Open the Black Box:
Introduction to Model Interpretability

@KevinLemagnen
@cambridgespark

CAMBRIDGE SPARK

Setup

Option 1: Github

https://github.com/klemag/pydata_nyc2018-intro-to-model-interpretability

Or in short: https://bit.ly/2yOwLaZ

(see setup instructions in README)

Option 2: Google Colab

https://bit.ly/2J5FIRZ

CAMBRIDGE SPARK

Interpretability - Outline

1. Introduction - Why do we need it?
2. Eli5
3. LIME - **L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations
4. SHAP - **SH**apley **A**dditive ex**P**lanation

CAMBRIDGE SPARK

Why do we need it?

Job Done!

- ✓ Cleaned and preprocessed messy data
- ✓ Engineered fancy new features
- ✓ Selected the best model and tuned hyperparameters
- ✓ Trained your final model
- ✓ Got great performances on the test set

Job Done … or not …

CAMBRIDGE SPARK

Job Done … or not …

Can you explain how your model works?

Why is Interpretability important?

Algorithms are everywhere, sometimes automating important decisions that have an impact on people.

- *Insurance*: model to predict the best price to charge a client
- *Bank*: model to predict who should get a loan or not
- *Police*: model to predict who is more likely to commit a crime
- *Social media*: model to predict who is most likely to buy a product
- [...]

Black-box models are not an option

CAMBRIDGE SPARK

Bias in the data

“*Models are opinions embedded in mathematics*” Cathy O’Neil

Example 1: Predicting employees’ performance at a big company

Data available:

- past performance reviews of individual employees for the last 10 years

What if that company tends to promote men more than women?

The model will learn the bias, and predict that men are more likely to be performant ...

CAMBRIDGE SPARK

Bias in the data

Example 2: Classify images - wolves vs husky

Data available:

- Pictures of wolves and huskies

What if pictures of wolves show something different in the background?

(a) Husky classified as wolf (b) Explanation

"Why Should I Trust You?": Explaining the Predictions of Any Classifier
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin

CAMBRIDGE SPARK

Some models are easy to interpret

Linear/Logistic regression

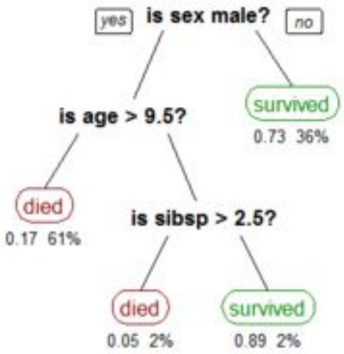
- Weight on each feature
- Know the exact contribution of each feature, negative or positive

$$Y = 3 * X1 - 2 * X2$$

Increasing X1 by 1 unit increases Y by 3 units

Single Decision Tree

- Easy to understand how a decision was made by reading from top to bottom

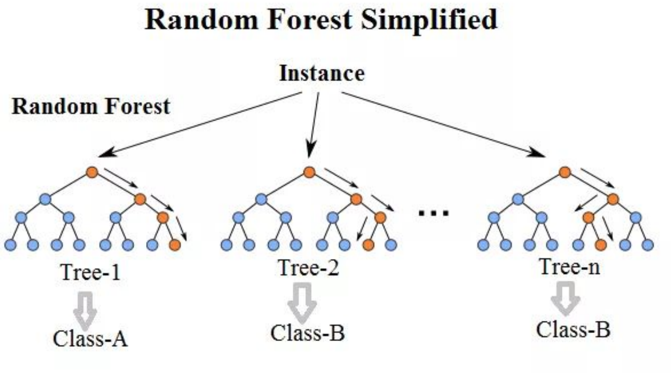


CAMBRIDGE SPARK

Some models are harder to interpret

Ensemble models (random forest, boosting, etc…)

- Hard to understand the role of each feature
- Usually comes with **feature importance**
- Doesn't tell us if feature affects decision positively or negatively

