Interpretable ML

Part 2

Model-agnostic methods

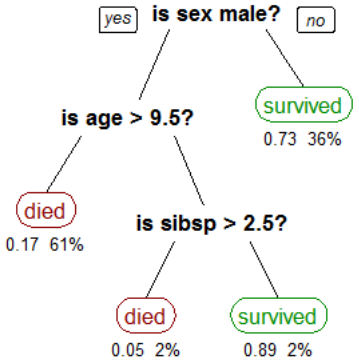
WHY DO WE NEED INTERPRETABLE ML?

- A user's trust is directly impacted by how much they can understand and predict the model's behavior, as opposed to treating it as a black box. (As well as a movie recommendation will be more trustful if the model can explain why is recommending that)
- As a valuable information to assess the model' predictions. With this information we can feedback to improve the model (feature engineering, parameter tuning, etc.)

SO, LET'S USE INTERPRETABLE MODELS!

- Decision trees, rules [1], additive models [2], or sparse linear models [3], among others.

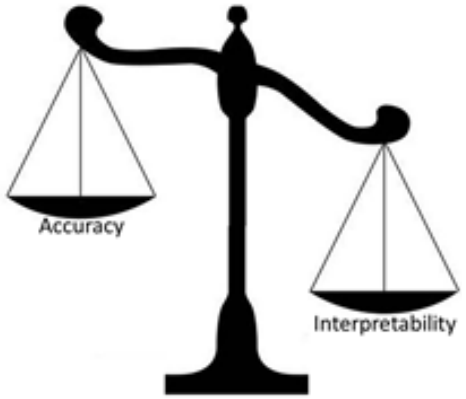
$\widehat{\beta_0}, \widehat{\beta_1}$

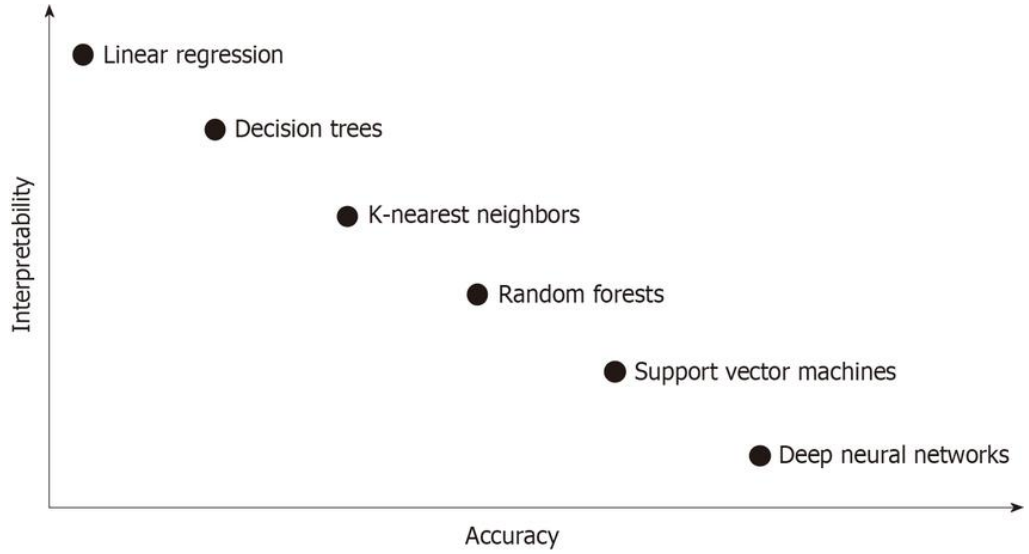


As long as the model is accurate for the task, and uses a reasonably restricted number of internal components (i.e. paths, rules, or features), such approaches provide extremely useful insights.

BUT… WAIT!

- A strong constraint as being interpretable translates into a less flexible, accurate model [1].

