

03



Interpretability of AI

Introduction to Responsible AI in Practice

In this module, you learn to ...

- 01 Define interpretability in ML
- 02 Discuss why interpretability is important and difficult
- 03 Discover some **best practices** on interpretability
- 04 Explore **techniques** and tools to study interpretability
- 05 **Lab:** Learning Interpretability Tool for Text Summarization



Topics

- | | |
|----|---|
| 01 | Overview of Interpretability |
| 02 | Metrics Selection |
| 03 | Taxonomy of interpretability in ML Models |
| 04 | Tools to Study Interpretability |
| 05 | Hands-on Lab |



Topics

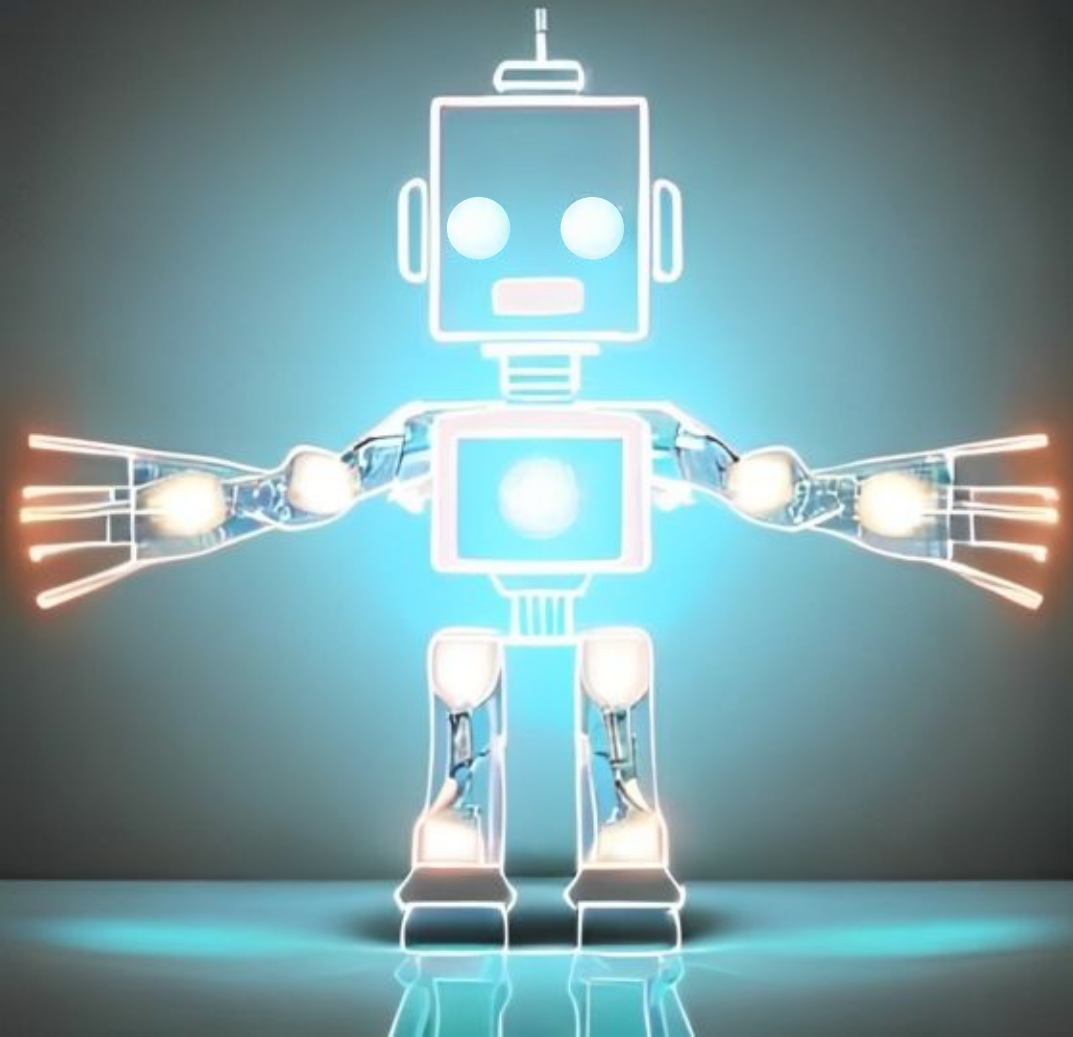
01	Overview of Interpretability
02	Metrics Selection
03	Taxonomy of interpretability in ML Models
04	Tools to Study Interpretability
05	Hands-on Lab



Interpretability relates to Google's AI

Principle #4

- 1 Be socially beneficial
- 2 Avoid creating or reinforcing unfair bias
- 3 Be built and tested for safety
- 4 Be accountable to people**
- 5 Incorporate privacy design principles
- 6 Uphold high standards of scientific excellence
- 7 Be made available for uses that accord with these principles



AI Interpretability

The ability to **explain** or to **present** an ML model's reasoning in **understandable** terms to a human.

Definition from
<https://developers.google.com/machine-learning/glossary>

What makes a good explanation?

- ✓ Completeness
- ✓ Accuracy
- ✓ Meaningfulness
- ✓ Consistency



How does Interpretability fit with explainability?

Interpretability = Explainability ?

Interest over time ?



● explainable ai
Search term

● interpretable machine learning
Search term



Why do you need Interpretability?

Question

Understand

Trust

Reflect our domain knowledge and societal values

Provide scientists and engineers with better means

Ensure AI systems are working as intended

Present models to stakeholders

Why is Interpretability difficult?

Not easy for anyone

Interpretability issues apply to humans as well as AI systems—after all, it's not always easy for a person to provide a satisfactory explanation of their own decisions.

Model complexity

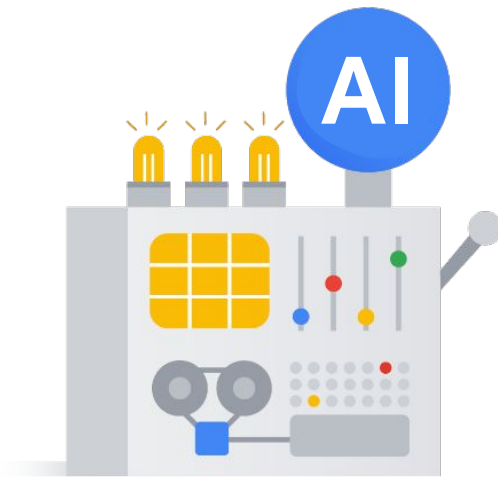
Understanding complex AI models, such as deep neural networks, can be challenging even for machine learning experts.

ML vs traditional software

Understanding and testing AI systems also offers new challenges compared to traditional software. It is much harder to pinpoint one specific bug that leads to a faulty decision.

How do you address Interpretability?

- Plan out your options to pursue interpretability
- Treat interpretability as a core part of the UX
- Design the model to be interpretable
- Choose metrics to reflect the end-goal and the end-task
- Understand the trained model
- Communicate explanations to model users
- Test, test, test



How do you address Interpretability?