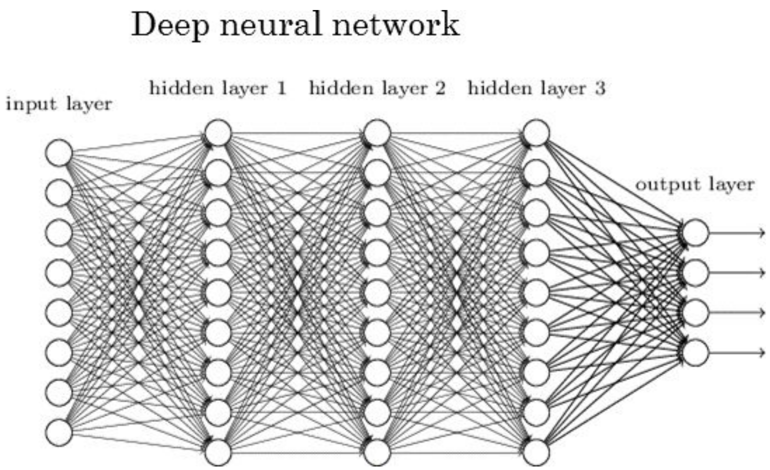


CAMBRIDGE SPARK

Some are really hard to interpret

Deep Neural Networks

- No straightforward way to relate output to input layer
- “Black-box”



CAMBRIDGE SPARK

Does it mean we can only use simple models?

- Sticking to simpler model is the best way to be confident about interpretation.
- Interpretability techniques allow usage of more complex models without losing all interpretation power

Interpretability - ELI5

“Explain Like I’m 5”

CAMBRIDGE SPARK

ELI5

- Useful to debug sklearn-like models and communicate with domain experts
- Provides global interpretation of “white box” models with a consistent API
- Provides local explanation of predictions

ELI5 - API

Explain model globally (features importance)

> `eli5.show_weights(model)`

Explain a single prediction

> `eli5.show_prediction(model, observation)`

Hands-on session

>>> ELI5

CAMBRIDGE SPARK

Interpretability - LIME

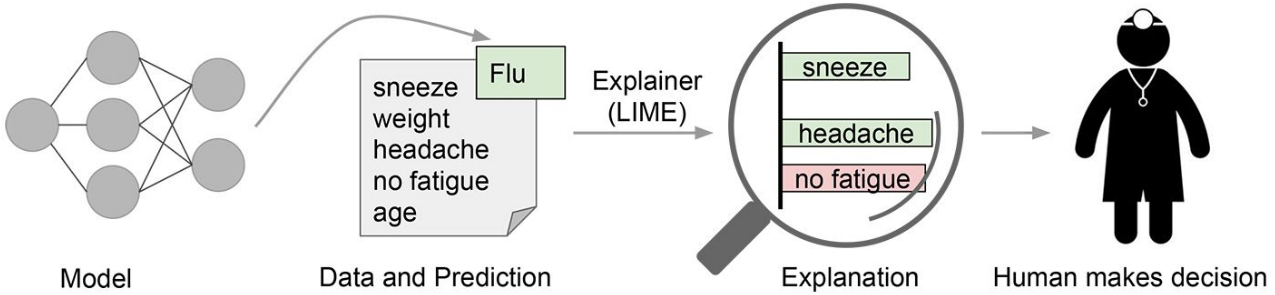
CAMBRIDGE SPARK

LIME - Local Interpretable Model-Agnostic Explanations

Local: Explains why a single data point was classified as a specific class

Model-agnostic: Treats the model as a black-box. Doesn't need to know *how* it makes predictions

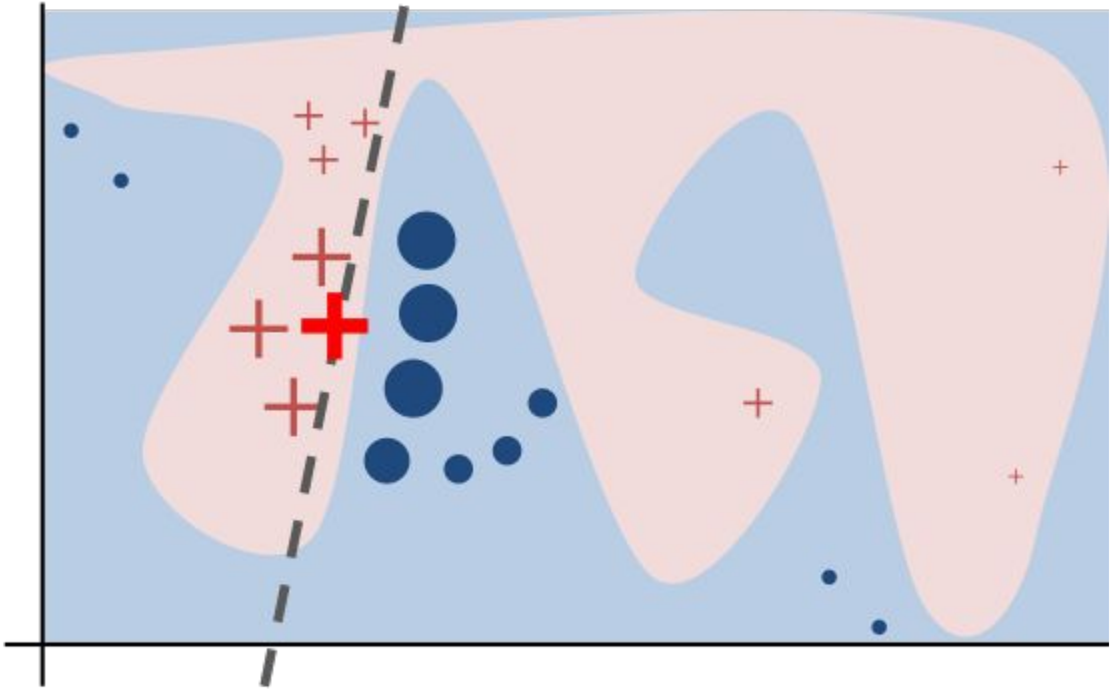
Paper "*Why should I trust you?*" published in August 2016.



