

MODEL INTERPRETATION: LIME AND SHAPLEY

KANRU WANG
JUNE 2019

MODEL INTERPRETATION *Agenda*

- Agenda
- Why is model interpretation important
- LIME
 - Idea
 - Global Surrogate vs Local Surrogate
 - LIME for images
 - LIME for text
 - LIME for tabular data
 - Pros and Cons
- Shapley Values
 - Idea
 - Calculation
 - Example
 - SHAP package
 - Pros and Cons
- References

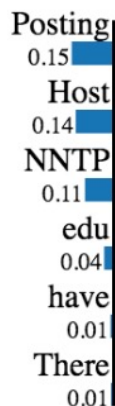
MODEL INTERPRETATION *Why is model interpretation important*

Prediction probabilities



atheism

christian



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.

The classifier predicts the instance correctly, but for the wrong reasons. The word "Posting" (part of the email header) appears in 21.6% of the examples in the training set, only two times in the class 'Christianity'. This is repeated on the test set, where it appears in almost 20% of the examples, only twice in 'Christianity'.

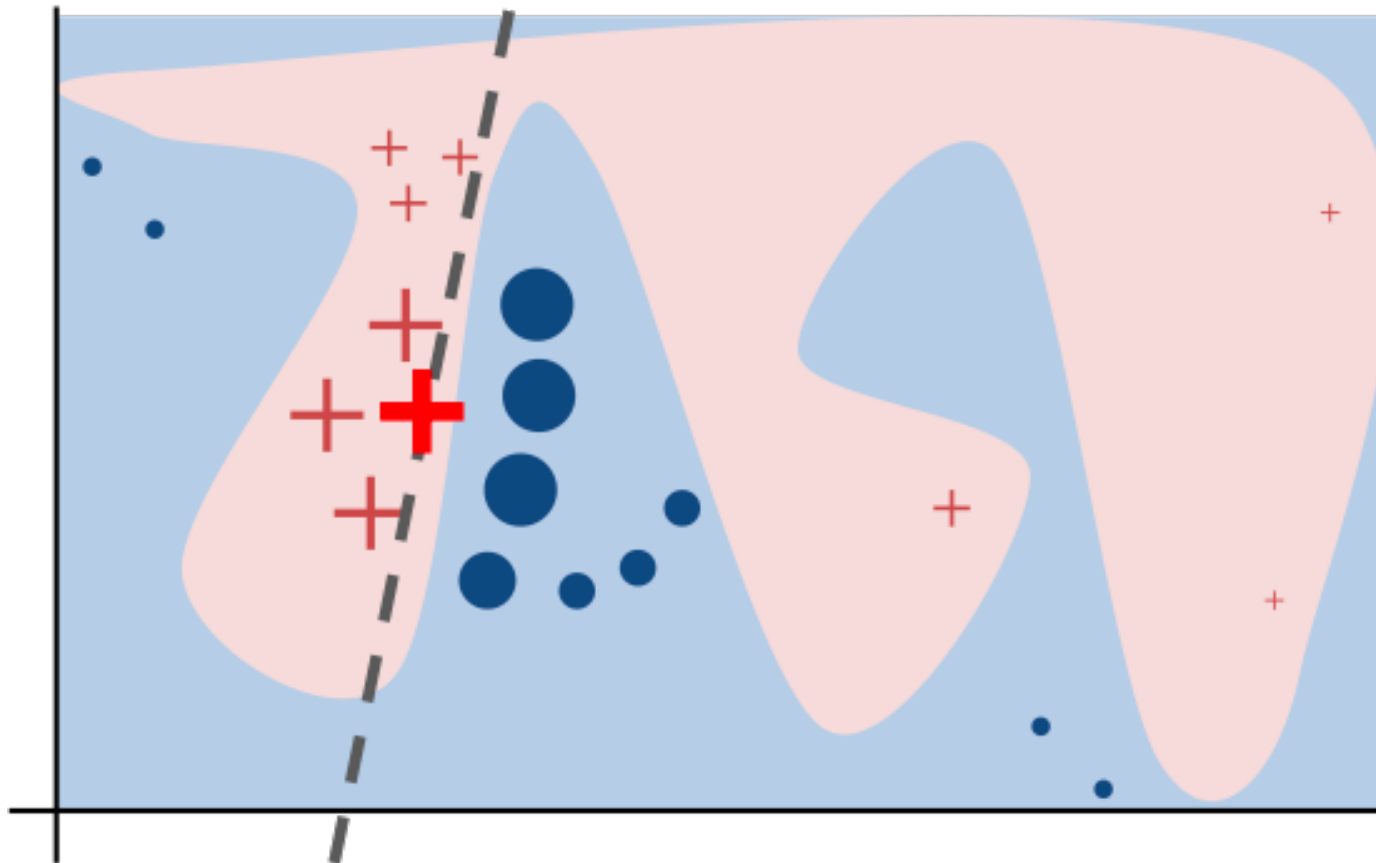
They correspond to a very sparse linear model (with only 6 features). Although the underlying classifier is a complicated random forest, in the neighborhood of this example it behaves roughly as a linear model. If we remove the words "Host" and "NNTP" from the example, the "atheism" prediction probability becomes close to $0.57 - 0.14 - 0.12 = 0.31$.

Local Interpretable Model-agnostic Explanations (LIME)