- **Plan out your options to pursue interpretability**

- Treat interpretability as a core part of the UX

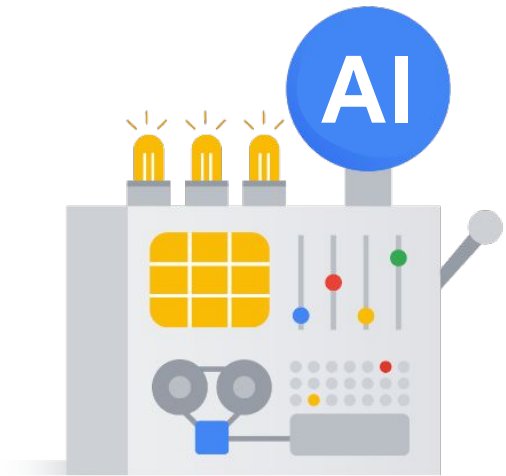
- Design the model to be interpretable

- Choose metrics to reflect the end-goal and the end-task

- Understand the trained model

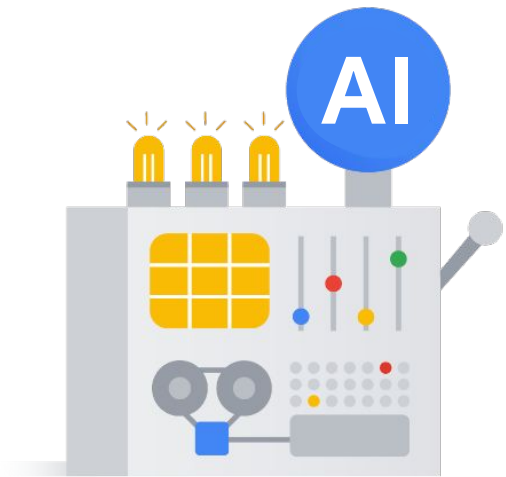
- Communicate explanations to model users

- Test, test, test



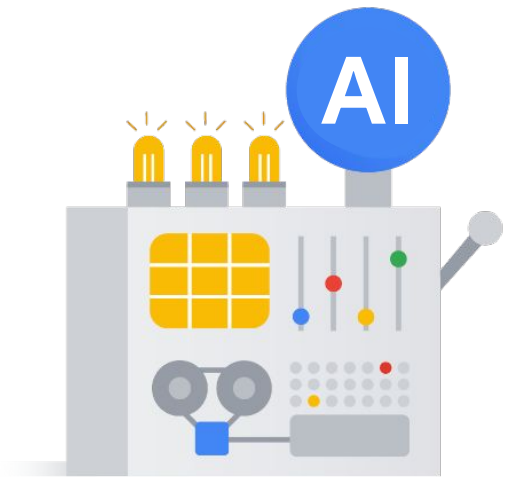
How do you address Interpretability?

- Plan out your options to pursue interpretability
- **Treat interpretability as a core part of the UX**
- Design the model to be interpretable
- Choose metrics to reflect the end-goal and the end-task
- Understand the trained model
- Communicate explanations to model users
- Test, test, test



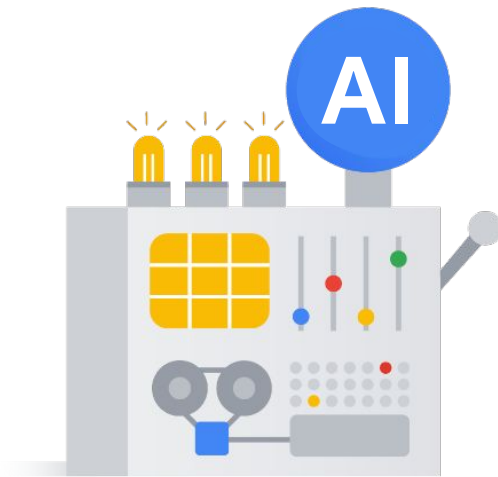
How do you address Interpretability?

- Plan out your options to pursue interpretability
- Treat interpretability as a core part of the UX
- **Design the model to be interpretable**
- Choose metrics to reflect the end-goal and the end-task
- Understand the trained model
- Communicate explanations to model users
- Test, test, test



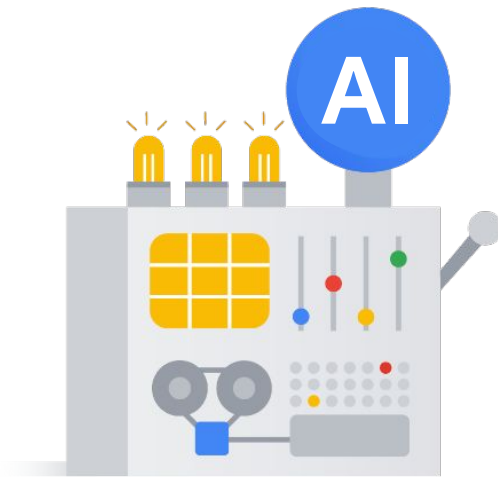
How do you address Interpretability?

- Plan out your options to pursue interpretability
- Treat interpretability as a core part of the UX
- Design the model to be interpretable
- **Choose metrics to reflect the end-goal and the end-task**
- Understand the trained model
- Communicate explanations to model users
- Test, test, test



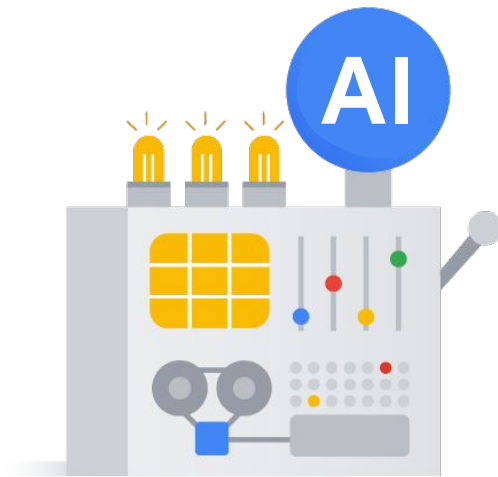
How do you address Interpretability?

- Plan out your options to pursue interpretability
- Treat interpretability as a core part of the UX
- Design the model to be interpretable
- Choose metrics to reflect the end-goal and the end-task
- **Understand the trained model**
- Communicate explanations to model users
- Test, test, test



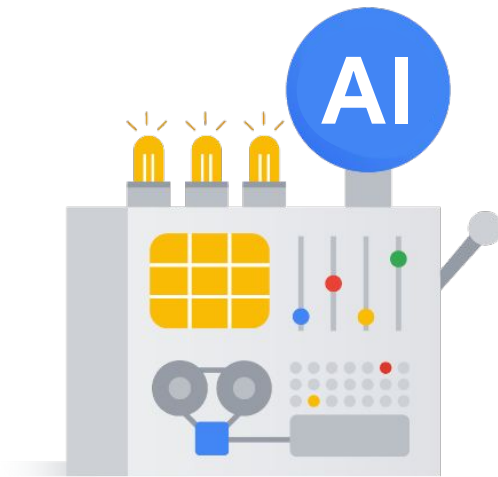
How do you address Interpretability?

- Plan out your options to pursue interpretability
- Treat interpretability as a core part of the UX
- Design the model to be interpretable
- Choose metrics to reflect the end-goal and the end-task
- Understand the trained model
- **Communicate explanations to model users**
- Test, test, test



How do you address Interpretability?