"Why Should I Trust You?": Explaining the Predictions of Any Classifier
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin

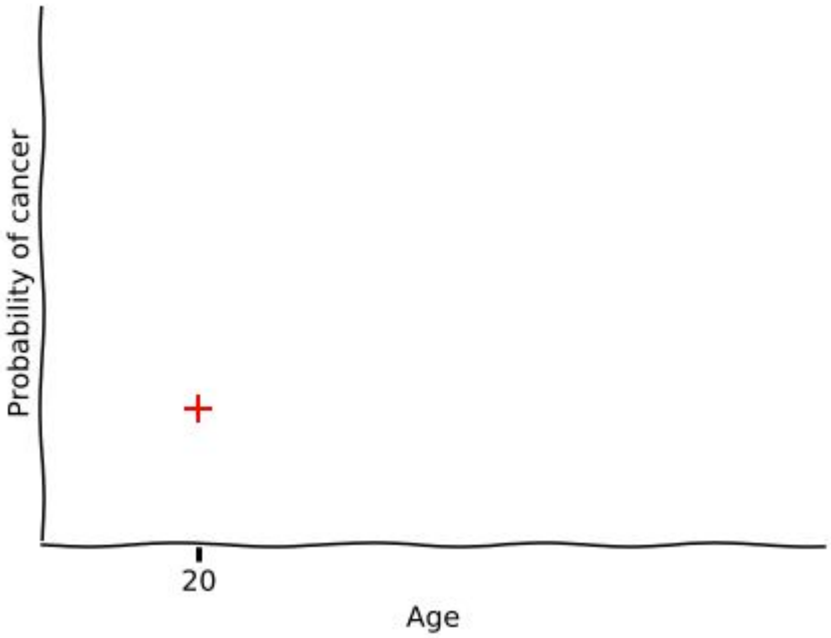
CAMBRIDGE SPARK

LIME - How does it work?

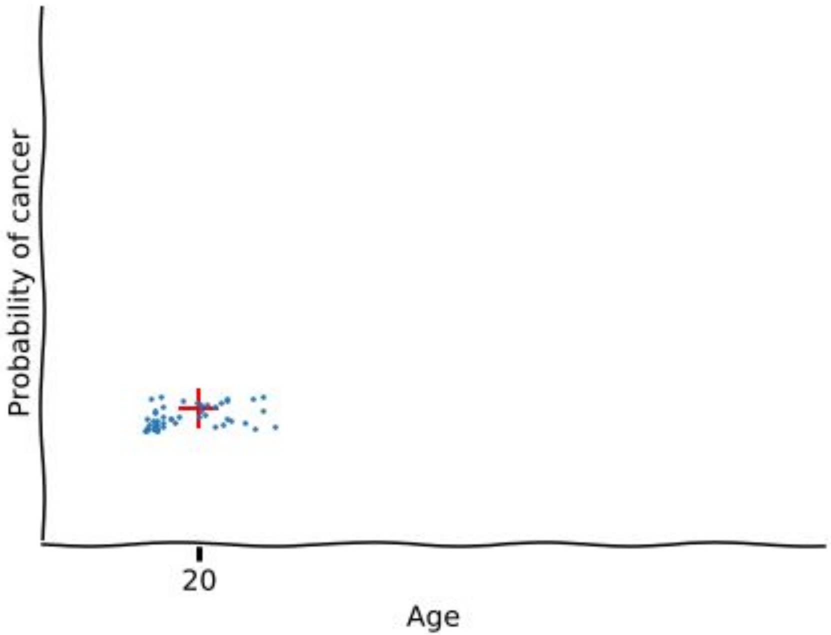


"Why Should I Trust You?": Explaining the Predictions of Any Classifier
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin

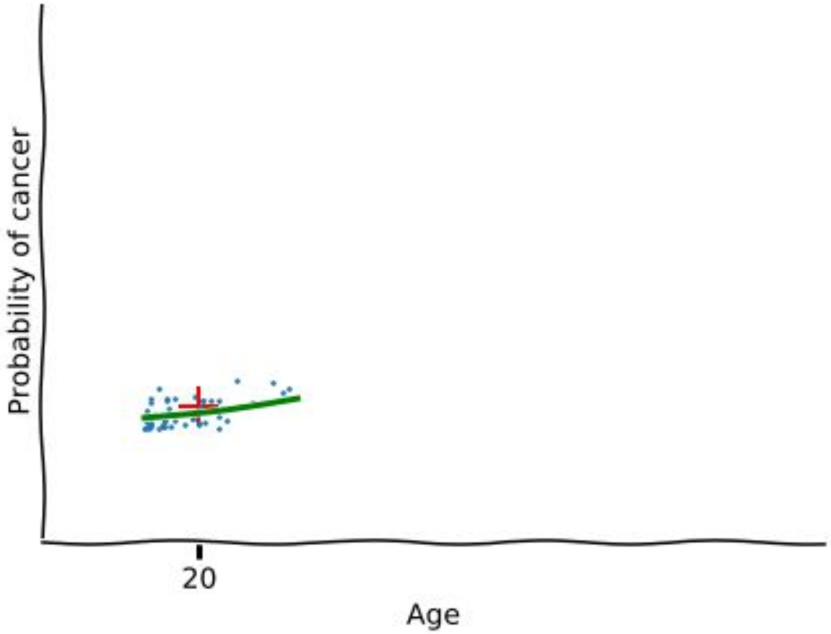
LIME - How does it work?



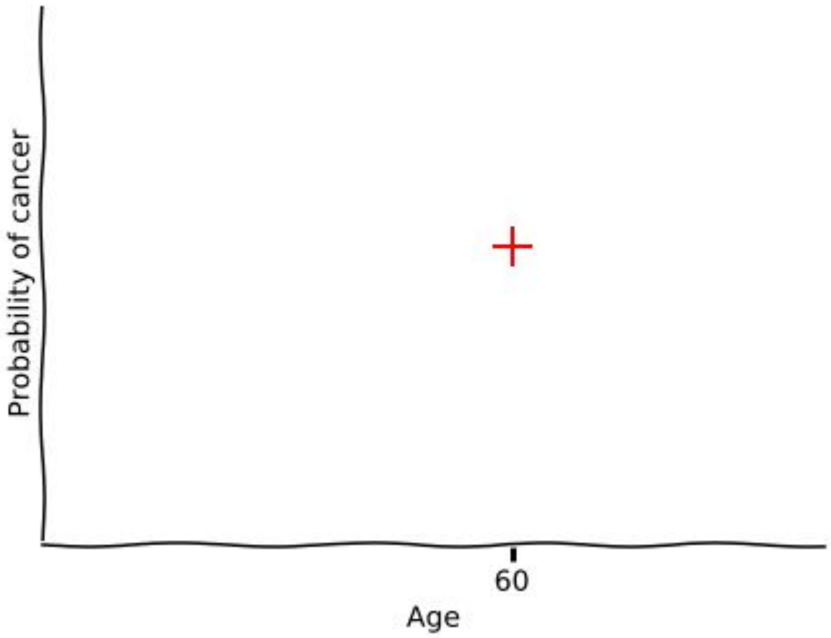
LIME - How does it work?



LIME - How does it work?