LIME Idea

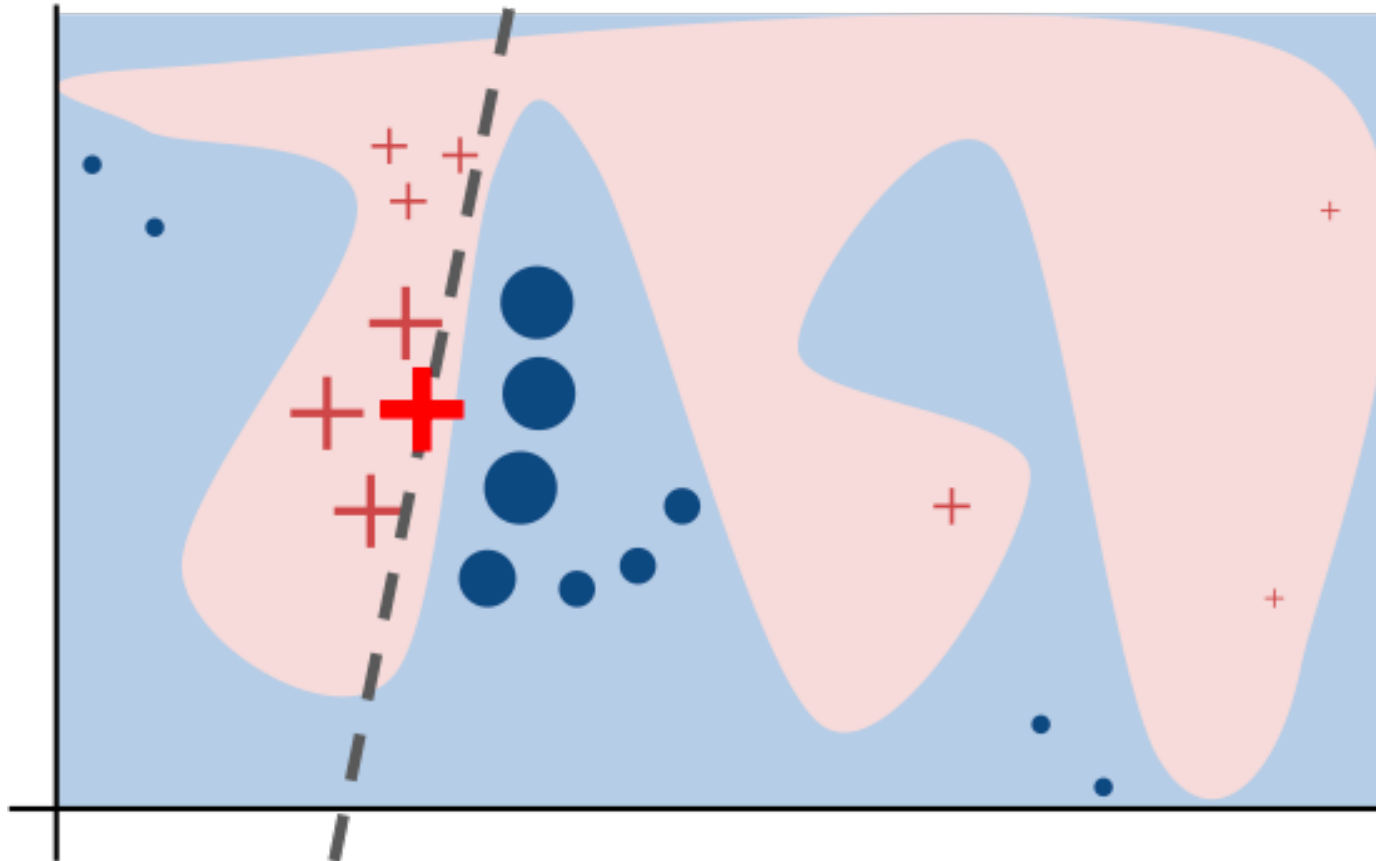
- Nonlinear decision function.
- Try to explain the bright red cross, X.
- Sample perturbed instances around X, and weight them according to their proximity to X.
- Get original model's prediction on these perturbed instances.
- Learn a linear model (dashed line) on the weighted perturbed instances
- The new model approximates the original model well at least in the vicinity of X.



LIME Idea

Linear regression can be chosen as interpretable surrogate model. In advance, you must select K , the number of features to have in the interpretable model. The lower K , the easier it is to interpret the model. A higher K potentially produces models with higher fidelity. There are several methods for training models with exactly K features. A good choice is Lasso.

Other strategies are forward or backward selection of features. This means either start with the full model (= containing all features) or with a model with only the intercept and then test which feature would bring the biggest improvement when added or removed, until a model with K features is reached.



LIME *Global Surrogate vs Local Surrogate*

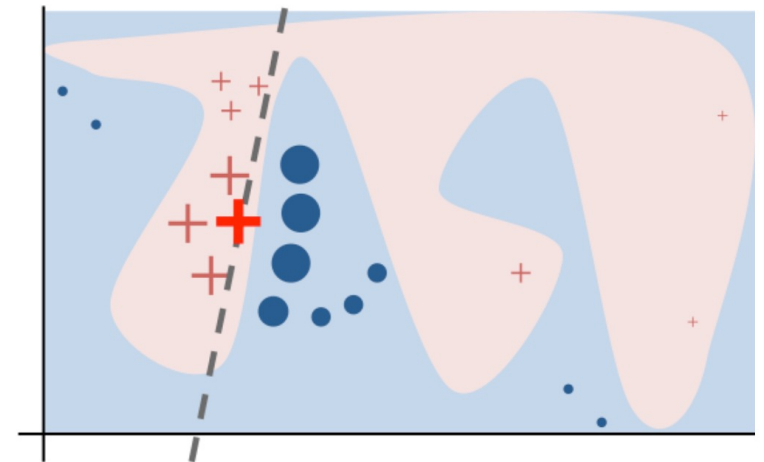
Global Surrogate Model

1. Choose a dataset. This could be the same dataset that was used for training the black box model, or a sample of it from the same distribution.
2. For the chosen dataset, get the predictions of your base black box model.
3. Choose an interpretable surrogate model (linear model, decision tree, ...).
4. Train the interpretable model on the dataset and the black model's predictions.
5. Measure how well the surrogate model replicates the prediction of the black box model. Interpret / visualize the surrogate model.

Instead of training a surrogate model using the full data, use a subset of the full data, and/or reweight the data

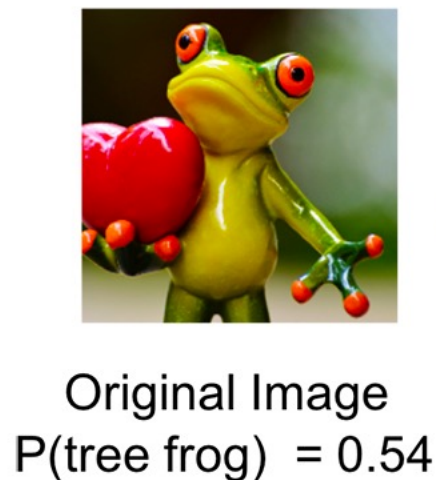
LIME, a Local Surrogate Model







1. Select your instance of interest for which you want to have an explanation of its black box prediction.
2. Perturb your dataset and get the black box predictions for these new points.
3. Weight the new samples according to their proximity to the instance of interest.
4. Train a weighted, interpretable model on the dataset. The y labels are from the black box predictions made above.
5. Explain the prediction by interpreting the local model.