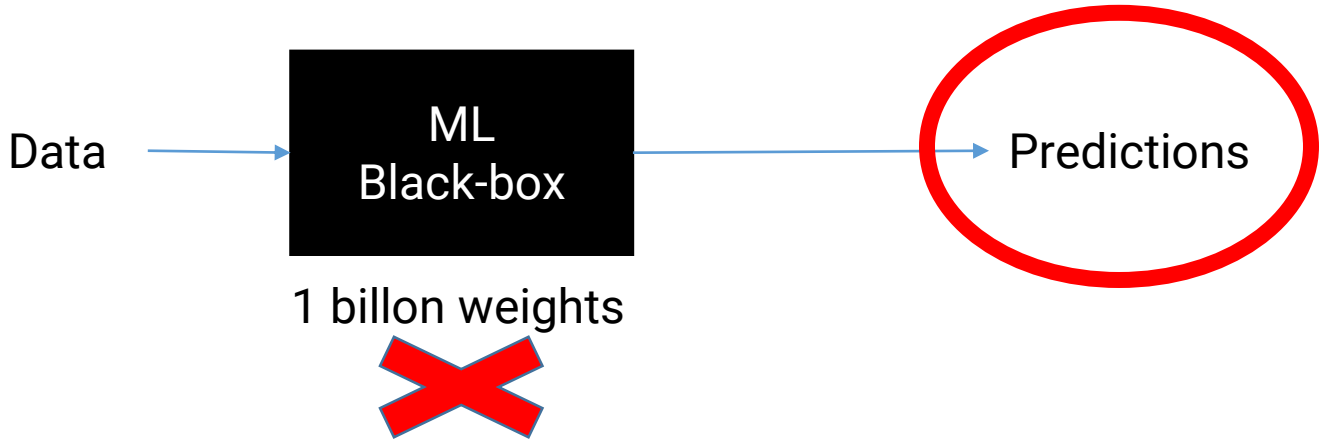




This trade-off between model flexibility and interpretability [2] implies one cannot use a model whose behavior is very complex, yet expect humans to fully comprehend it globally.

MODEL-AGNOSTIC APPROACH

Alternative approach: model-agnostic. i.e. to extract post-hoc explanations by treating the original model as a black box.



We learn an interpretable model on the predictions of the black box model [1,2], perturbing inputs and seeing how the black box model reacts

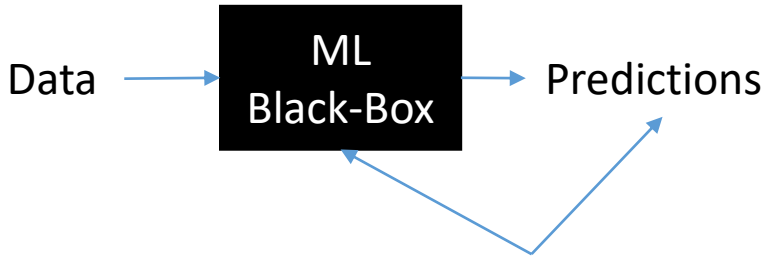
- We free the model of the interpretability constraint. We can go as expresable as necessary. (Deep Learning)

AIM OF THIS CHUNK

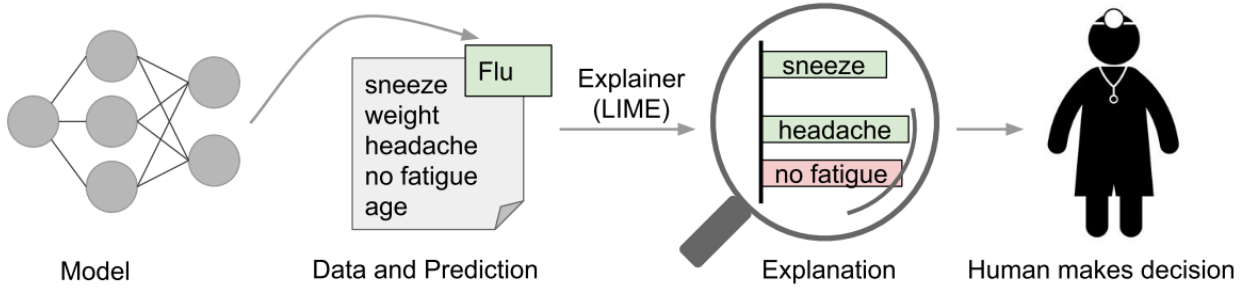
Introduce the method called Local interpretable Model-agnostic Explanations or LIME [1].