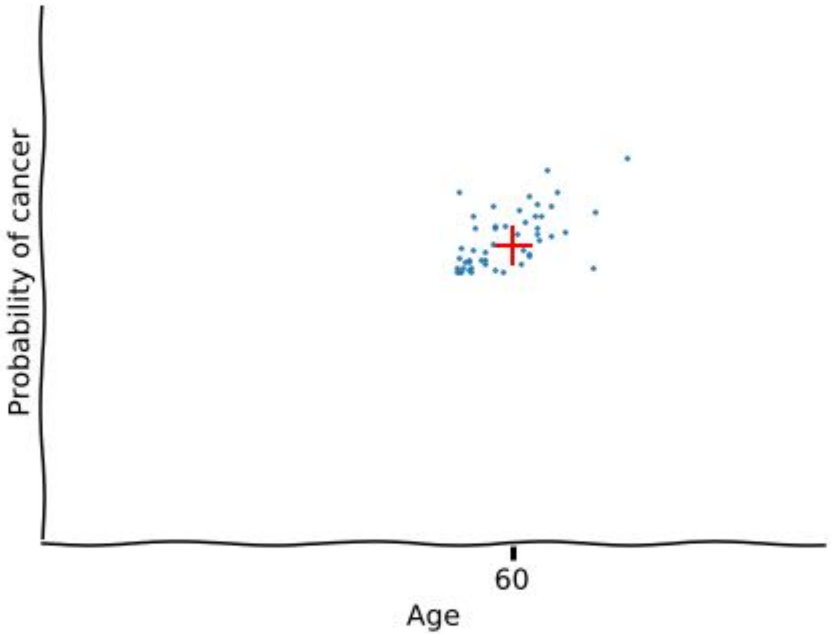


LIME - How does it work?



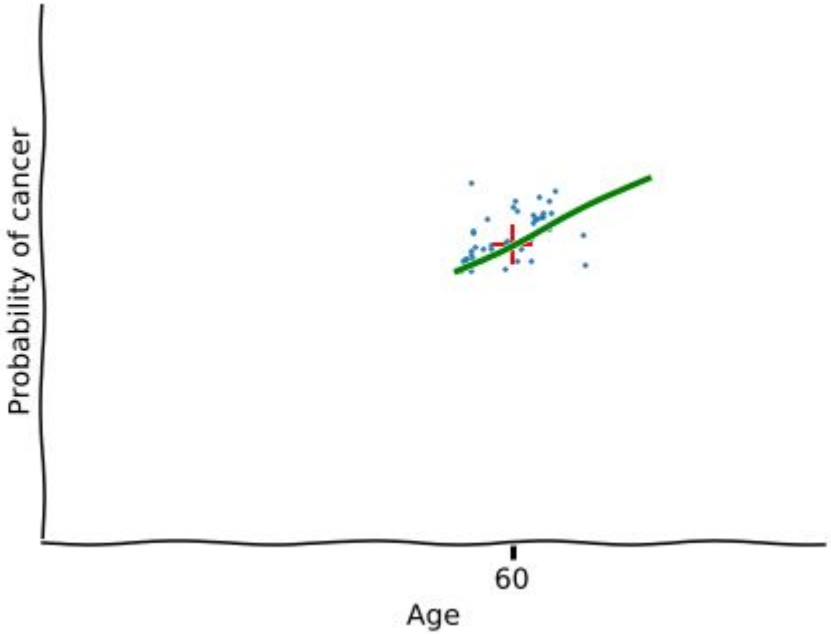
CAMBRIDGE SPARK

LIME - How does it work?



CAMBRIDGE SPARK

LIME - How does it work?