- Plan out your options to pursue interpretability
- Treat interpretability as a core part of the UX
- Design the model to be interpretable
- Choose metrics to reflect the end-goal and the end-task
- Understand the trained model
- Communicate explanations to model users
- **Test, test, test**



Topics

01	Overview of Interpretability
02	Metrics Selection
03	Taxonomy of interpretability in ML Models
04	Tools to Study Interpretability
05	Hands-on Lab



A metric is a statistic that you care about

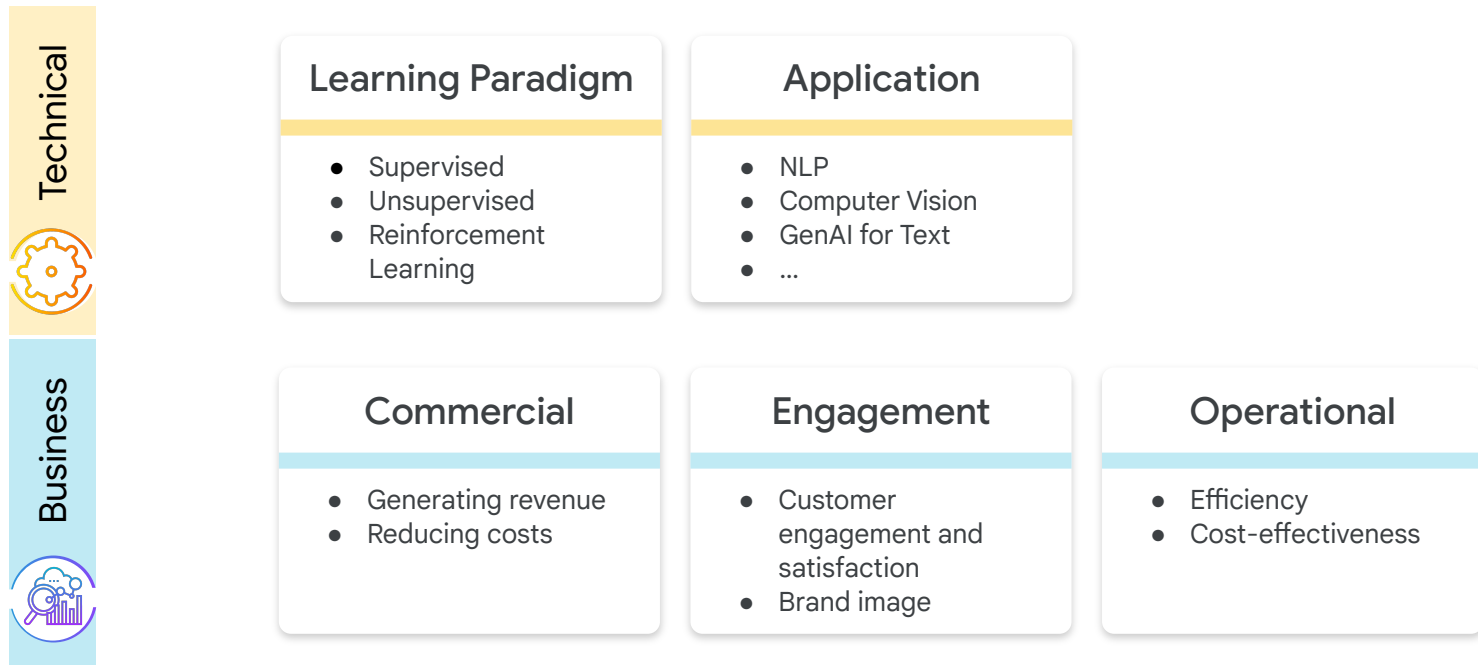


Technical metric

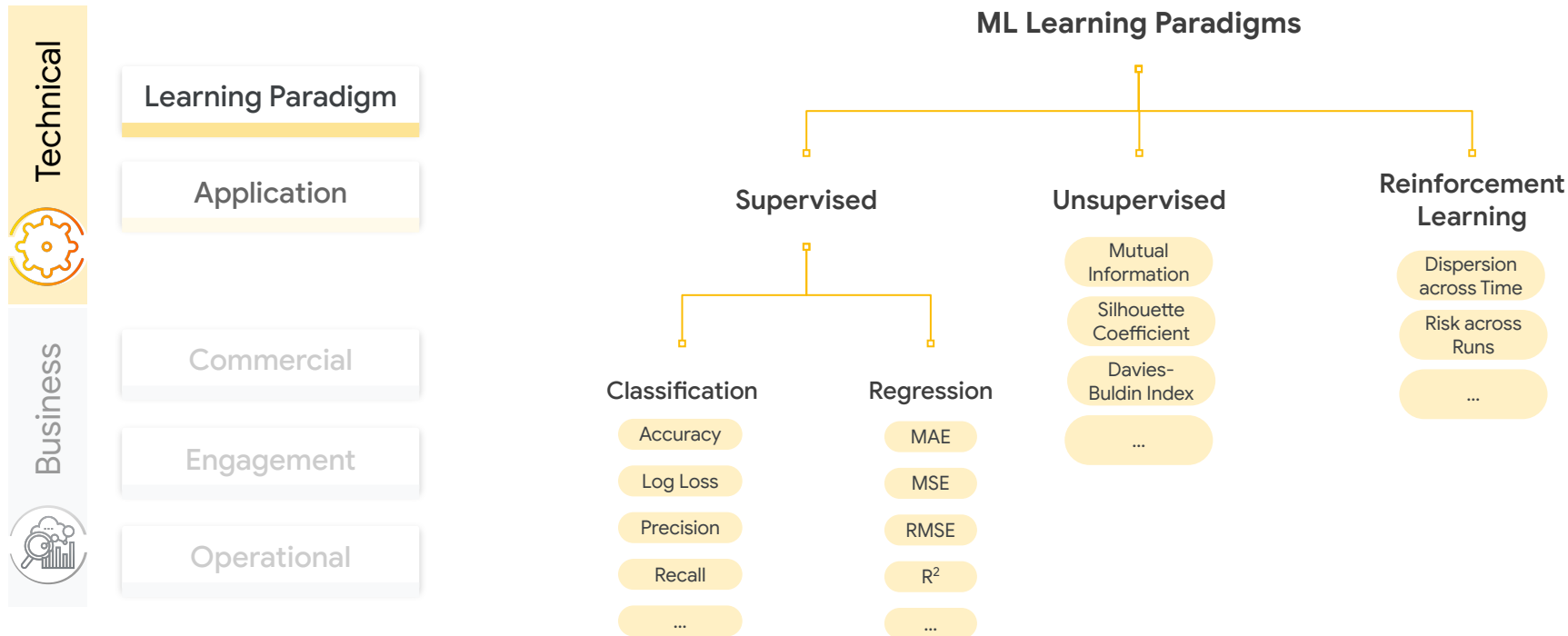


Business metric

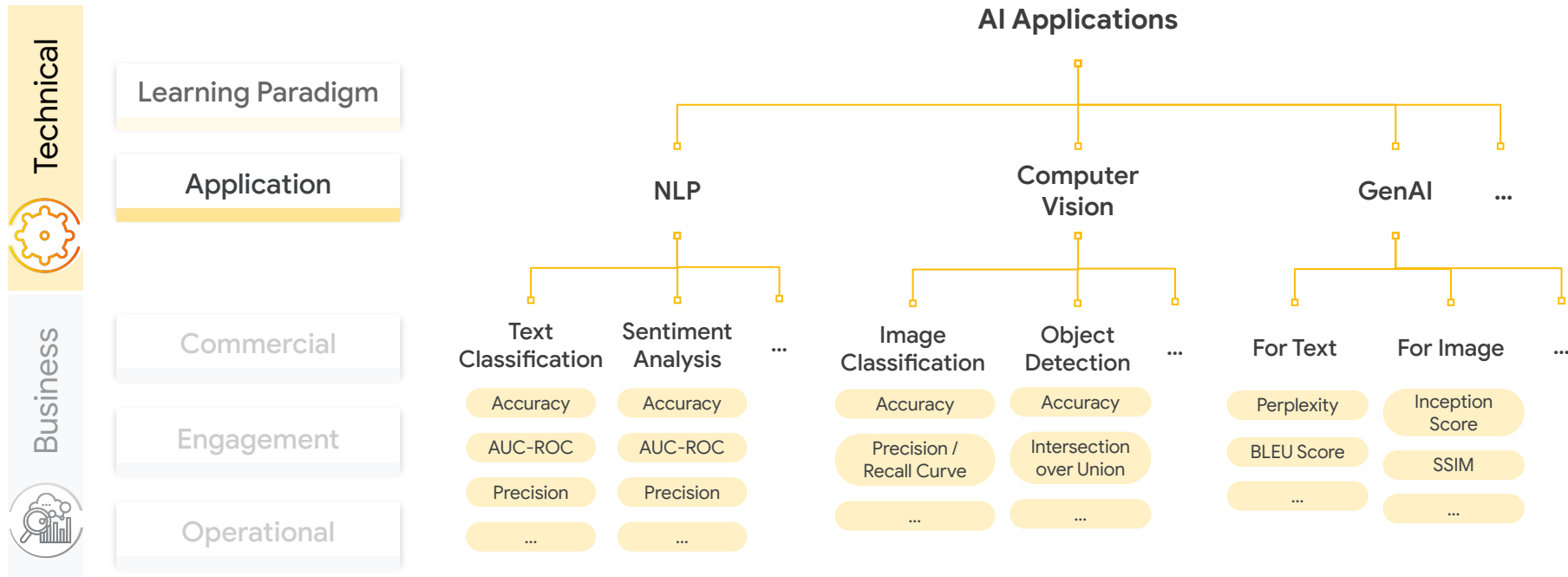
Different metrics are used for different ML scenarios



