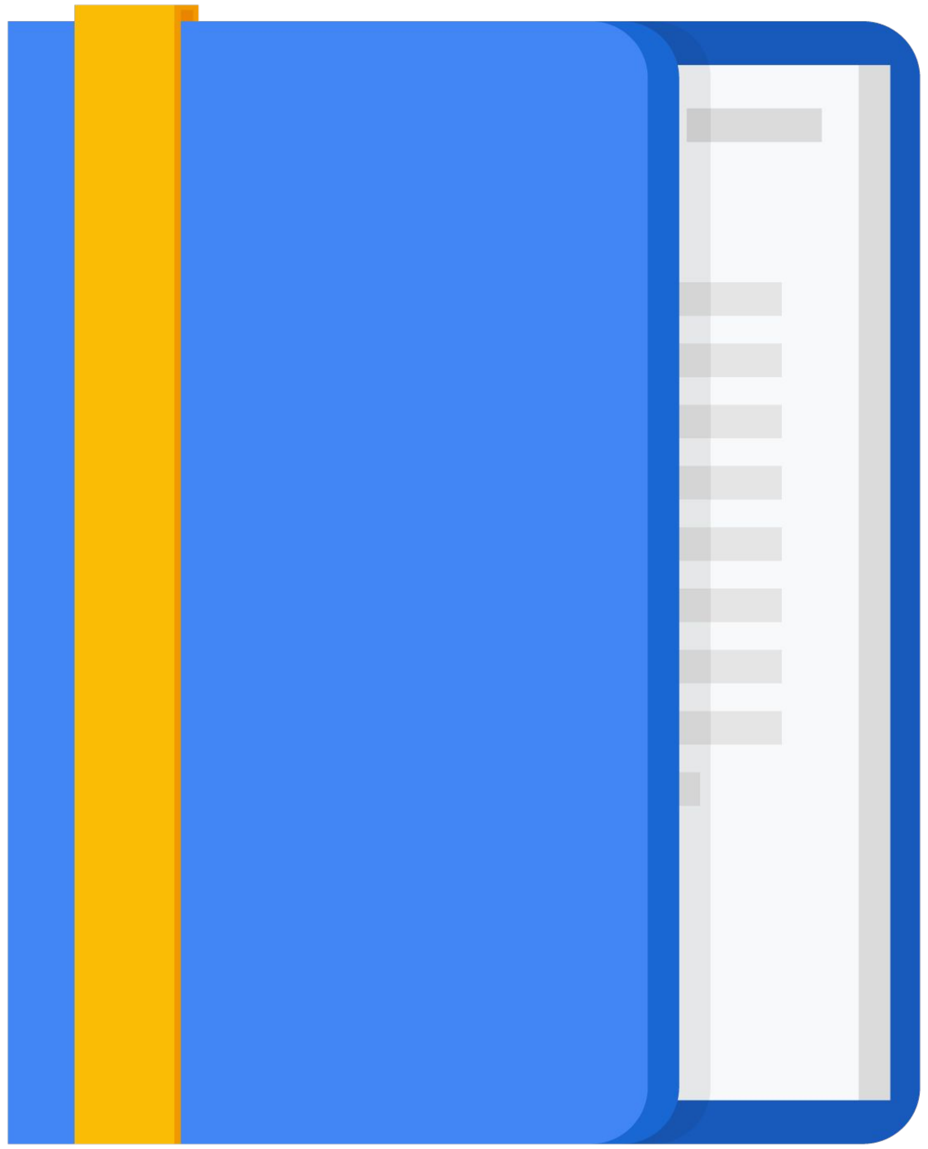
What is Explainable AI?