LIME

- LIME main goals is to explain the predictions of any classifier in an interpretable and faithful manner, **by learning an interpretable model locally around the prediction.**
- Even though an interpretable model may not be able to approximate the black box model globally, approximating it in the vicinity of an individual instance may be feasible.



Why this ML model predict this?
We want a human-understandable answer!



LIME – LOSS FUNCTION

We want the 'best agreement' for the predictions of the ML model f and the interpretable model g in the surroundings of the instance x

$$explanation(x) = argmin_{g \in G} \mathcal{L}(f, g, \pi_x)$$

- Where G is a class of interpretable models.
- π_x is a soft-rule of nearness of x
- We will estimate $\mathcal{L}$ by generating perturbed samples around x

Moreover, we want to penalize this interpretable model by the complexity of g.

$$explanation(x) = argmin_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

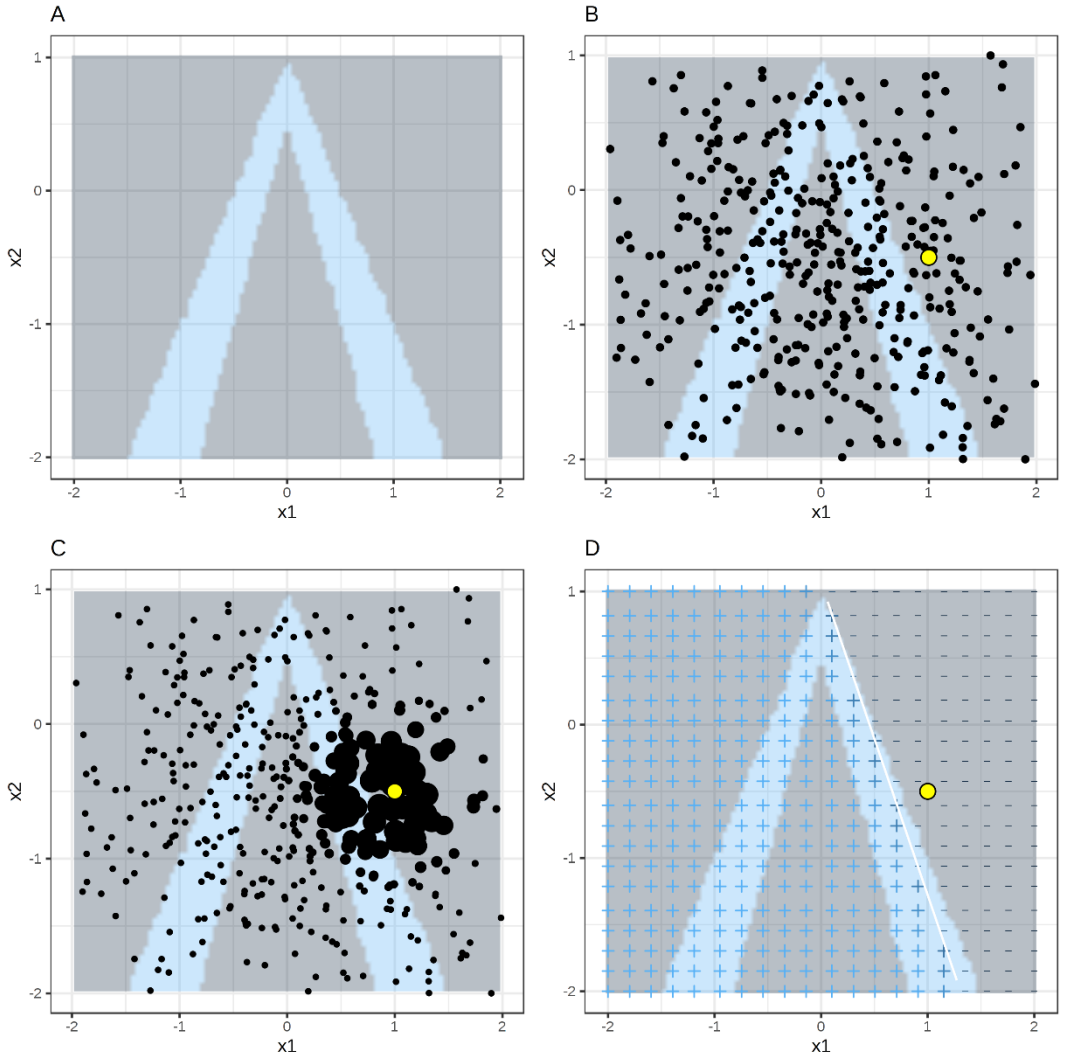
- Where $\Omega(g)$ is a measure of complexity of model g.