Different metrics can be applied to different ML models and use cases



Different metrics can be applied to different ML models and use cases



Different metrics can be applied to different ML models and use cases



Different metrics can be applied to different ML models and use cases

Technical	Learning Paradigm	Accuracy	Precision	AUC ROC	MAE	..
	Application	Log Loss	Recall	Precision / Recall Curve	RMSE	..
Business	Commercial	Growth	Sales Conversion Rate	Average Order Value	Customer Acquisition Cost	..
	Engagement	Customer Retention Rate	Customer Churn Rate	Net Promoter Score	Customer Lifetime Value	..
	Operational	Savings	Resource Utilization	Response Time	Process Time Reduction	..

Some best practices for metrics selection are:

- Define problem type and goals early-on
- For classification, look at per-class metrics individually if possible
- For regression, evaluate errors proportionally to the label
- Check metrics for important data slices

Some best practices for metrics selection are:

- **Define problem type and goals early-on**

- For classification, look at per-class metrics individually if possible

- For regression, evaluate errors proportionally to the label

- Check metrics for important data slices



Some best practices for metrics selection are:

- Define problem type and goals early-on

- For classification, look at per-class metrics individually if possible**

- For regression, evaluate errors proportionally to the label

- Check metrics for important data slices

Object Detection

Overall performance: 70% ACC

Per-class performance:



76%



?



63%



?



38%



?



...



...

...

Some best practices for metrics selection are:

- Define problem type and goals early-on

- For classification, look at per-class metrics individually if possible

- For regression, evaluate errors proportionally to the label**

- Check metrics for important data slices

Price prediction |

<i>y_pred</i>	<i>y_true</i>	<i>MAE</i>	<i>MAPE</i>
\$5	\$10	5	100%
\$100	\$105	5	5%
...

Some best practices for metrics selection are:

- Define problem type and goals early-on
- For classification, look at per-class metrics individually if possible
- For regression, evaluate errors proportionally to the label
- Check metrics for important data slices**

Income prediction

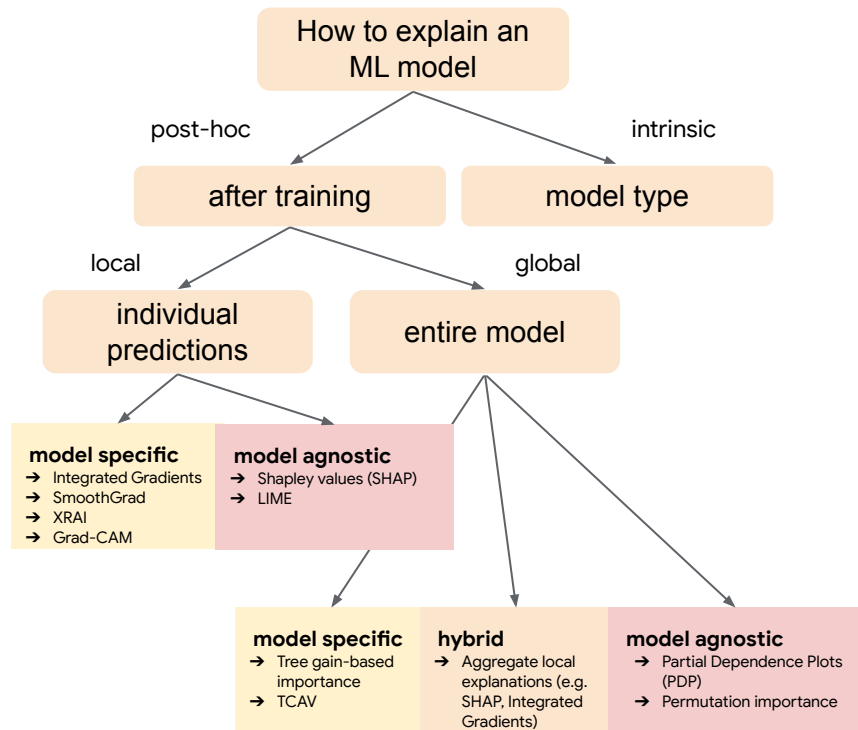
<i>Feature</i>		<i>Count</i>
<i>Name</i>	<i>Values</i>	
gender	Female	33%
	Male	67%
age	<30	48%
	>=30	52%
...

Topics

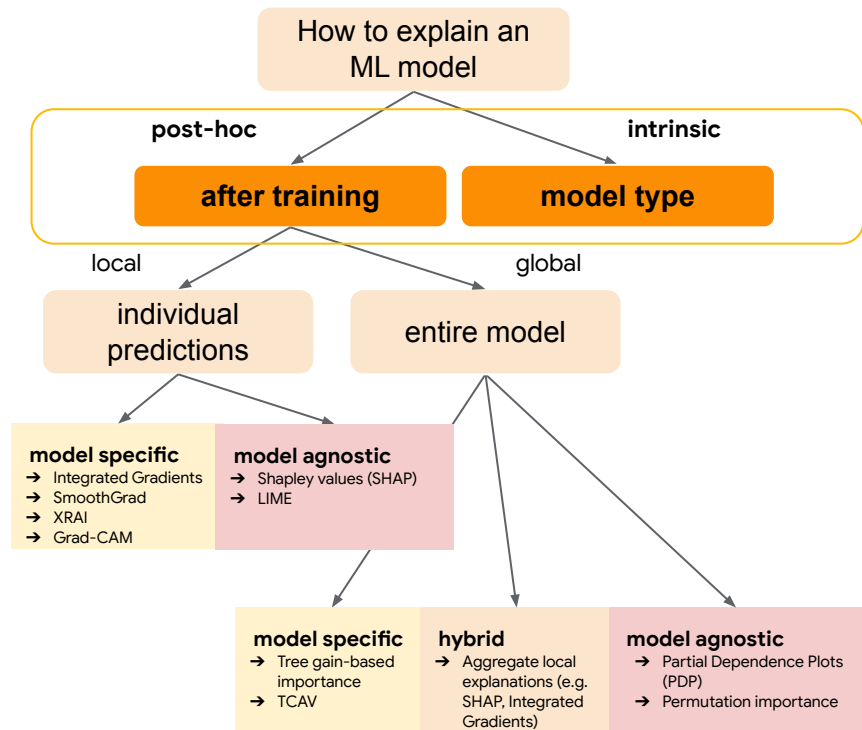
01	Overview of Interpretability
02	Metric Selection
03	Taxonomy of interpretability in ML Models
04	Tools to Study Interpretability
05	Hands-on Lab



How do you explain an ML model?