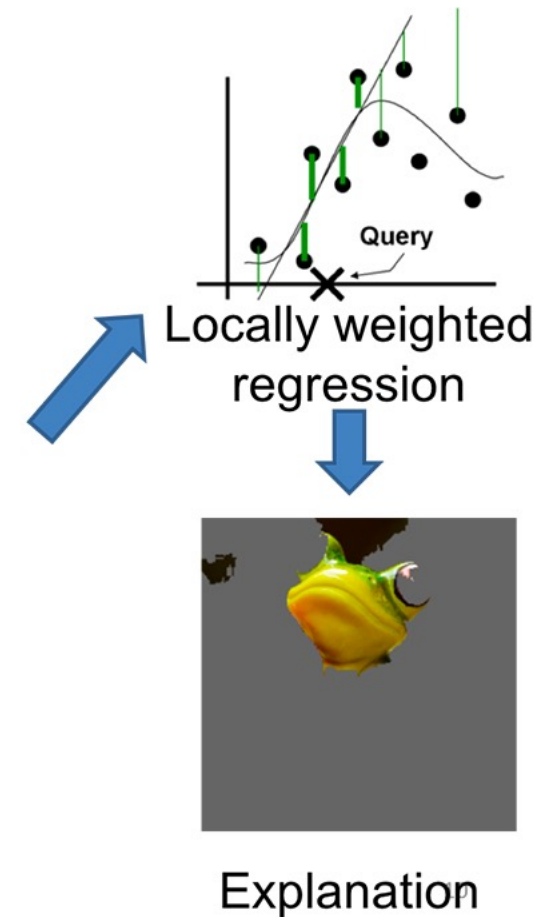


LIME for Image

- How do you get the variations of the data?
For images, the solution is to turn super-pixels on or off.
- Super-pixels are interconnected pixels with similar colors and can be turned off by replacing each pixel with for example grey.
- Relating to the previous example, here the original image is the “bright red cross”, each super-pixel is a feature, and multiple super-pixels can be turned on for one instance.
- Both image data and text data have binary features.



Perturbed Instances	$P(\text{tree frog})$
	 0.85
	 0.00001
	 0.52



LIME *for Image*

It does not make sense to perturb individual pixels, since more than one pixel contribute to one class. Randomly changing individual pixels usually do not change the predictions by much. Therefore, variations of the images are created by segmenting the image into “superpixels” and turning superpixels off or on. Superpixels are interconnected pixels with similar colors and can be turned off by replacing each pixel with a user-defined color such as grey. The user can also specify a probability for turning off a superpixel in each permutation. Superpixels can be obtained by an image segmentation algorithm such as quick shift.

We can generate a dataset of perturbed instances by turning some of the interpretable components “off” (in this case, making them grey).

For each perturbed instance, we can

1. Get the probability that a tree frog is in the image according to the model.
2. Learn a simple (linear) model on this data set, which is locally weighted—that is, we care more about making mistakes in perturbed instances that are more similar to the original image.
3. Present the superpixels with highest positive weights as an explanation, greying out everything else.

Also read: <https://towardsdatascience.com/understanding-how-lime-explains-predictions-d404e5d1829c>

LIME for Text

- How do you get the variations of the data?
For text, the solution is to turn single words on or off.
- In the example below, 1 is for spam, 0 is for normal comment.
- Both image data and text data have binary features.

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 for Text

Each column corresponds to one word in the sentence. Each row is a variation, 1 means that the word is part of this variation and 0 means that the word has been removed. The corresponding sentence for one of the variations is “Christmas Song visit my ;)”.

The “prob” column shows the predicted probability of spam for each of the sentence variations.

The “weight” column shows the proximity of the variation to the original sentence, calculated as 1 minus the proportion of words that were removed, for example if 1 out of 7 words was removed, the proximity is $1 - 1/7 = 0.86$.