- This function will vary regarding the kind of data we are using (structured, text, images)
- In practice, LIME only optimizes the $\mathcal{L}$ function, while the complexity $\Omega(g)$ has to be determined by the user (specifying K)

LIME PIPELINE

Why model f predicted instance x as $\hat{f}_x$?

(1) Train the ML Black-box model.
(2) Select your instance of interest for which you want to have an explanation of its black box prediction.
(3) Perturb your dataset and get the black box predictions for these new points.
(4) Weight the new samples according to their proximity to the instance of interest.
(5) Train a weighted, interpretable model on the predictions of ML, with this new dataset. [The features can be different / a subset of the ones used by the ML model.]
(6) Explain the prediction by interpreting the local model.



LIME – TABULAR DATA

Features / Covariates

Target / Response

Instances/ Obs.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa

$$explanation(x) = argmin_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

- f: here will be a Random Forest.
- g: will be the class of linear models, such that $g(z) = w_g . z$
- $\mathcal{L}$ will be the locally weighted square loss:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z) . \left(f(z) - g(z')\right)^2$$

- π_x: be an exponential kernel defined on some distance function D, with width σ.
- $\Omega(g)$: number of features to explain the instance x. (predefined with K)

For multiple classes, we explain each class separately, thus f(x) is the prediction of the relevant class.

How we create the required disturbed dataset?