Interpretable ML methods

Deepdive: Integrated Gradients (IG)

Picking baselines and future research directions

Explainable AI on Google Cloud

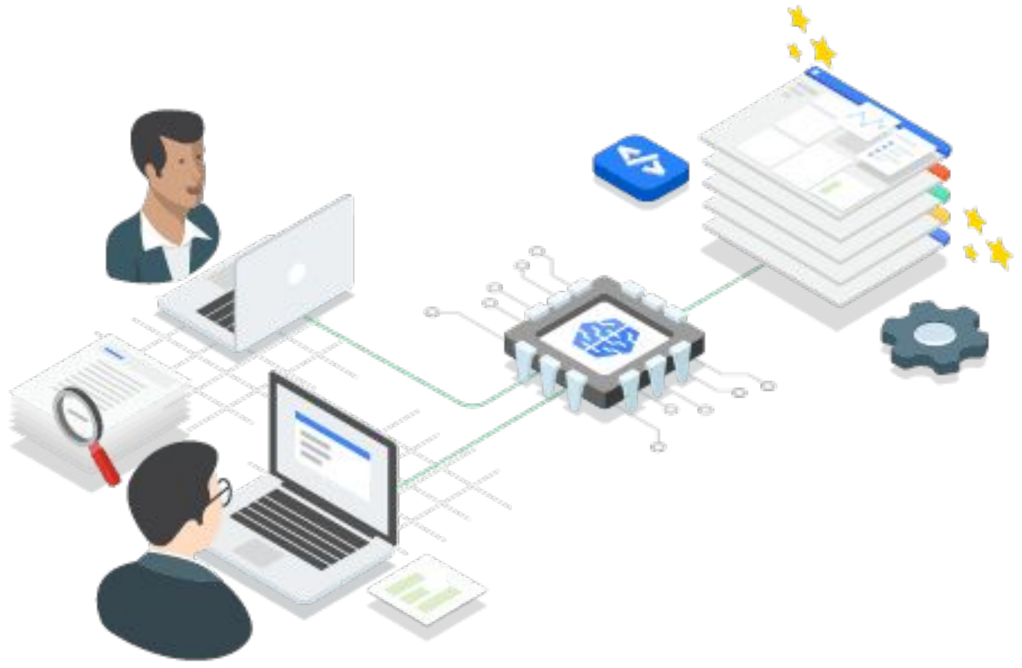


Incorporating Explainable AI into your ML workflow on Google Cloud

