Explainable Models
A Primer

By

Hardeep Arora

Based on book - https://christophm.github.io/interpretable-ml-book/

About Me

- 18+ years experience in Banking and Finance (MBA Finance, B.E. Computers)
- Have worked as Developer, DBA, Data Scientist, and Leading Analytics functions at Bank.
- Have delivered Data Warehouses and BI platforms for banks.
- Have worked in Marketing Analytics
- Currently Heading COE AI/Analytics @ Accenture focusing on Finance and Risk domain in ASEAN region.
- Advisor for some startup's on AI Strategy
- Work Life : Credit Risk and Financial Crime
- Night Life : Deep Learning, Blockchain, AGI, Kaggle

Why Explainability is important?

- **Why** is more important than **What** in certain cases.
  - The treatment work on the patient as opposed to the user liked the movie we recommended.
- We need explanations when things don't work the way we understand them. I was rejected on loan, I need to know why?
- Understanding why is the part of learning and help forward the scientific endeavors.
- Interpretability is not that important if the model **has no significant impact**.
- Interpretability is not that important when the **problem is well-studied**.

Key questions

- **Algorithm transparency**
  - *How does the algorithm create the model?*
- **Global, Holistic Model Interpretability**
  - *How does the trained model make predictions?*
- **Global Model Interpretability on a Modular Level**
  - *How do parts of the model influence predictions?*
- **Local Interpretability for a Single Prediction**
  - *Why did the model make a specific decision for an instance?*
- **Local Interpretability for a Group of Prediction**
  - *Why did the model make specific decisions for a group of instances?*

Key Approaches

Intrinsic interpretability means selecting and training a machine learning model that is considered to be intrinsically interpretable (for example short decision trees).

Post hoc interpretability means selecting and training a black box model (for example a neural network) and applying interpretability methods after the training (for example measuring the feature importance).

Dataset 1 – Bike Sharing (Regression)

- season : spring (1), summer (2), autumn (3), winter (4).
- holiday : Binary feature indicating if the day was a holiday (1) or not (0).
- yr: The year (2011 or 2012).
- days_since_2011: Number of days since the 01.01.2011 (the first day in the dataset). This feature was introduced to account for the trend, in this case that the bike rental service became more popular over time.
- workingday : Binary feature indicating if the day was a workingday (1) or weekend / holiday (0).
- weathersit : The weather situation on that day
  - Clear, Few clouds, Partly cloudy, Cloudy
  - Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  - Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
  - Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
- temp : Temperature in degrees Celsius.
- hum: Relative humidity in percent (0 to 100).
- windspeed: Wind speed in km per hour.
- cnt: Count of total rental bikes including both casual and registered. The count was used as the target in the regression tasks.

Dataset 2 – Risk Factors for Cervical Cancer (Classification)