	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

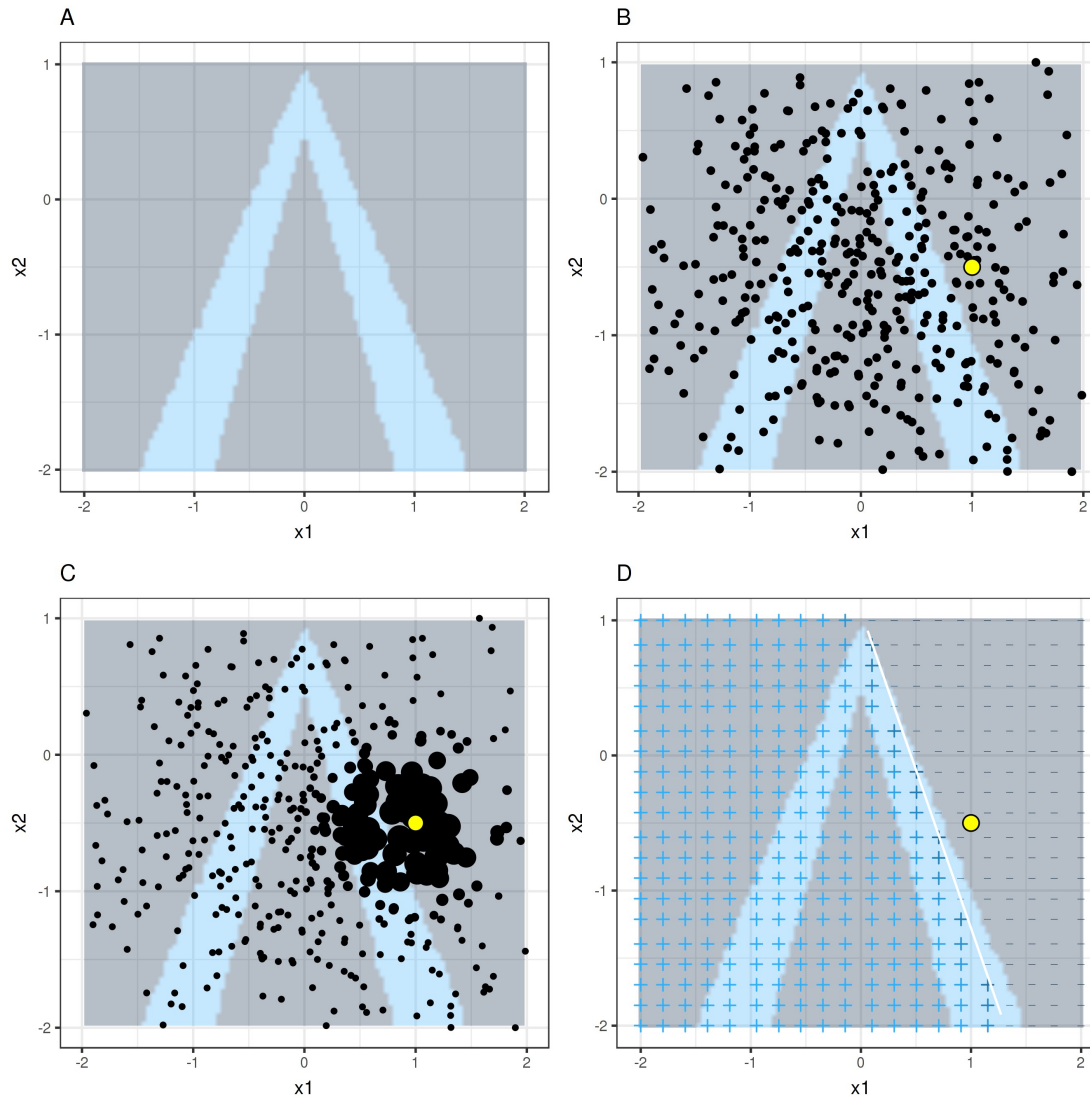
LIME for Text

- Two sentences are interpreted.
- The word “channel” indicates a high probability of spam.
- Notice that the feature_weight is not the weight of each sample, but the coefficient (or effect).

case	label_prob	feature	feature_weight
1	0.1701170	good	0.000000
1	0.1701170	PSY	0.000000
1	0.1701170	a	0.000000
2	0.9939024	channel!	6.180747
2	0.9939024	Song	0.000000
2	0.9939024	Christmas	0.000000

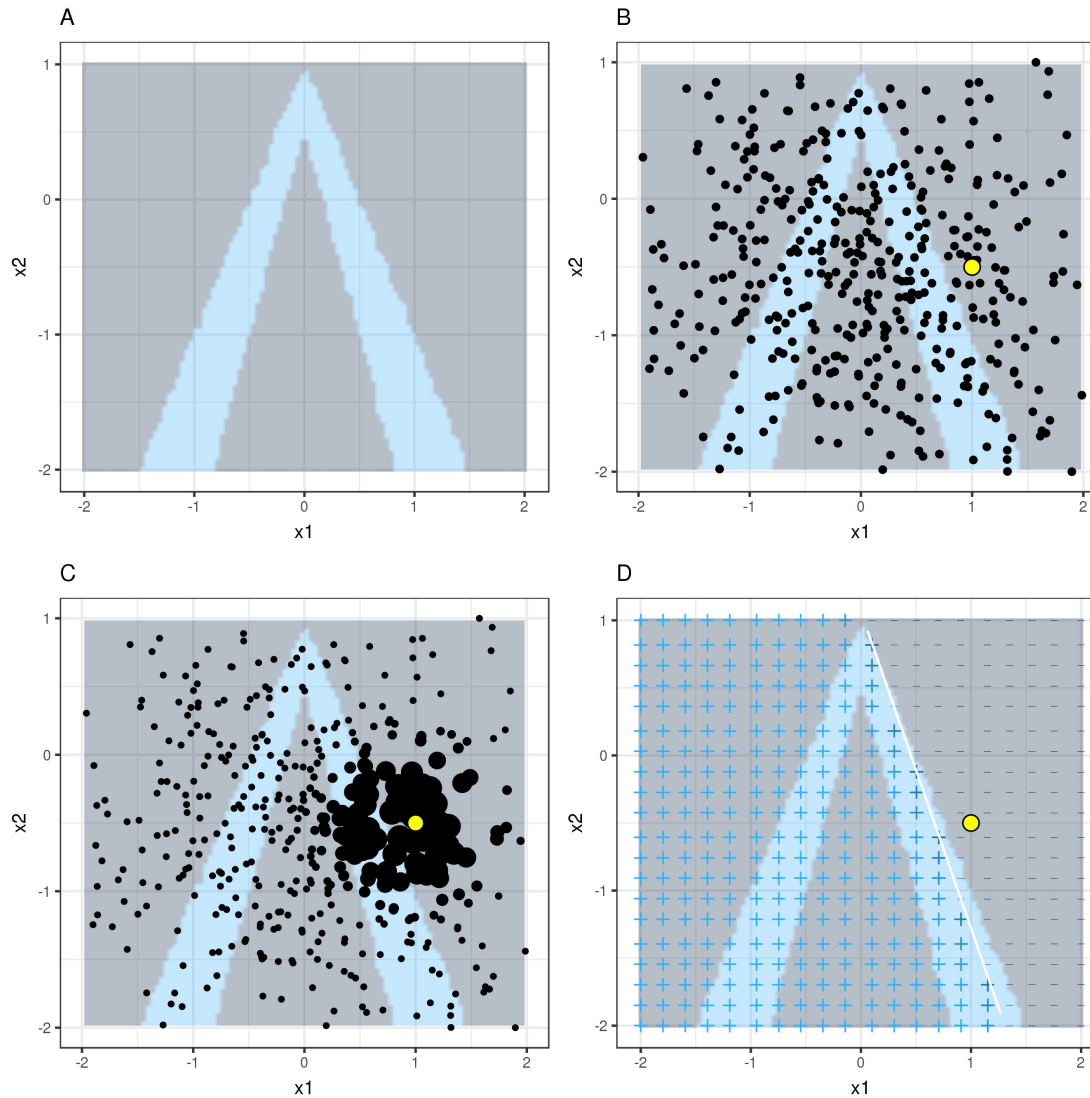
LIME for Tabular Data

- How do you get the variations of the data?
For tabular data, LIME perturbs each feature individually, and draws from a normal distribution with mean and standard deviation taken from the feature.
- Samples are not drawn around the instance of interest, but from the training data's mass center.