- Explainable AI is a set of tools and frameworks to help you develop interpretable and inclusive ML models.

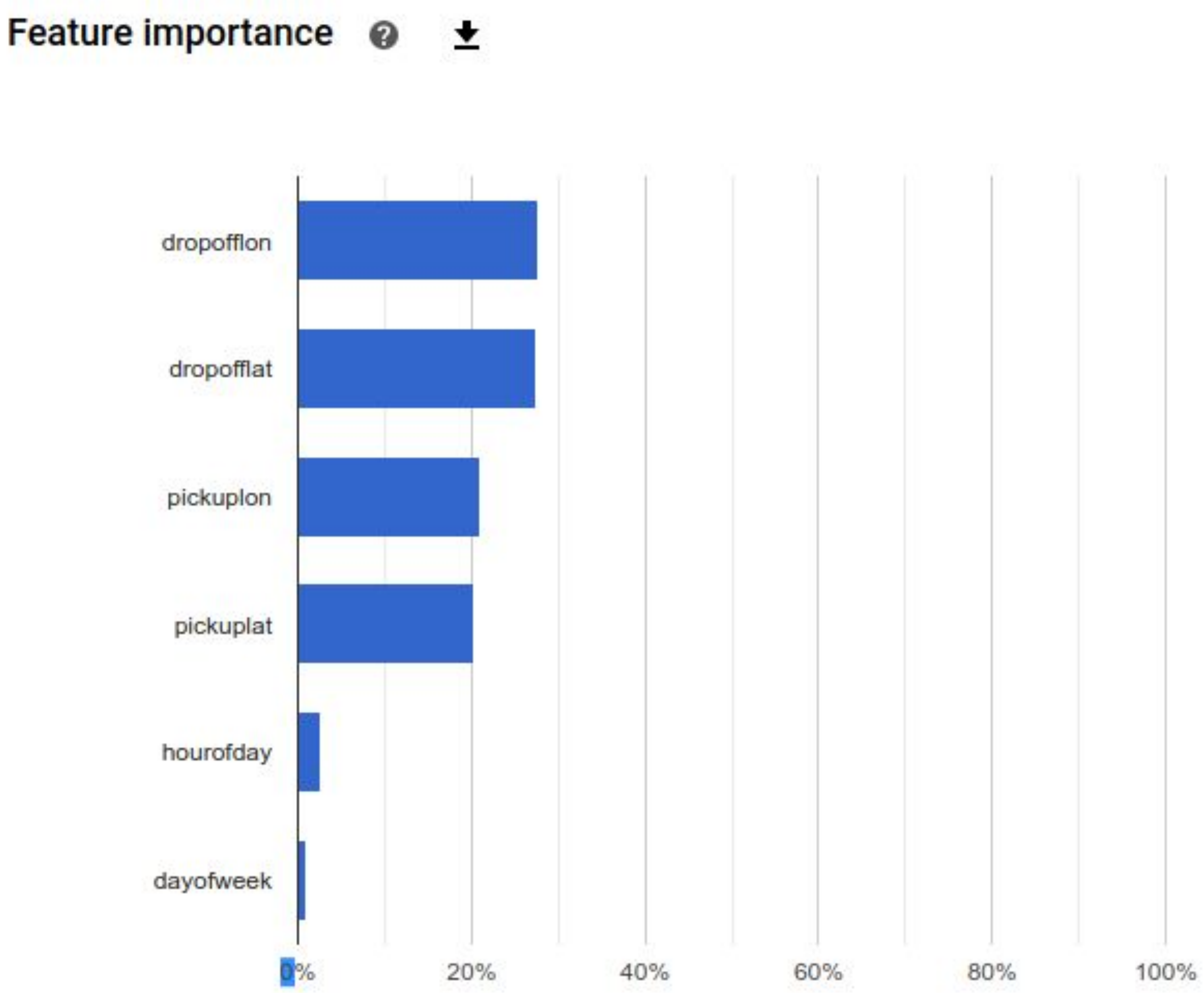
- Vertex supported methods: IG, sampled Shapley, XRAI