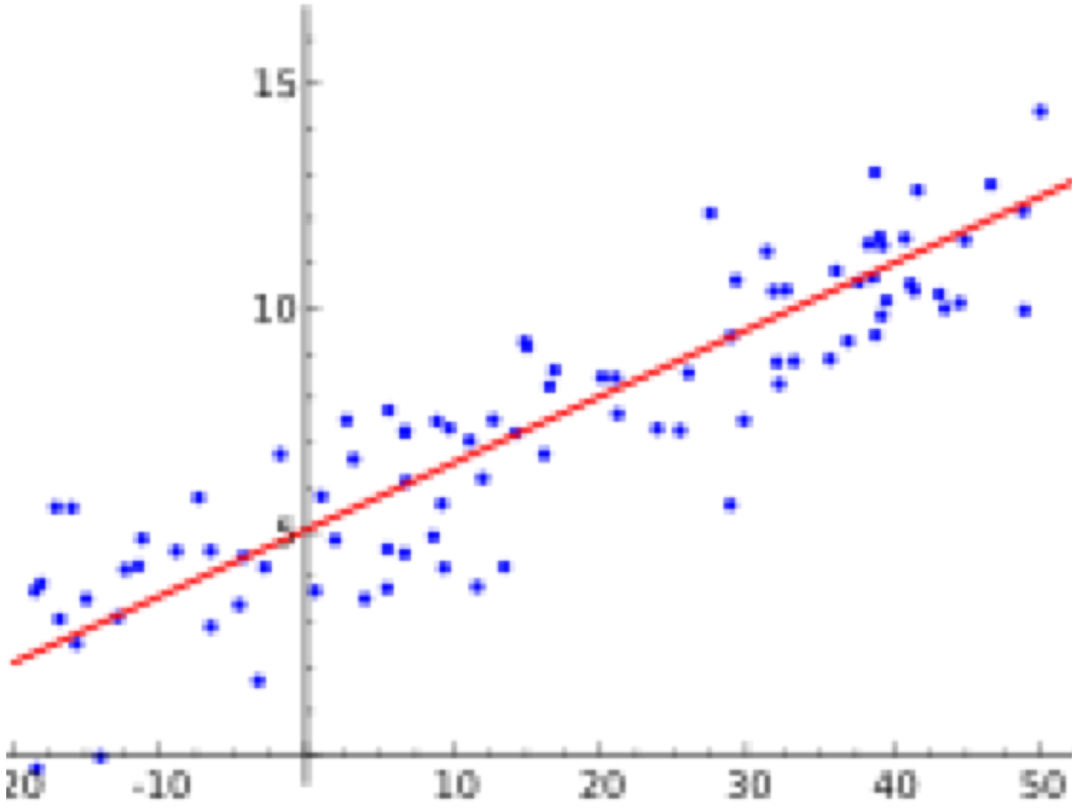
- Age in years
- Number of sexual partners
- First sexual intercourse (age in years)
- Number of pregnancies
- Smokes yes (1) or no (1)
- Smokes (years)
- Hormonal Contraceptives yes (1) or no (0)
- Hormonal Contraceptives (years)
- IUD: Intrauterine device yes (1) or no (1)
- IUD (years): Number of years with an intrauterine device
- STDs: Ever had a sexually transmitted disease? Yes (1) or no (0)
- STDs (number): Number of sexually transmitted diseases.
- STDs: Number of diagnosis
- STDs: Time since first diagnosis
- STDs: Time since last diagnosis
- Biopsy: Biopsy results “Healthy” or “Cancer”. Target outcome.