How do you explain an ML model?



Post-hoc

Apply post-training methods.

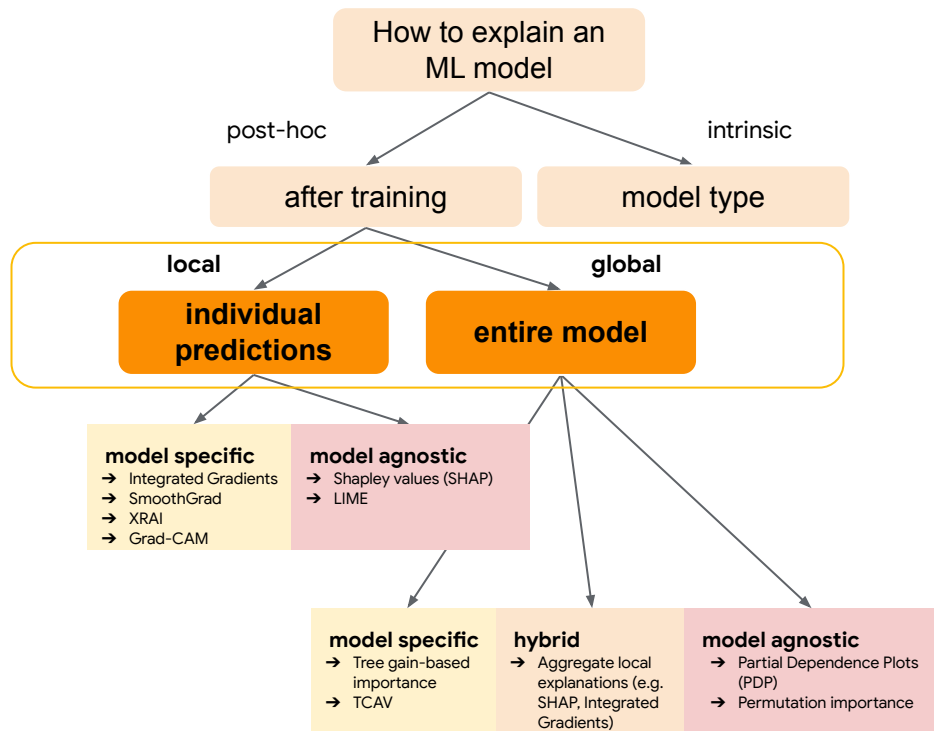
- ✓ For non-intrinsic models
- ✓ For a standardized approach across model types

Intrinsic

Apply specialized methods to the model type.

- ✓ For simple models
(linear models, decision tree, bayesian networks, ...)

How do you explain an ML model?



Local

Interpretability of individual predictions or a small part of the model's prediction space.

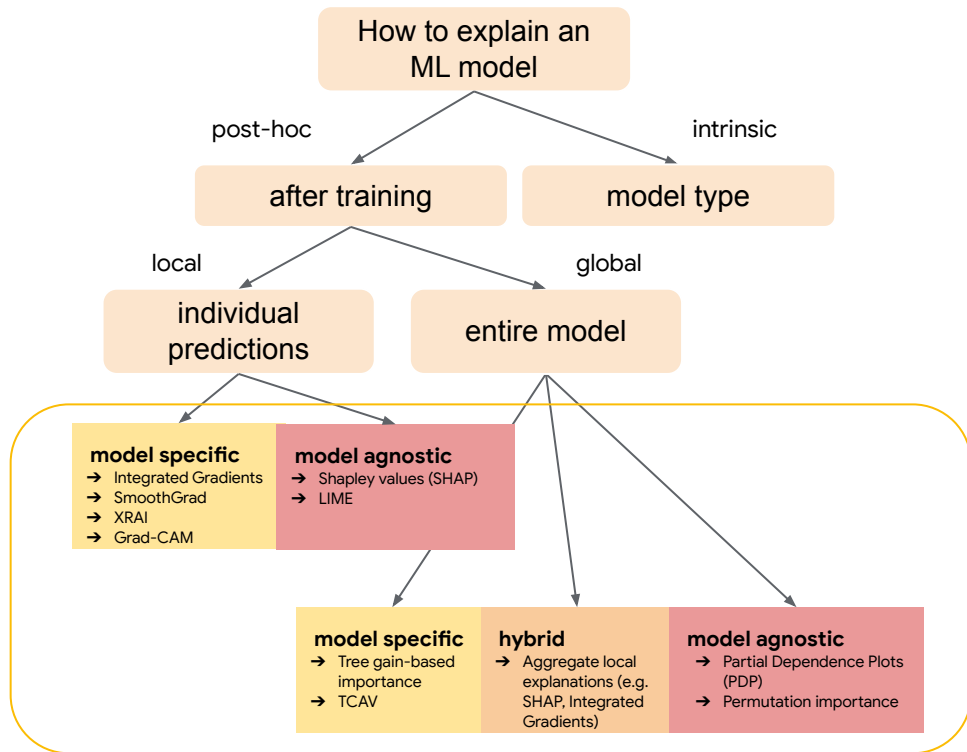
- ✓ Higher precision
- ✗ Lower recall

Global

Aggregated, ranked contributions of input variables for the entire model's prediction space.

- ✓ Higher recall
- ✗ Lower precision

How do you explain an ML model?



Model specific

Apply specialized post-training methods to the model type.

Model agnostic

Apply generic post-training method for any model.

(Global) Model-agnostic post-hoc methods: Permutation Feature Importance

Measures the importance of a feature by calculating the difference in the model's prediction error after permuting the feature.

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



(Global) Model-agnostic post-hoc methods: Permutation Feature Importance

Measures the importance of a feature by calculating the difference in the model's prediction error after permuting the feature.

- ✓ Very intuitive
- ✓ Easy to implement
- ✓ Highly compressed global insight
- ✓ No re-training needed
- ✓ All features interactions are accounted for
- ✗ Unreliable for correlated features
- ✗ No insights into individual predictions
- ✗ Needs support of feature distribution view
- ✗ Results can vary with different permutations

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

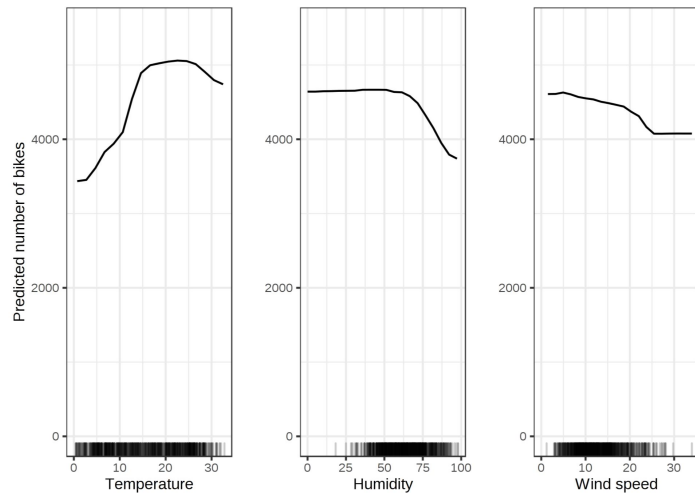


Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24

This method is **meaningful** and partially **accurate**, but it is **not complete** and **not consistent**.