- AutoML Tables includes Permutation Feature Importances by default for global importances and sampled Shapley for single instance importances.

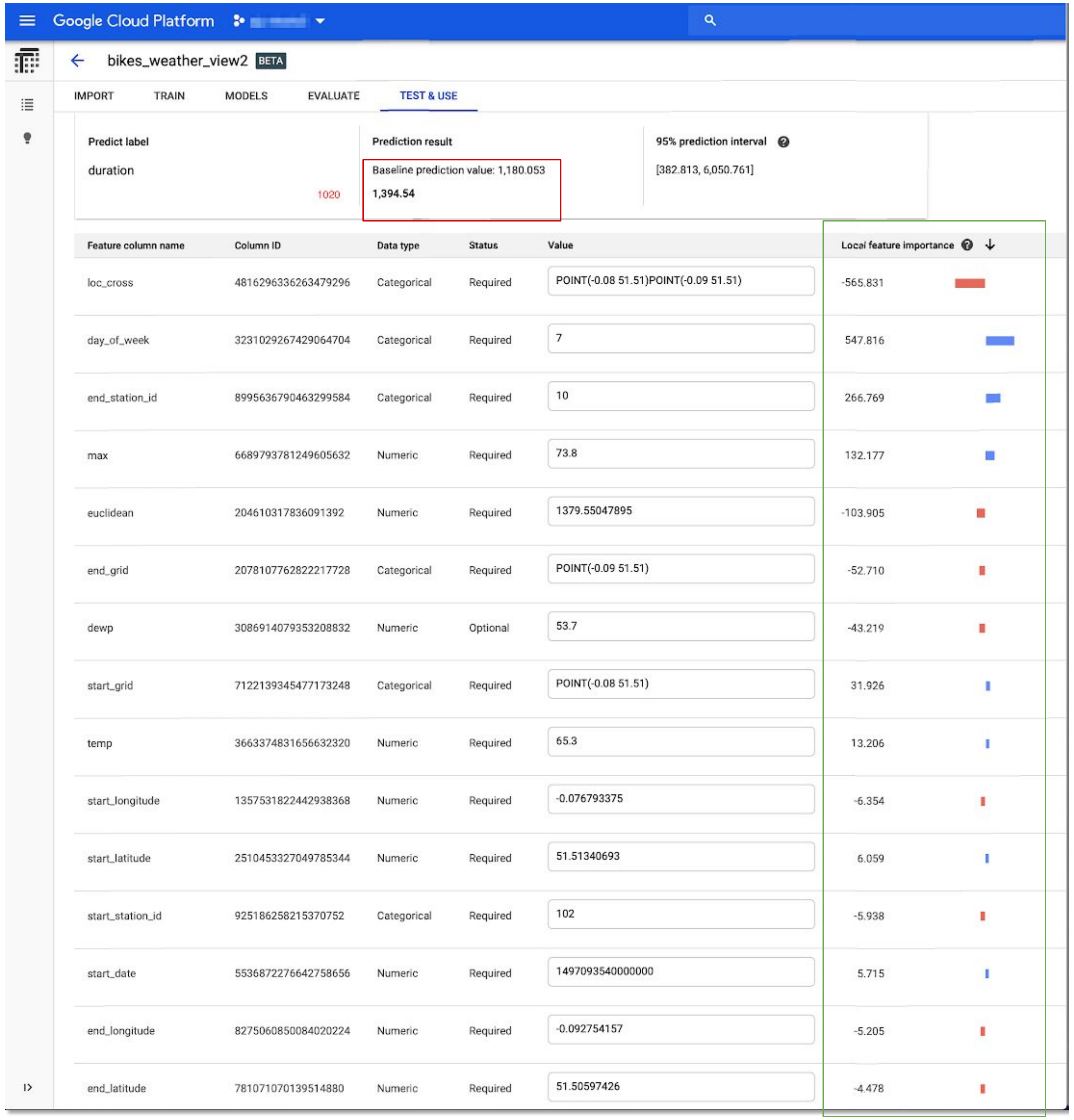


Permutation Feature Importances on AutoML Tables

- PFIs are computed for every feature in the input.
- The raw values are computed by taking the difference in the model evaluation metric, and then rescaled so that all of the values add to 1.
- In this case, for our (favorite) NYC Taxi Fare problem, the location features prove to be more important via PFI than the time features.



Local feature importance with AutoML tables



- Deploy your model
- Go to TEST & USE tab
- select ONLINE PREDICTION
- enter fields for prediction
- check GENERATE feature importance at the bottom of the page

Feature Attributions with Vertex AI

1. Save your Tensorflow model as a SavedModel on Cloud Storage

```
model.fit(...)
model.save(gcs_bucket/path/to/saved/model)
```

Feature Attributions with Vertex AI

1. Save your Tensorflow model as a SavedModel on Cloud Storage
2. Create explainability metadata

```
metadata = aiplatform.explain.ExplanationMetadata(
    inputs={"image": input_metadata}, outputs={"class": output_metadata}
)

parameters = aiplatform.explain.ExplanationParameters(
    {"integrated_gradients_attribution": {"step_count": 50}}
)
```

Feature Attributions with Vertex AI

1. Save your Tensorflow model as a SavedModel on Cloud Storage
2. Create explainability metadata
3. Upload your model to Vertex Model Registry

```
model = aiplatform.Model.upload(
    display_name=MODEL_NAME,
    artifact_uri=OUTDIR,
    serving_container_image_uri=SERVING_IMAGE,
    explanation_parameters=parameters,
    explanation_metadata=metadata,
)
```

Feature Attributions with Vertex AI

1. Save your Tensorflow model as a SavedModel on Cloud Storage
2. Create explainability metadata
3. Upload your model to Vertex Model Registry
4. Submit prediction request for explanation

```
response = endpoint.explain(instances_list)
```

Lab

Deploying an Explainable Image Model with Vertex AI

In this lab you will deploy a train and deploy an image model with Vertex AI. Then you will add the necessary model signatures, and upload/deploy the model to Vertex AI to serve online predictions with explanations.

notebooks/ml_fairness_explainability/explainable_ai/labs/xai_image_vertex.ipynb