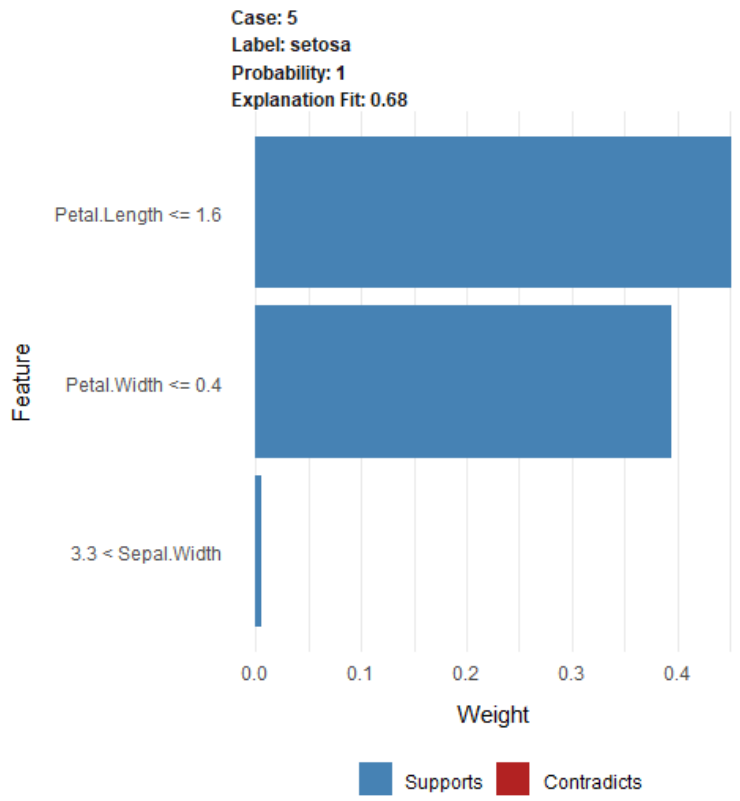
LIME – TABULAR DATA

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa

1. Fit a RF model
2. Select your instance of interest for which you want to have an explanation of its black box prediction.
3. Perturb your dataset and get the black box predictions for these new points.
4. Weight the new samples according to their proximity to the instance of interest.
5. We train a linear model (target= predictions), using K features, through minimizing the locally loss function $\mathcal{L}$
6. Explain the prediction by interpreting the local model.

Let's see this in R

LIME – TABULAR DATA - EXAMPLE



This would be interpreted as the following:

- Since Petal.Length is less than 1.6, this prediction is driven approx. 0.5 above than the average predicted probability in the whole sample.

From the figure it becomes clear that it is easier to interpret categorical features than numerical features. One solution is to categorize the numerical features into bins.

Discretized features make for more intuitive explanations.

LIME – TEXT

- LIME for text differs from LIME for tabular data. Variations of the data are generated differently.
- In this example we classify YouTube comments as spam or normal. (1 for spam, 0 normal comment)
- The black box model is a deep decision tree trained on the document word matrix.

	CONTENT	CLASS
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

	For	Christmas	Song	visit	my	channel!	;)	prob	weight
2	1	0	1	1	0	0	1	0.17	0.57
3	0	1	1	1	1	0	1	0.17	0.71
4	1	0	0	1	1	1	1	0.99	0.71
5	1	0	1	1	1	1	1	0.99	0.86
6	0	1	1	1	0	0	1	0.17	0.57