Intrinsic - Interpretable Models

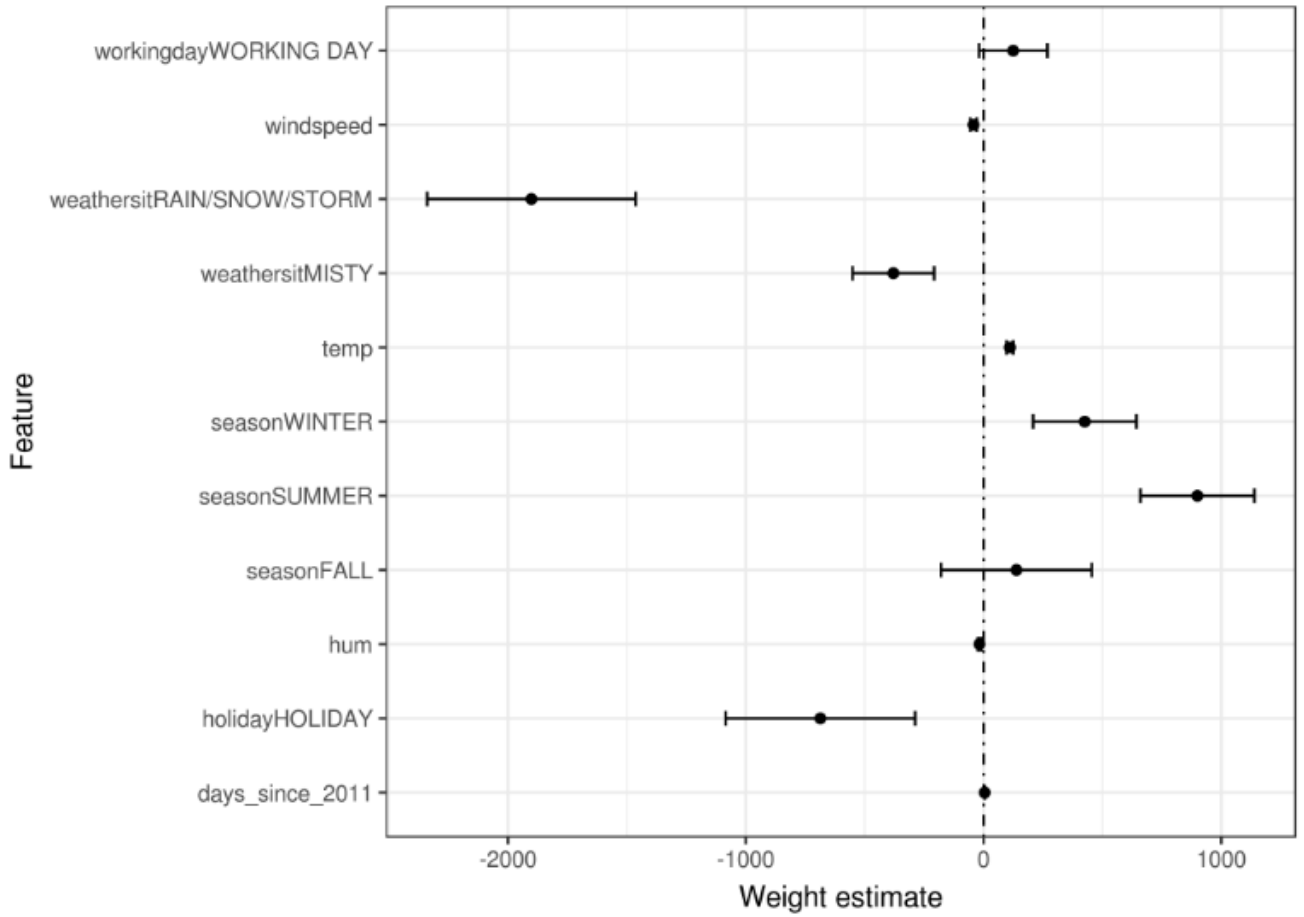
Algorithm	Linear	Monotone	Interaction	Task
Linear models	Yes	Yes	No	Regr.
Logistic regression	No	Yes	No	Class.
Decision trees	No	No	Yes	Class. + Regr.
RuleFit	Yes	No	Yes	Class. + Regr.
Naive Bayes	Yes	Yes	No	Class.n
k-nearest neighbours	No	No	No	Class. + Regr.