LIME for Tabular Data

- A) Random forest predictions given features x_1 and x_2 . Predicted classes: 1 (dark) or 0 (light).
- B) Instance of interest (big dot) and data sampled from a normal distribution (small dots).
- C) Assign higher weight to points near the instance of interest.
- D) Signs of the grid show the classifications of the locally learned model from the weighted samples. The white line marks the decision boundary ($P(\text{class}=1) = 0.5$).



LIME for Tabular Data

```
# Create Random Forest model on iris data
model <- train(iris_train, iris_lab, method = 'rf')

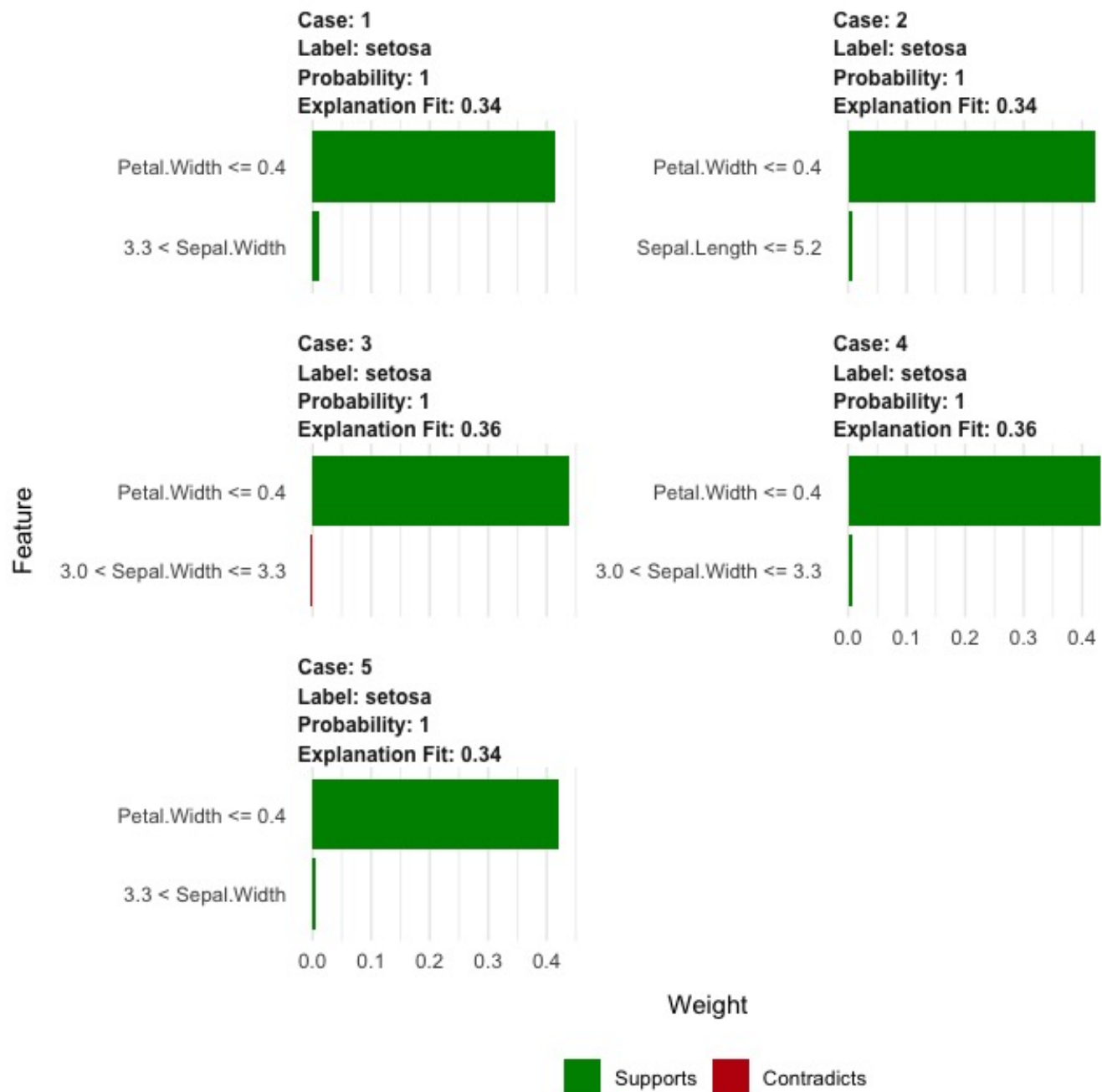
# Create an explainer object
explainer <- lime(iris_train, model)

# Explain new observation
explanation <- explain(iris_test, explainer, n_labels = 1, n_features = 2)

# The output is provided in a consistent tabular format and includes the
# output from the model.
explanation
#> # tibble [10 × 13]
#>   model_type case  label label_prob model_r2 model_intercept
#>   <chr>      <chr> <chr>      <dbl>    <dbl>         <dbl>
#> 1 classific... 1     seto...      1    0.340         0.263
#> 2 classific... 1     seto...      1    0.340         0.263
#> 3 classific... 2     seto...      1    0.336         0.259
#> 4 classific... 2     seto...      1    0.336         0.259
#> 5 classific... 3     seto...      1    0.361         0.258
#> 6 classific... 3     seto...      1    0.361         0.258
#> 7 classific... 4     seto...      1    0.364         0.247
#> 8 classific... 4     seto...      1    0.364         0.247
#> 9 classific... 5     seto...      1    0.343         0.256
#> 10 classific... 5     seto...      1    0.343         0.256
#> # ... with 7 more variables: model_prediction <dbl>, feature <chr>,
#> #   feature_value <dbl>, feature_weight <dbl>, feature_desc <chr>,
#> #   data <list>, prediction <list>
```

- model_r2 stands for the quality of the model used for the explanation (i.e. R square).
- One model for each case to be explained.
- Both Python and R packages are available.
- Python (lime and Skater) and R (lime package and iml package).

LIME for Tabular Data



- model_r2 stands for the quality of the model used for the explanation (i.e. R square).
- One model for each case to be explained.

LIME Pros

- Suppose the people looking at the explanations understand decision trees the best. Because you use local surrogate models, you can use decision trees as explanations.
- A text classifier can rely on word embeddings as features, but the explanation can be based on the presence or absence of words in a sentence.