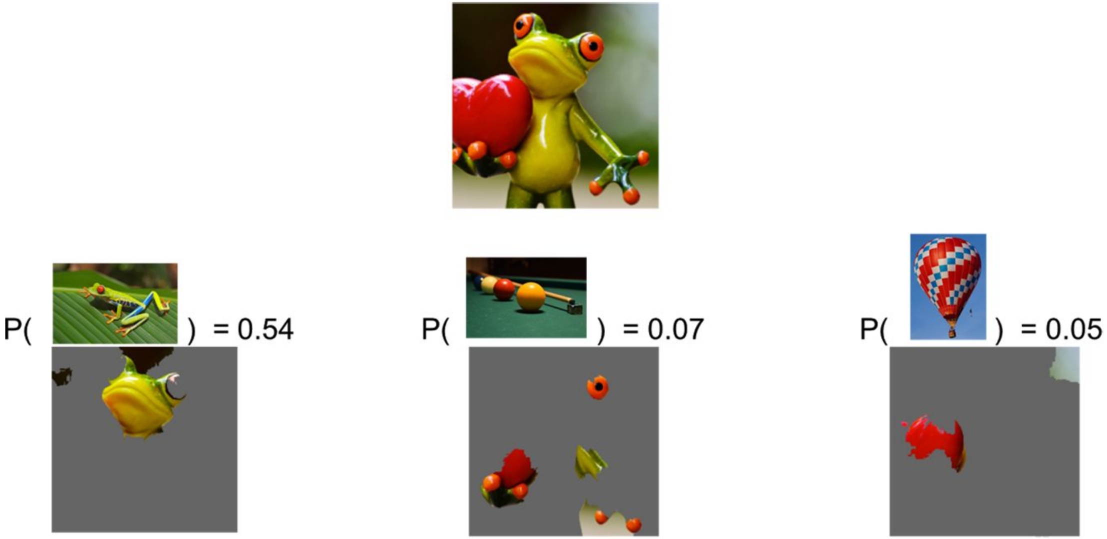


LIME - How does it work?

1. Choose an **observation** to explain
2. Create new dataset around **observation** by sampling from distribution learnt on training data
3. Calculate distances between new points and **observation**, that’s our measure of similarity
4. Use model to predict class of the new points
5. Find the subset of **m** features that has the strongest relationship with our target class
6. Fit a linear model on fake data in **m** dimensions weighted by similarity
7. Weights of linear model are used as explanation of decision

CAMBRIDGE SPARK

LIME - Can be used on images too



"Why Should I Trust You?": Explaining the Predictions of Any Classifier
Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin

LIME - Some drawbacks

- Depends on the random sampling of new points, so it can be unstable
- Fit of linear model can be inaccurate
  - But we can check the r-squared score to know if that’s the case
- Relatively slow for a single observation, in particular with images

LIME - Available “Explainers”

Lime supports many types of data:

- Tabular Explainer
- Recurrent Tabular Explainer
- Image Explainer
- Text Explainer

LIME - API

1. **Create a new explainer**
   - `my_explainer = Explainer()`
2. **Select an observation and create an explanation for it**
   - `observation = np.array([...])`
   - `my_explanation = explainer.explain_instance(`*`observation`*`, `*`predict_function`*`)`
3. **Use methods on explanation to visualise results**
   - `my_explanation.show_in_notebook()`
   - `my_explanation.get_image_and_mask()`
   - `[...]`

CAMBRIDGE SPARK

Hands-on session

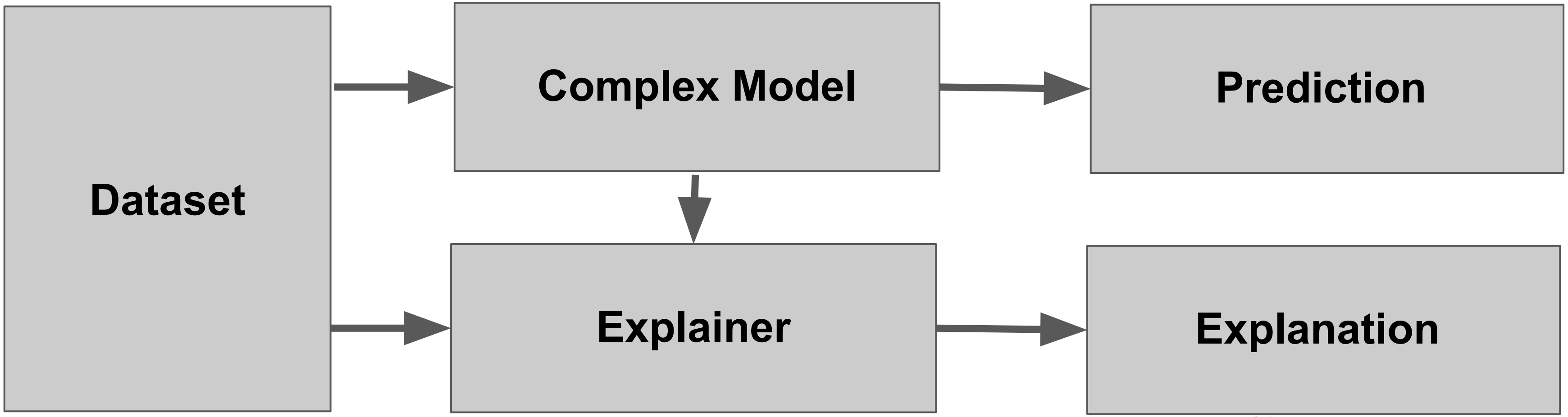
>>> LIME

CAMBRIDGE SPARK

Interpretability - SHAP

Shapley Additive Explanations (SHAP)