Linear Model -Interpretability

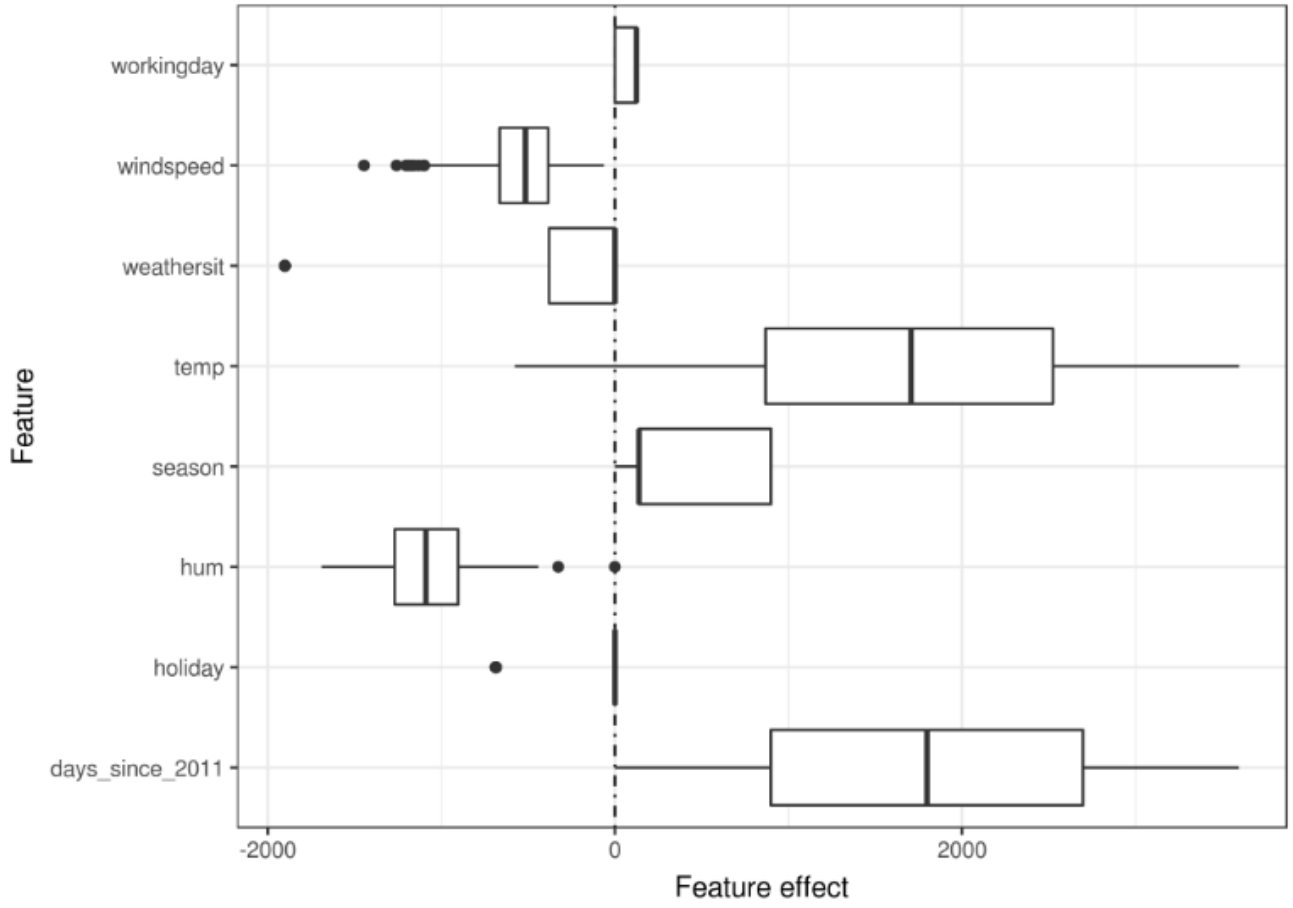
	Weight estimate	Std. Error
(Intercept)	2399.4	238.3
seasonSUMMER	899.3	122.3
seasonFALL	138.2	161.7
seasonWINTER	425.6	110.8
holidayHOLIDAY	-686.1	203.3
workingdayWORKING DAY	124.9	73.3
weathersitMISTY	-379.4	87.6
weathersitRAIN/SNOW/STORM	-1901.5	223.6
temp	110.7	7.0
hum	-17.4	3.2