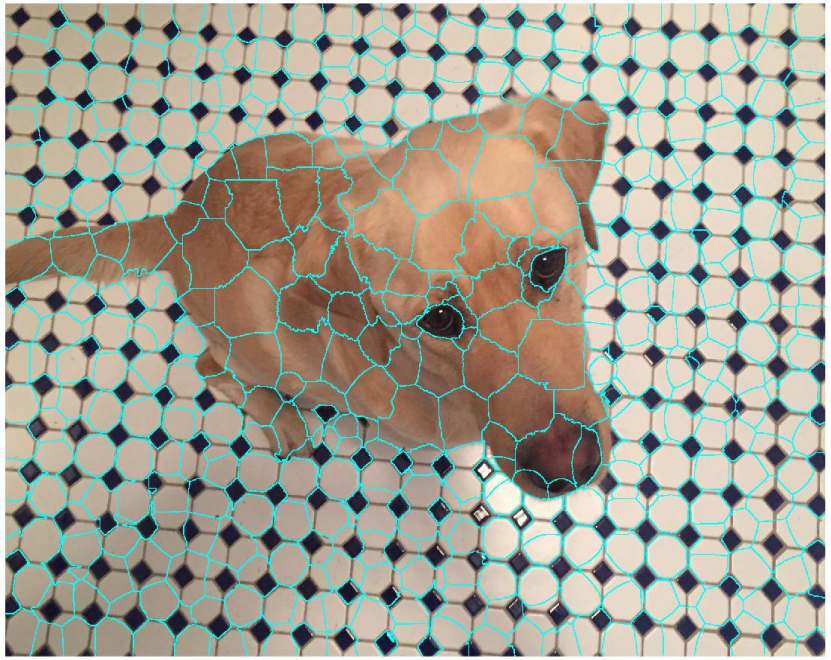
LIME – TEXT - EXAMPLE

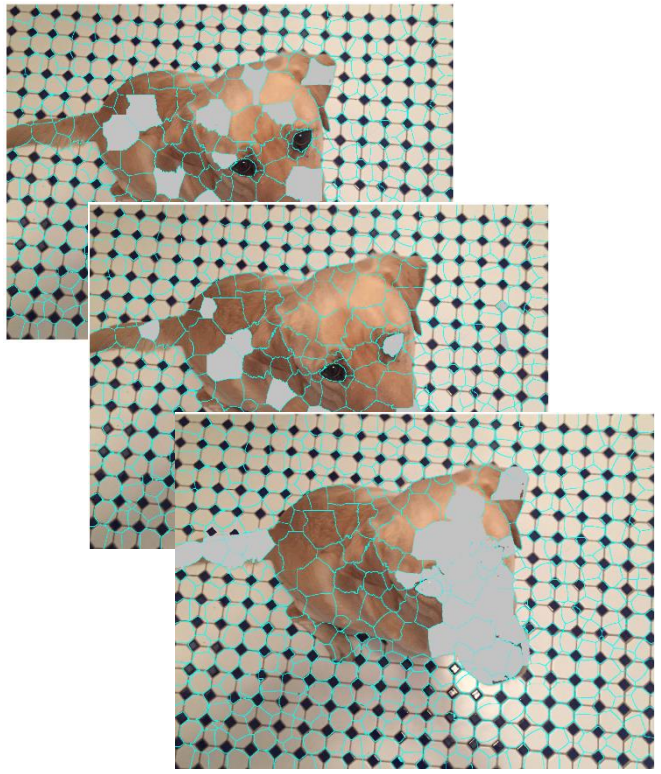
case	label_prob	feature	feature_weight
1	0.1701170	good	0.000000
1	0.1701170	a	0.000000
1	0.1701170	is	0.000000
2	0.9939024	channel!	6.180747
2	0.9939024	For	0.000000
2	0.9939024	;)	0.000000

LIME – IMAGES

- Here we have again a different situation.
- Intuitively, it would not make much sense to perturb individual pixels, since many more than one pixel contribute to one class
- We will select and turn on and off ‘superpixels’

Figure – Selecting 500 superpixels

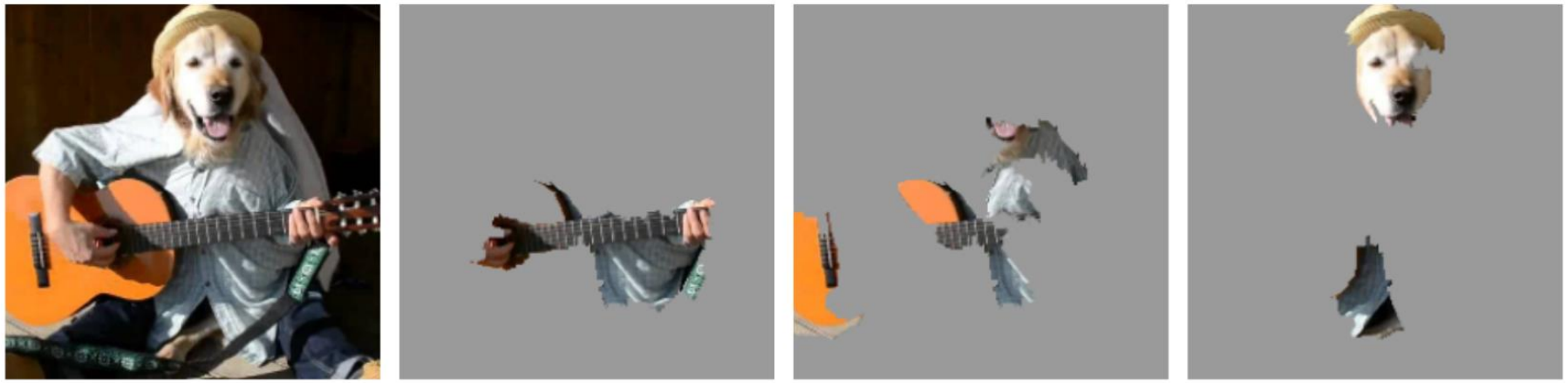




The number of superpixels will work as a measure for Ω. More superpixels will tend to more interpretable solution.