(Global) Model-agnostic post-hoc methods: Partial Dependence Plots (PDPs)

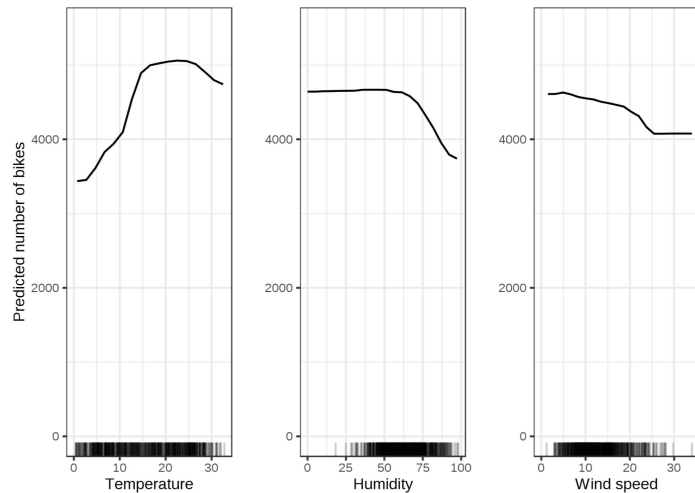
Shows the marginal effect one or two features have on the predicted outcome of a machine learning model if you force all data points to assume that feature value.



(Global) Model-agnostic post-hoc methods: Partial Dependence Plots (PDPs)

Shows the marginal effect one or two features have on the predicted outcome of a machine learning model if you force all data points to assume that feature value.

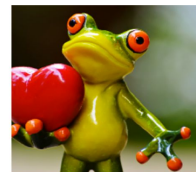
- ✓ Very intuitive
- ✓ Easy to implement
- ✗ Features are assumed to be independent
- ✗ Missing insights for individual predictions
- ✗ At most two features
- ✗ Needs support of feature distribution view



This method is **meaningful**, partially **accurate**, and **consistent**, but it is **not complete**

(Local) Model-agnostic post-hoc methods: LIME

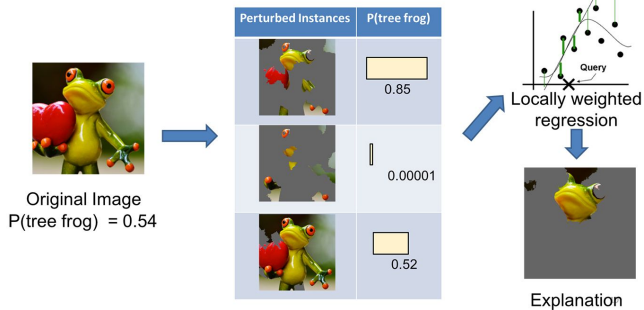
Creates explanations by approximating the underlying model locally with an interpretable one (usually linear or decision tree).



Original Image



Interpretable Components

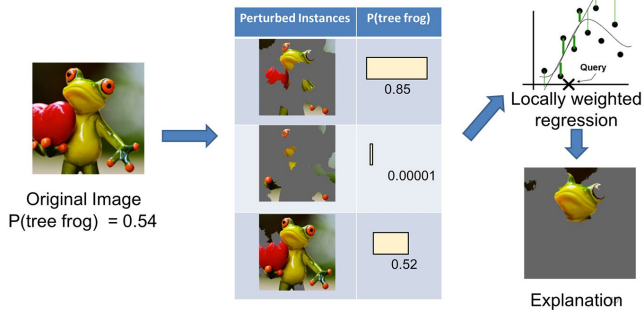
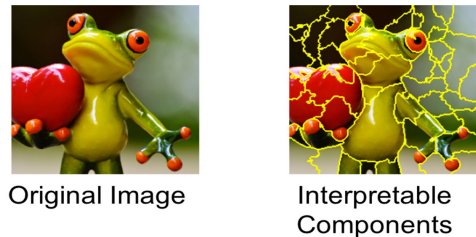


Sources: Marco Tulio Ribeiro, [Pixabay](#)

(Local) Model-agnostic post-hoc methods: LIME

Creates explanations by approximating the underlying model locally with an interpretable one (usually linear or decision tree).

- ✓ Very intuitive
- ✓ Provides interpretations from local models
- ✓ Can work on text, images, and tabular data
- ✓ Can do global interpretation with SP-LIME*
- ✗ Linear assumption reduces accuracy
- ✗ Only works on individual predictions
- ✗ Results can vary upon generation of different synthetic data

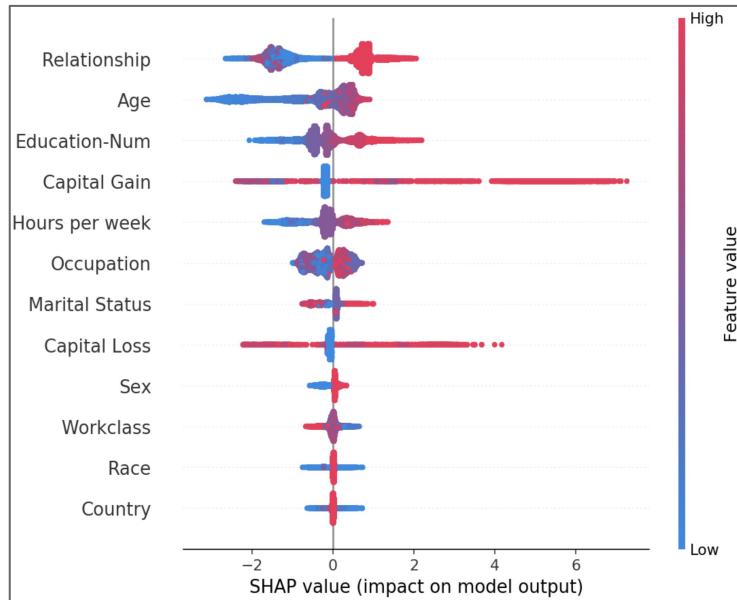


Sources: Marco Tulio Ribeiro, [Pixabay](#)

This method is **meaningful**, but it is **not accurate** and **not complete** and **not consistent**.

(Local) Model-agnostic post-hoc methods: SHAP

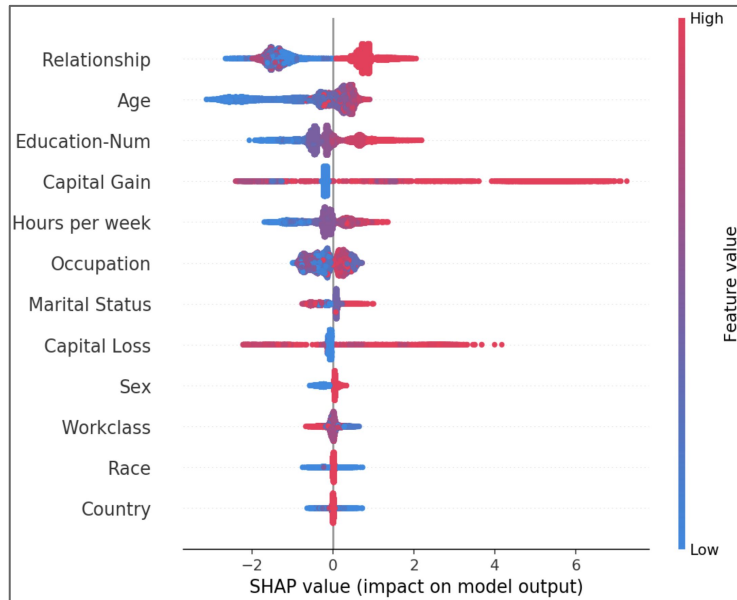
Generates individual prediction features scores that can be aggregated for global model feature importances on tabular and text data.