A regression model can rely on a non-interpretable transformation of some attributes, but the explanations can be created with the original attributes.

- Local fitness measure can be obtained.

LIME *Cons*

- Kernel

The definition of the neighborhood is a big and unsolved problem when using LIME with tabular data.

The kernel width determines how large the neighborhood is: A small kernel width means that an instance must be very close to influence the local model, a larger kernel width means that instances that are farther away also influence the model.

For each application need to try different kernel settings and see if the explanations make sense.

- Instability

The explanations of two very close points varied greatly in a simulated setting. If you repeat the sampling process, the explanations that come out can be different.

SHAPLEY VALUES

Shapley Values

Pronounced /'ʃæpli/

A prediction can be explained by assuming that each feature value of the instance is a “player” in a game where the prediction is the payout.

The Shapley value tells us how to fairly distribute the “payout” among the features. E.g. Marketing Attribution

SHAPLEY VALUES *Idea*

Meeting the following conditions will mean the game is 'fair' according to Shapley values:

1. The sum of what everyone receives should equal the total reward
2. If two people contributed the same value, then they should receive the same amount from the reward
3. Someone who contributed no value should receive nothing
4. If a game is composed of two subgames, then an individual's reward from the full game should equal their reward from their first game plus their reward from the second game

An intuitive way to understand the Shapley value:

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 is the **average marginal contribution** of a feature value across **all possible coalitions**. The Shapley value is NOT the difference in prediction when we would remove the feature from the model.

Without Shapley Value, usually when two features are similar, the first entered feature is more attributed, the second entered feature is less attributed.

SHAPLEY VALUES *Calculation*

Calculation of Shapley Value:

Finding each player's marginal contribution, averaged over every possible sequence in which the players could have been added to the group.

$$\phi_i(p) = \sum_{S \subseteq N/i} \frac{|S|!(n-|S|-1)!}{n!} (p(S \cup i) - p(S))$$

Similar to the inverse of the combination formula as the weight of this grouping

Importance of $i = p(\text{with } i) - p(\text{without } i)$

$p(\cdot)$ is the prediction of the model.

ϕ_i is the contribution of the i -th feature on the prediction p .

S is a subset of all features, excluding i . It is a set of any size, drawn from the full set.

N is the full set of features.

$|S|$ is the size of that subset.

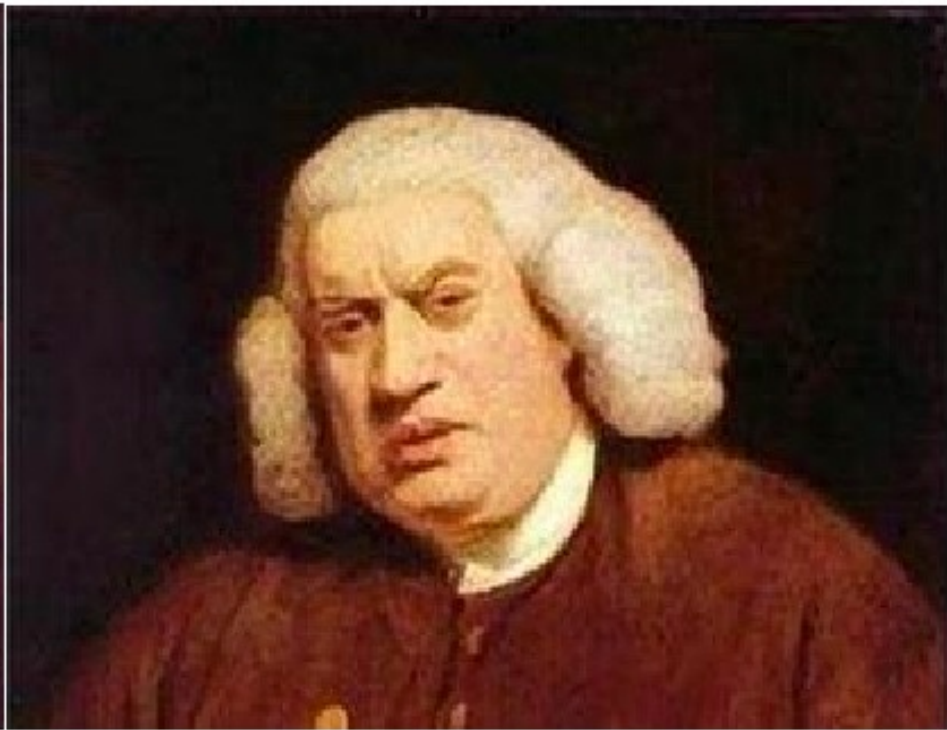
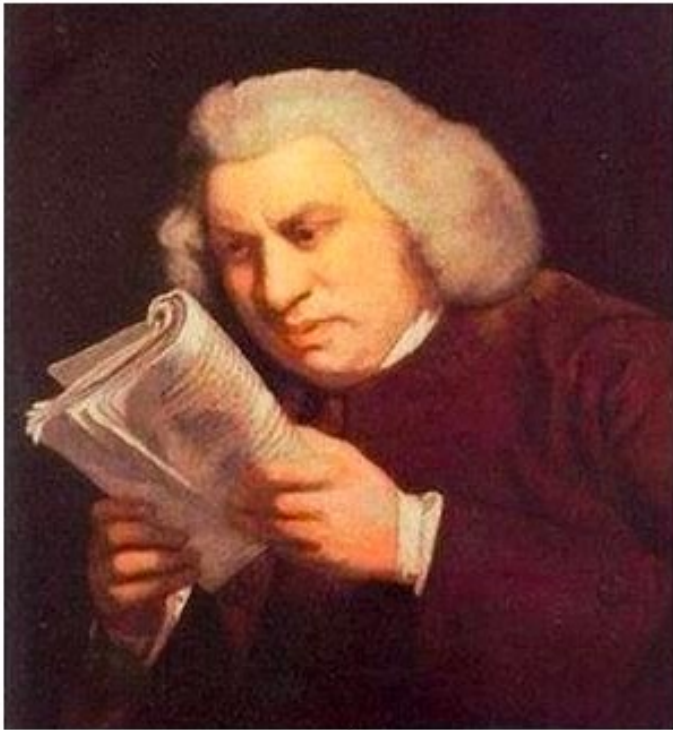
n is number of features, i.e. the size of set N .