Weights Plot

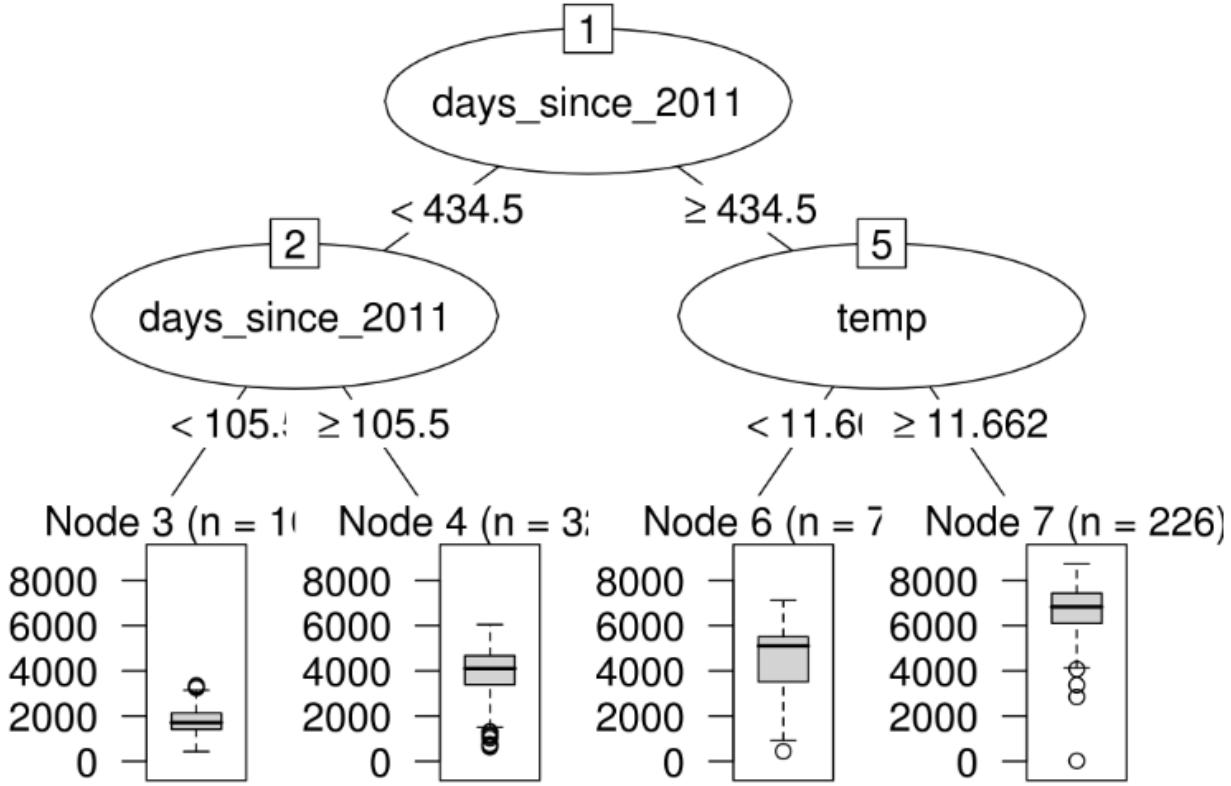


Effects Plot



Decision Trees

- Perfectly suited to **cover interactions** between features in the data
- Has a **natural visualization**, with its nodes and edges
- Trees **create good explanations**



Model-Agnostic Methods

Model flexibility

Not being tied to an underlying particular machine learning model. The method should work for random forests as well as deep neural networks.

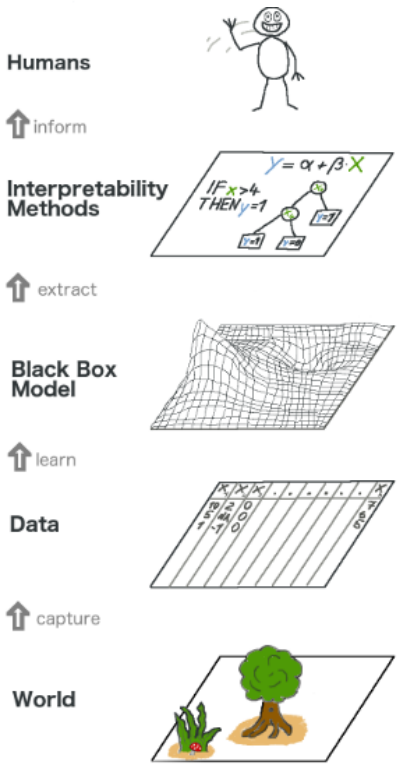
Explanation flexibility

Not being tied to a certain form of explanation. In some cases it might be useful to have a linear formula in other cases a decision tree or a graphic with feature importance's.

Representation flexibility

The explanation system should not have to use the same feature representation as the model that is being explained. For a text classifier that uses abstract word embedding vectors it might be preferable to use the presence of single words for the explanation.