(Local) Model-agnostic post-hoc methods: SHAP

Generates individual prediction features scores that can be aggregated for global model feature importances on tabular and text data.

- | | |
|---|---|
| ✓ Easy to interpret | ✗ Ignores feature interactions |
| ✓ Compact representation | ✗ Can create new data points that may be irrepresentative |
| ✓ Individual explanation can be aggregated into global model explanations | ✗ Sampling provides an approximate accuracy |
| ✓ All contributing features are considered | ✗ Computational cost is high on large feature sets |



This method is mostly **meaningful**, mostly **accurate**, **complete**, and mostly **consistent**.

Topics

01	Overview of Interpretability
02	Metric Selection
03	Taxonomy of interpretability in ML Models
04	Tools to Study Interpretability
05	Hands-on Lab



What are good tools to study interpretability?

Data Card Playbook
&
Model Card Toolkit

Learning
Interpretability Tool
(LIT)

Vertex
Explainable AI

Data Card Playbook : a toolkit for transparency in AI dataset documentation

Dataset Name (Acronym)	Write a short summary describing your dataset (limit 200 words). Include information about the content and topic of the data, sources and motivations for the dataset, benefits and the problems or use cases it is suitable for.
<small>DATASET LINK</small> Dataset Link	<small>DATA CARD AUTHOR(S)</small> <ul style="list-style-type: none">• Name, Team: (Owner / Contributor / Manager)• Name, Team: (Owner / Contributor / Manager)• Name, Team: (Owner / Contributor / Manager)
Authorship ⓘ	
Dataset Overview ⓘ	
Example of Data Points	
Motivations & Intentions ⓘ	
Access, Retention, & Wipeout ⓘ	
Provenance ⓘ	
Human and Other Sensitive Attributes	
Extended Use ⓘ	
Transformations ⓘ	
Annotations & Labeling ⓘ	
Validation Types ⓘ	
Sampling Methods	
Known Applications & Benchmarks	
Terms of Art ⓘ	
Reflections on Data ⓘ	

<https://sites.research.google/datacardsplaybook/>

Model Card Toolkit : a toolkit for transparency in AI model documentation

Model Cards are jinja templates.

A few pre-made templates exist, but you can freely edit them or create your own.

```
import model_card_toolkit

# Initialize the Model Card Toolkit with a path
# to store generate assets
model_card_output_path = ...
mct = model_card_toolkit.ModelCardToolkit(
    model_card_output_path
)

# Initialize the model_card_toolkit.ModelCard,
# which can be freely populated
model_card = mct.scaffold_assets()
model_card.model_details.name = 'My Model'
model_card(...)

# Write the model card data to a JSON file
mct.update_model_card_json(model_card)

# Return the model card document as an HTML page
html = mct.export_format()
```

Model Card Toolkit : a toolkit for transparency in AI model documentation

Improves communication between model builders and product developers.

Educates users about ML models.

Provides transparency for public oversight.

Model Card for Census Income Classifier

Model Details

Overview

This is a wide and deep Keras model which aims to classify whether or not an individual has an income of over \$50,000 based on various demographic features. The model is trained on the US Census Income Dataset. This is not a production model, and this dataset has traditionally only been used for research purposes. In this Model Card, you can review quantitative components of the model's performance and data, as well as information about the model's intended use, limitations, and ethical considerations.

Version

name: 36dea2e860670aa74691b5695587afe7

Owners

- Model Cards Team, model-cards@google.com

References

- Interactive 2020-07-28T20:17:47.911887

Considerations

Use Cases

- This dataset that this model was trained on was originally created to support the machine learning community in conducting empirical analysis of ML algorithms. The Adult Data Set can be used in fairness-related studies that compare inequalities across sex and race, based on people's annual incomes.

Limitations

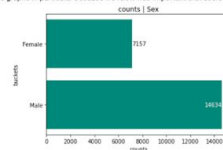
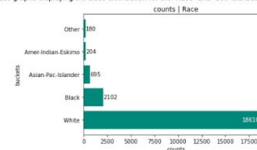
- This is a class-imbalanced dataset across a variety of sensitive classes. The ratio of male-to-female examples is about 2:1 and there are far more examples with the "white" attribute than every other race combined. Furthermore, the ratio of \$50,000 or less earners to \$50,000 or more earners is just over 3:1. Due to the imbalance across income levels, we can see that our true negative rate seems quite high, while our true positive rate seems quite low. This is true to an even greater degree when we only look at the "female" sub-group, because there are even fewer female examples in the \$50,000+ earner group, causing our model to overfit these examples. To avoid this, we can try various remediation strategies in future iterations (e.g. undersampling, hyperparameter tuning, etc), but we may not be able to fix all of the fairness issues.

Ethical Considerations

- Risk: We risk expressing the viewpoint that the attributes in this dataset are the only ones that are predictive of someone's income, even though we know this is not the case. Mitigation Strategy: As mentioned, some interventions may need to be performed to address the class imbalances in the dataset.

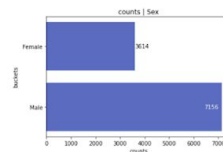
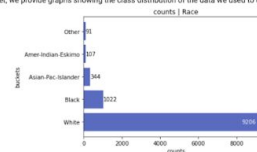
Train Set

This section includes graphs displaying the class distribution for the "Race" and "Sex" attributes in our training dataset. We chose to show these graphs in particular because we felt it was important that users see the class imbalance.

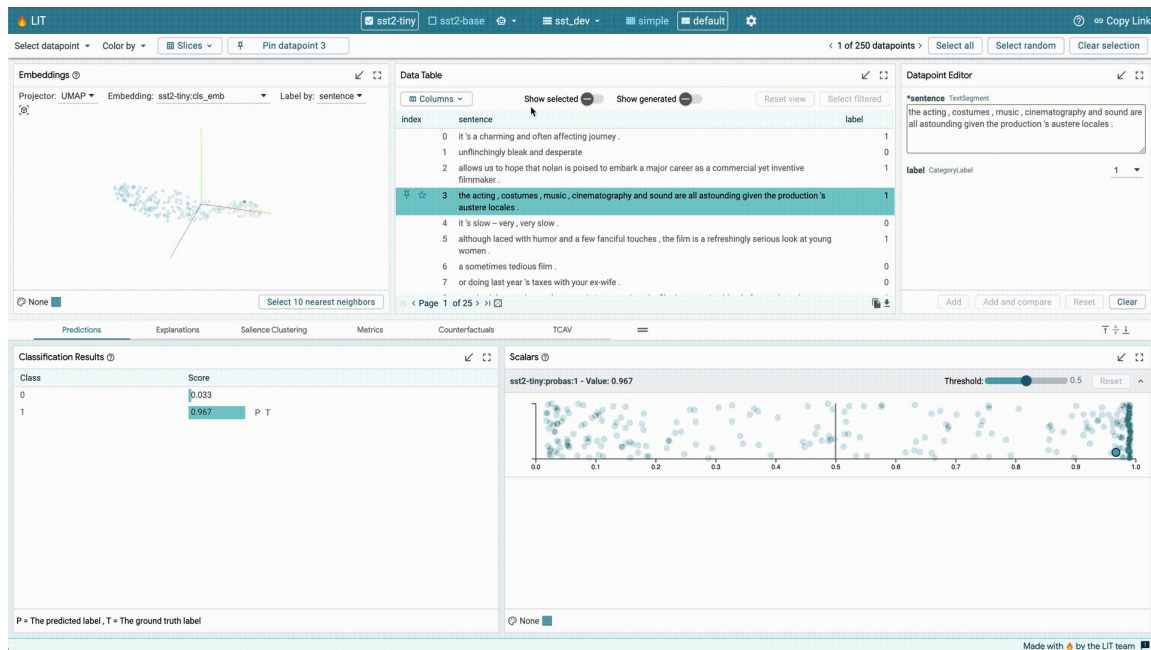


Eval Set

Like the training set, we provide graphs showing the class distribution of the data we used to evaluate our model's performance.