SHAPLEY VALUES *Calculation*

Calculation of Shapley Value:

Finding each player's marginal contribution, averaged over every possible sequence in which the players could have been added to the group.

$$\phi_i(p) = \sum_{S \subseteq N/i} \frac{|S|!(n - |S| - 1)!}{n!} (p(S \cup i) - p(S))$$



SHAPLEY VALUES *Calculation*

Calculation of Shapley Value:

Finding each player's marginal contribution, averaged over every possible sequence in which the players could have been added to the group.

$$\phi_i(p) = \sum_{S \subseteq N/i} \frac{|S|!(n-|S|-1)!}{n!} (p(S \cup i) - p(S))$$

Assume that A, B, C are three categorical features.

Before A joins (i.e. S)	When A (i.e. i) joins S	Full sequence (i.e. N)	A's contribution	Weight
{ }	{A}	{A,B,C}	80 - 0 = 80	0!2! / 3! = 2/6
{ }	{A}	{A,C,B}		
{B}	{B,A}	{B,A,C}	80 - 56 = 24	1!1! / 3! = 1/6
{C}	{C,A}	{C,A,B}	85 - 70 = 15	1!1! / 3! = 1/6
{B,C}	{B,C,A}	{B,C,A}	90 - 72 = 18	2!0! / 3! = 2/6
{C,B}	{C,B,A}	{C,B,A}		

Lookup table	
Set	Prediction
{A}	80
{B}	56
{C}	70
{A,B}	80
{A,C}	85
{B,C}	72
{A,B,C}	90

Feature A's contribution: $80 * 2 / 6 + 24 * 1 / 6 + 15 * 1 / 6 + 18 * 2 / 6 = 39.2$

SHAPLEY VALUES *Example*

Interpret Classification Models:

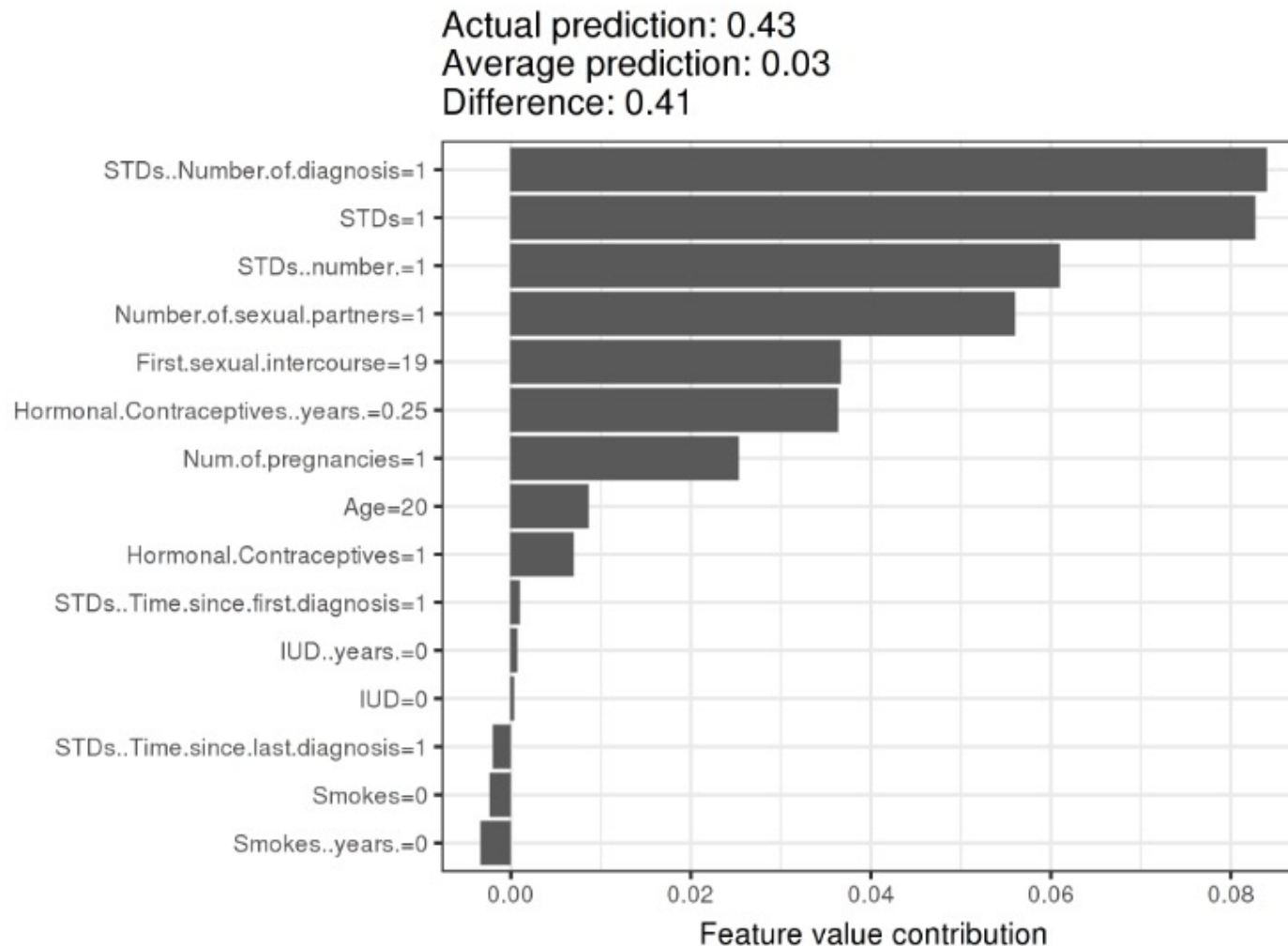
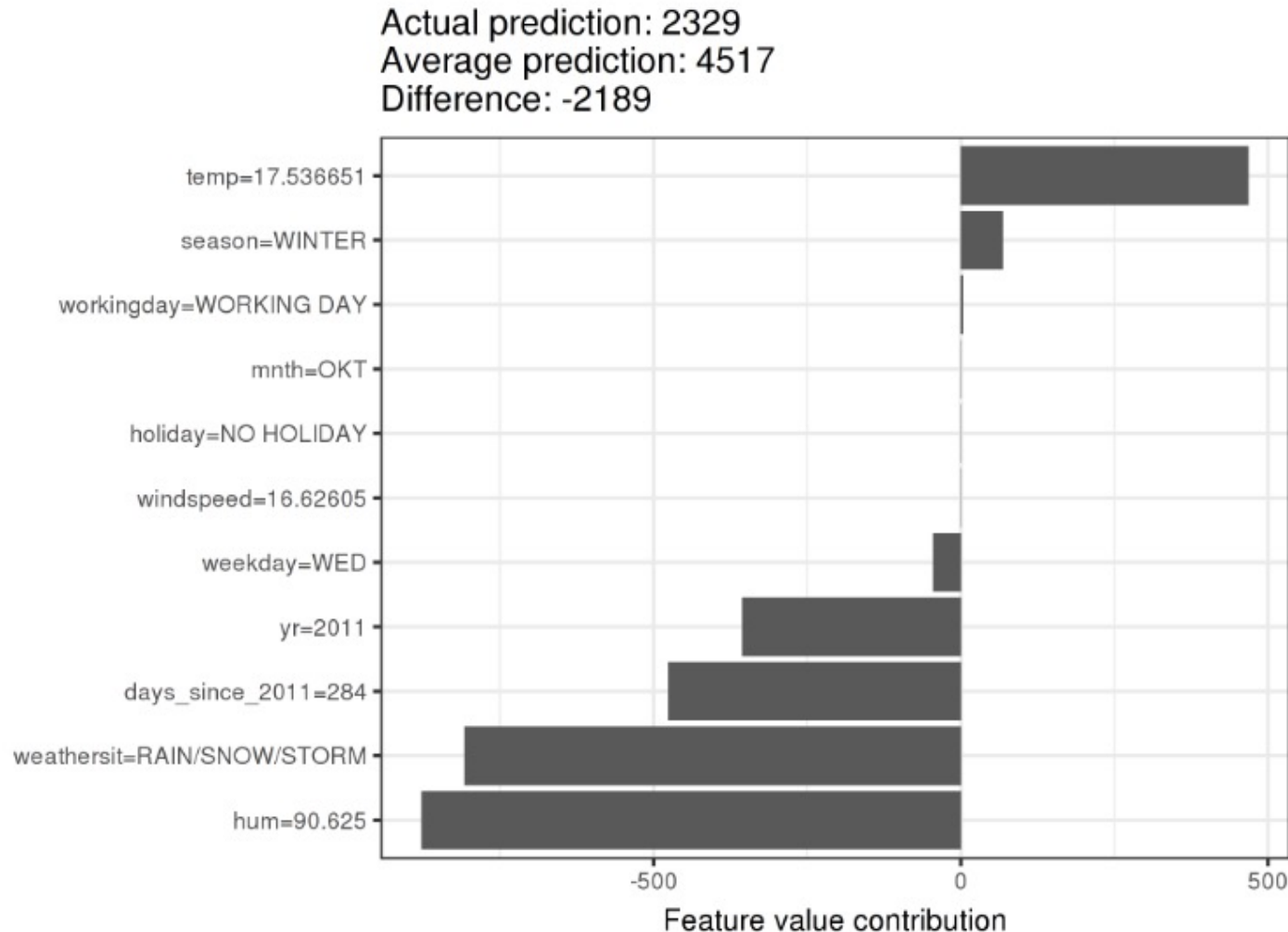