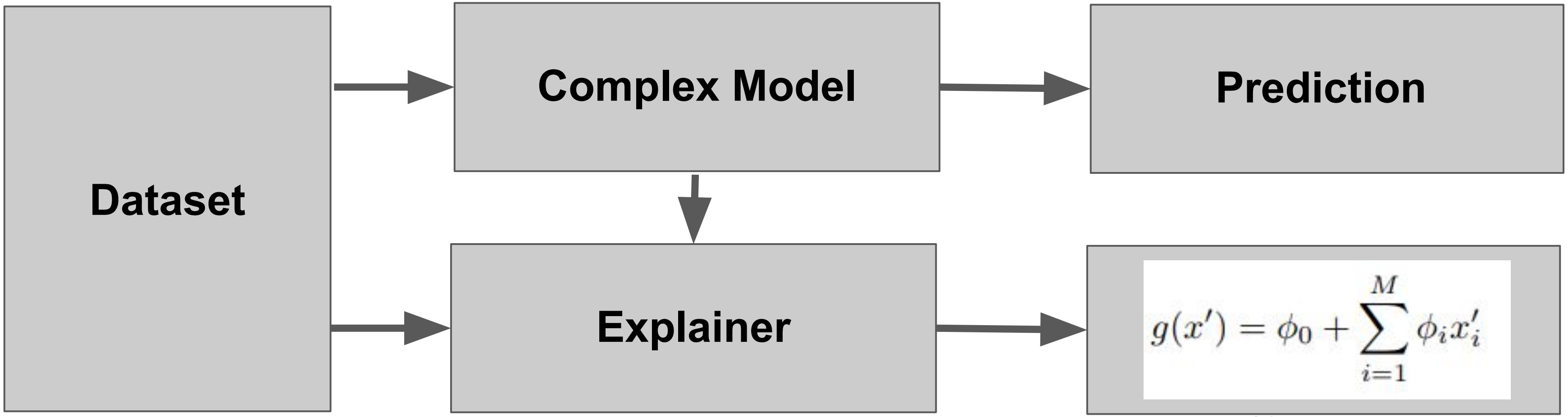
- An explanation model is a simpler model that is a good approximation of our complex model.



CAMBRIDGE SPARK

Shapley Additive Explanations (SHAP)

- An explanation model is a simpler model that is a good approximation of our complex model.
- “Additive feature attribution methods”: the local explanation is a linear function of the features.



Shapley Additive Explanations (SHAP)

In SHAP, the weight of each feature is computed using the Shapley values method from game theory.

Shapley Additive Explanations (SHAP)

In SHAP, the weight of each feature is computed using the Shapley values method from game theory.

To get the importance of feature **X{i}**:

- Get all subsets of features **S** that do not contain **X{i}**
- Compute the effect on our predictions of adding **X{i}** to all those subsets

Shapley Additive Explanations (SHAP)

In SHAP, the weight of each feature is computed using the Shapley values method from game theory.

To get the importance of feature **X{i}**:

- Get all subsets of features **S** that do not contain **X{i}**
- Compute the effect on our predictions of adding **X{i}** to all those subsets

That can be computationally expensive, but SHAP has optimisations for different models (linear, trees, etc..)

Shapley Additive Explanations (SHAP)

- TreeExplainer
  - Only for tree based models
  - Works with scikit-learn, xgboost, lightgbm, catboost
- KernelExplainer
  - Model agnostic explainer

SHAP - Tree Explainer API

1. Create a new explainer, with our model as argument
   - `explainer = TreeExplainer(my_tree_model)`
2. Calculate shap_values from our model using some observations
   - `shap_values = explainer.shap_values(observations)`
3. Use SHAP visualisation functions with our shap_values
   - `shap.force_plot(base_value, shap_values[0]) # local explanation`
   - `shap.summary_plot(shap_values) # Global features importance`

CAMBRIDGE SPARK

Hands-on session

>>> SHAP

Conclusion

- Gives trust that our complex model makes correct predictions in an ethical way
- Can help debugging our model and spot biases in our data
- Can explain to others why a prediction was made
- Regulations make it mandatory (finance, GDPR, …)