🔥 Learning Interpretability Tool : an open-source platform for interpretability



<https://pair-code.github.io/lit/>

🔥 Learning Interpretability Tool : an open-source platform for interpretability

The screenshot displays the LIT interface with several components:

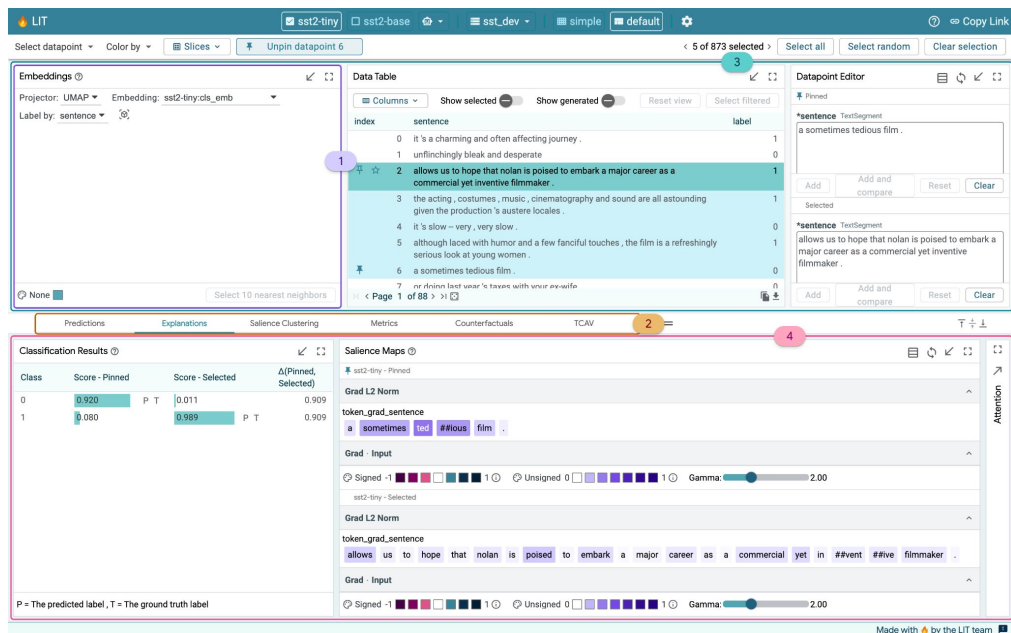
- Top Bar:** Includes the LIT logo, project name (sst2-tiny), and various settings like 'simple' and 'default'.
- Main Workspace (Top):** Contains three panels:
 - Embeddings:** Shows the projector (UMAP) and embedding (sst2-tinycls_emb).
 - Data Table:** A table with columns 'index', 'sentence', and 'label'. It shows a list of sentences with their corresponding labels. A red circle '1' highlights the 'index' column, and a red circle '3' highlights the 'label' column.
 - Datapoint Editor:** Allows editing of selected datapoints, showing a sentence and its label.
- Bottom Panel:** Contains two sub-sections:
 - Classification Results:** A table showing predicted vs. ground truth labels for two classes (0 and 1).
 - Salience Maps:** Displays visualizations of word importance for selected sentences, with a red circle '4' highlighting the 'Attention' column.

Arrows point from the text labels to the corresponding sections in the interface.

Main workspace

Group-based workspace

🔥 Learning Interpretability Tool : an open-source platform for interpretability



Embeddings

Explore UMAP / TSNE model embeddings.

Data Table

Explore, navigate, and select data.

Datapoint Editor

Deep-dive into individual examples.

Slice Editor

Create and manage data slices.

Performance

Compare models overall or by slices.

Predictions

Explore model results on single data points.

Explanations

Investigate attention for various data points.

Counterfactuals

Autogenerate new data points for evaluation.

Vertex Explainable AI : Google Cloud managed service for interpretability

Example-based explanations

Return a list of examples that are most similar to the input.



Feature-based explanations

Return feature attributions, i.e. contributions, of each feature.



Vertex Explainable AI : Google Cloud managed service for interpretability

Set up explanations for custom models via:

[Console](#) [gcloud CLI](#) [REST](#) [Python](#)

Simply import the model in Model Registry, and configure your desired explanations in the Explainability tab!

Import model

1 Name and region

2 Model settings

3 Explainability (optional)

IMPORT

CANCEL

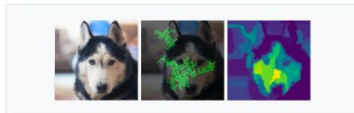
In Vertex AI, models are made explainable through feature attribution and example based explanations. You can use this information to verify that the model is behaving as expected, recognize bias in your models, and get ideas for ways to improve your model and your training data. Explainability can incur additional charges. [Learn more](#)

Explainability options

☐ No explainability

☐ Feature attribution

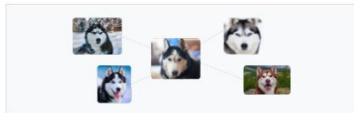
Feature attribution tells you how much each feature contributed to the predicted result.



What image pixels or regions most contributed to the classification?

☒ Example-based explanation

Example-based explanations provides approximate nearest neighbor explanations.



What images in the training dataset are most similar to the new image?

Dataset *

☒ gs://xai-ebe-fishfooding/models/bike-data-train-examples.json

BROWSE

The JSONL file that contains the examples, typically the training or evaluation dataset, to be indexed for nearest neighbor search.

Number of neighbors returned *

10

?

☒ Basic configuration

☐ Advanced configuration

Vertex Explainable AI : Google Cloud managed service for interpretability



BigQuery ML

▼ AI Explanation functions

ML.EXPLAIN_PREDICT

ML.EXPLAIN_FORECAST

ML.GLOBAL_EXPLAIN

ML.FEATURE_IMPORTANCE

ML.ADVANCED_WEIGHTS



AutoML

← bikes_weather_view2

BETA

IMPORT

TRAIN

MODELS

EVALUATE

TEST & USE

Predict label

duration

1000

Prediction result

Baseline prediction value: 1,180.053

1,394.54

95% prediction interval ⓘ

[382.813, 6,050.761]

Feature column name	Column ID	Data type	Status	Value	Local feature importance ⓘ ↓
loc_cross	481629636263479296	Categorical	Required	<div>POINT(-0.08 51.51)POINT(-0.09 51.51)</div>	-565.831 <div></div>
day_of_week	3231029267429064704	Categorical	Required	<div>7</div>	547.816 <div></div>
end_station_id	8995636790463299584	Categorical	Required	<div>10</div>	266.769 <div></div>
max	6689793781249605632	Numeric	Required	<div>73.8</div>	132.177 <div></div>
euclidean	204610317836091392	Numeric	Required	<div>1379.55047895</div>	-103.905 <div></div>
end_grid	207810776282217728	Categorical	Required	<div>POINT(-0.09 51.51)</div>	-52.710 <div></div>
dewp	308691407953208832	Numeric	Optional	<div>53.7</div>	-43.219 <div></div>

<https://cloud.google.com/vertex-ai/docs/explainable-ai>

Topics

01	Overview of Interpretability
02	Metric Selection
03	Taxonomy of interpretability in ML Models
04	Tools to Study Interpretability
05	Hands-on Lab



Lab: Explaining Text Classification with Vertex Explainable AI