FIGURE 5.40: Shapley values for a woman in the cervical cancer dataset. With a prediction of 0.43, this woman's cancer probability is 0.41 above the average prediction of 0.03. The number of diagnosed STDs increased the probability the most. The sum of contributions yields the difference between actual and average prediction (0.41).

SHAPLEY VALUES *Example*

Interpret Regression Models:



Be careful to interpret the Shapley value correctly: The Shapley value is the average contribution of a feature value to the prediction in different coalitions. The Shapley value is NOT the difference in prediction when we would remove the feature from the model.

FIGURE 5.41: Shapley values for day 285. With a predicted 2329 rental bikes, this day is -2189 below the average prediction of 4517. The weather situation and humidity had the largest negative contributions. The temperature on this day had a positive contribution. The sum of Shapley values yields the difference of actual and average prediction (-2189).

SHAPLEY VALUES *SHAP*

SHAP library

Going through all possible combinations of features is computationally unfeasible.

The SHAP library calculates Shapley values significantly faster than if a model prediction had to be calculated for every possible combination of features.

It calculates Shapley Values by developing model specific algorithms, which take advantage of different model's structures. For instance, SHAP's integration with gradient boosted decision trees takes advantage of the hierarchy in a decision tree's features to calculate the SHAP values.

The results of SHAP are sparse (many Shapley values are estimated to be zero), which is the biggest difference from the classic Shapley values.

SHAP values have been added to the XGBoost library in Python.

Python package: shap

R package: iml, breakdown

SHAPLEY VALUES *Pros and Cons*

Pros

- LIME does not guarantee that the prediction is fairly distributed among the features, while the Shapley value might be the only method to deliver a full explanation.
- In situations where the law requires explainability, the Shapley value might be the only legally compliant method, because it is based on a solid theory and distributes the effects fairly.

Cons

- The accurate meaning of Shapley value may be hard to understand.
- The Shapley value returns a simple value per feature, but not prediction model like LIME. This means it cannot be used to make statements about changes in prediction for changes in the input, such as: “If I were to earn \$300 more a year, my credit score would increase by 5 points.”

MODEL INTERPRETATION *References*

LIME

Important reading:

- <https://christophm.github.io/interpretable-ml-book/lime.html>
- <https://cran.r-project.org/web/packages/lime/readme/README.html>
- <https://github.com/marcotcr/lime>

Other reading:

- <https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime>
- <https://homes.cs.washington.edu/~marcotcr/blog/lime/>

Shapley Values

Important reading:

- <https://christophm.github.io/interpretable-ml-book/shapley.html>
- <https://www.youtube.com/watch?v=w9O0fkfMkx0>

Other reading:

- <https://towardsdatascience.com/one-feature-attribution-method-to-supposedly-rule-them-all-shapley-values-f3e04534983d>
- <https://medium.com/@gabrieltseng/interpreting-complex-models-with-shap-values-1c187db6ec83>