LIME – IMAGES - EXAMPLE

- Google's Inception neural network example.
- Since we can have several predicted labels per image (sorted by probability), we can explain the top n_labels. For the following image the top 3 predictions were electric guitar; acoustic guitar; and Labrador.

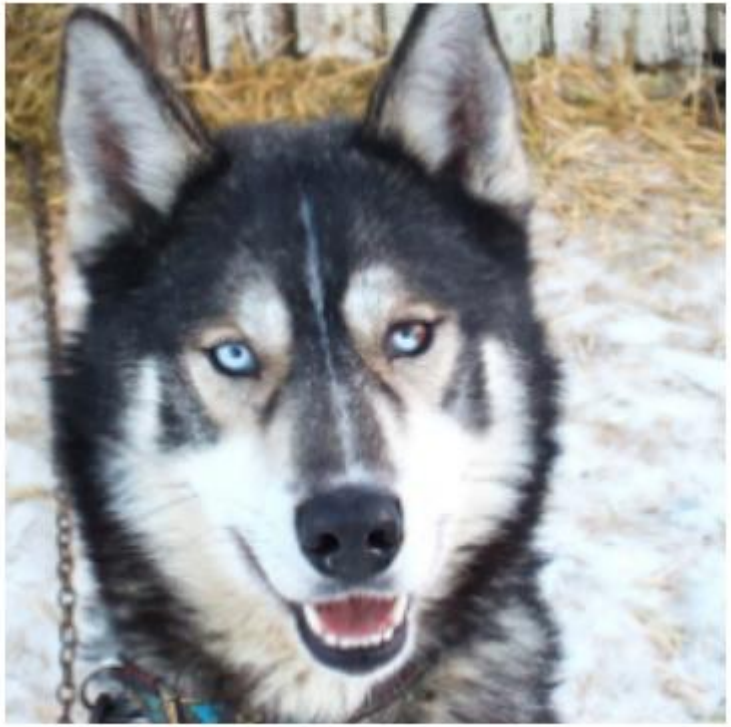


(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

The top 3 classes predicted are "Electric Guitar" (p = 0.32), "Acoustic guitar" (p = 0.24) and "Labrador" (p = 0.21)

LIME – IMAGES - EXAMPLE

- Here we have another example
- We have a model to classify Wolves and Huskes, which has high accuracy.
- But..



(a) Husky classified as wolf

(b) Explanation

LIME – SUMMARY

Advantages

- LIME is one of the few methods that works for tabular data, text and images.
- Easy to use.
- A regression model can rely on a non-interpretable transformation of some attributes, but the explanations can be created with the original attributes. (model trained with PCA variables and then represent through original variables)

Downsides:

- Interpretability achieved is sensible to the definition of the neighborhood (in tabular data).
- In tabular data: sampling could be improved in the current implementation of LIME. Data points are sampled from a Gaussian distribution, ignoring the correlation between features. This can lead to unlikely data points which can then be used to learn local explanation models.

FURTHER READING

- Model-agnostic methods:

https://christophm.github.io/interpretable-ml-book/agnostic.html

- LIME git:

https://github.com/thomasp85/lime

- **"Why Should I Trust You?": Explaining the Predictions of Any Classifier (LIME)**

https://arxiv.org/abs/1602.04938

- **Model-Agnostic Interpretability of Machine Learning**

https://arxiv.org/abs/1606.05386

- Tutorials, examples with LIME

https://github.com/marcotcr/lime