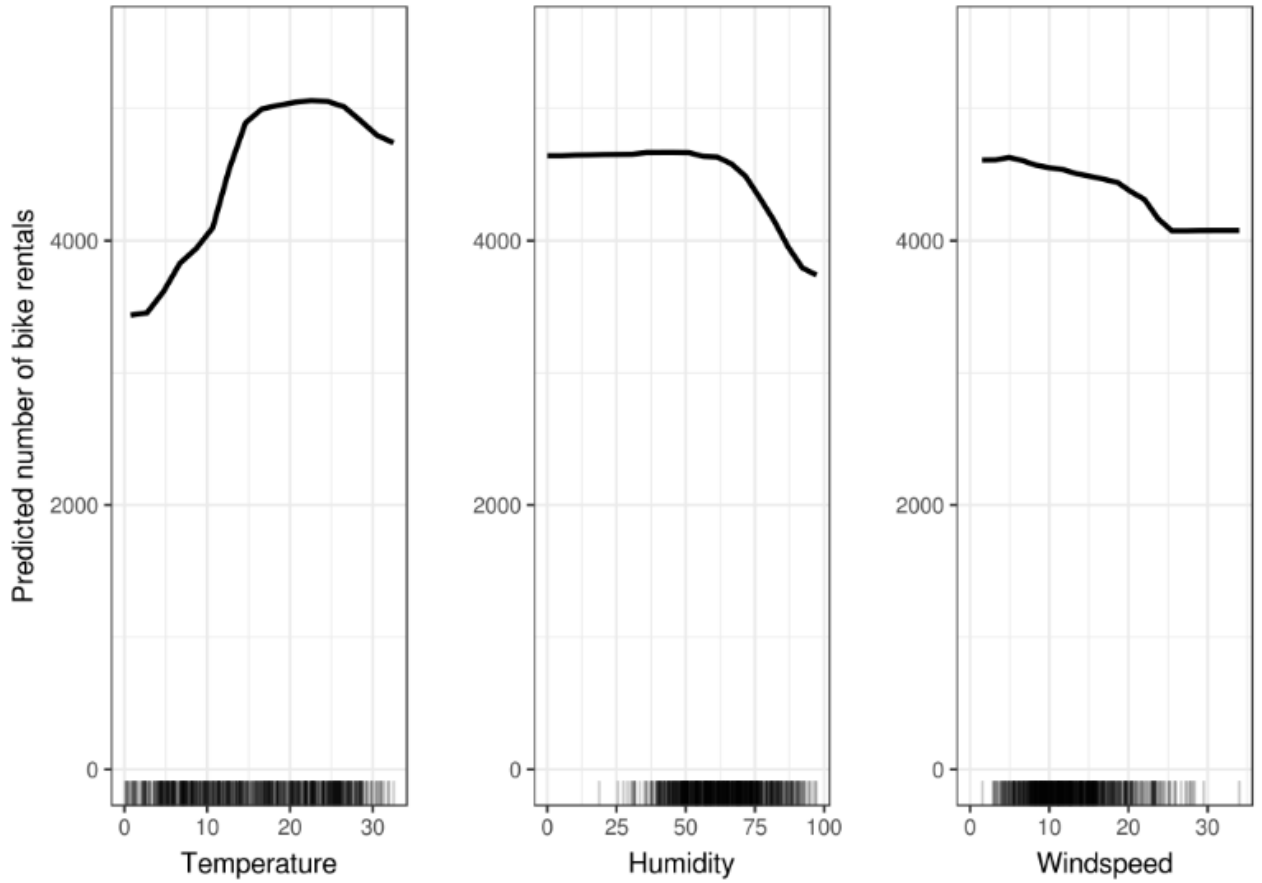
Simple Overview



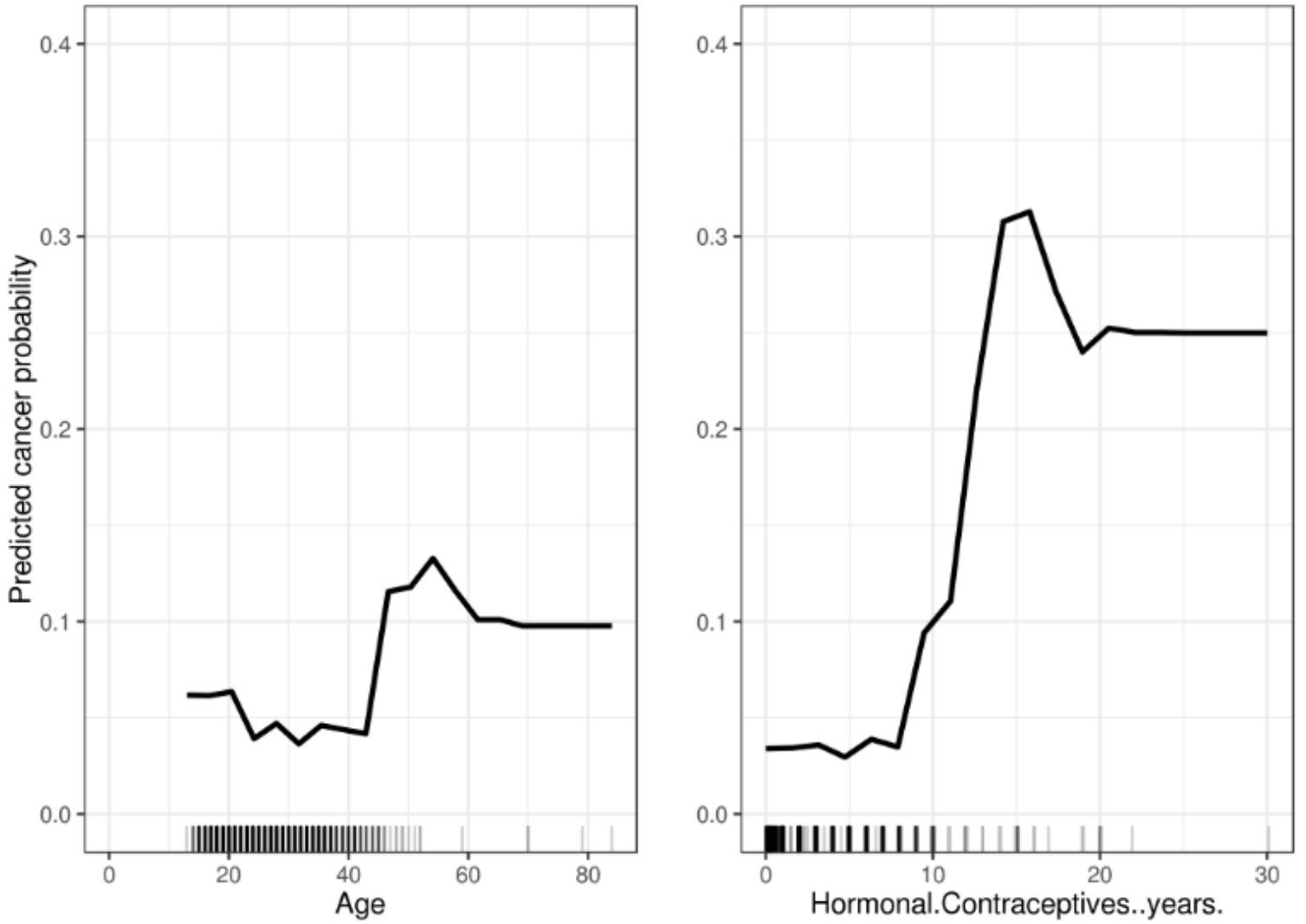
- The last layer is occupied by a ‘Human’. Humans are the consumers of the explanations, ultimately.
- On top of the ‘Black Box Model’-layer is the ‘Interpretability Methods’-layer that helps us deal with the opaqueness of machine learning models. What were the important features for a particular diagnosis? Why was a financial transaction classified as fraud?
- By fitting machine learning models on top of the ‘Data’-layer we get the ‘Black Box Model’-layer. Machine learning algorithms learn with data from the real world to make predictions or find structures.
- The second layer is the ‘Data’-layer. We have to digitalize the ‘World’ in order to make it processable for computers and also to store information. The ‘Data’-layer contains anything from images, texts, tabular data and so on.
- The ‘World’-layer contains everything that can be observed and is of interest. Ultimately we want to learn something about the ‘World’ and interact with it.

Partial Dependence Plot (PDP)

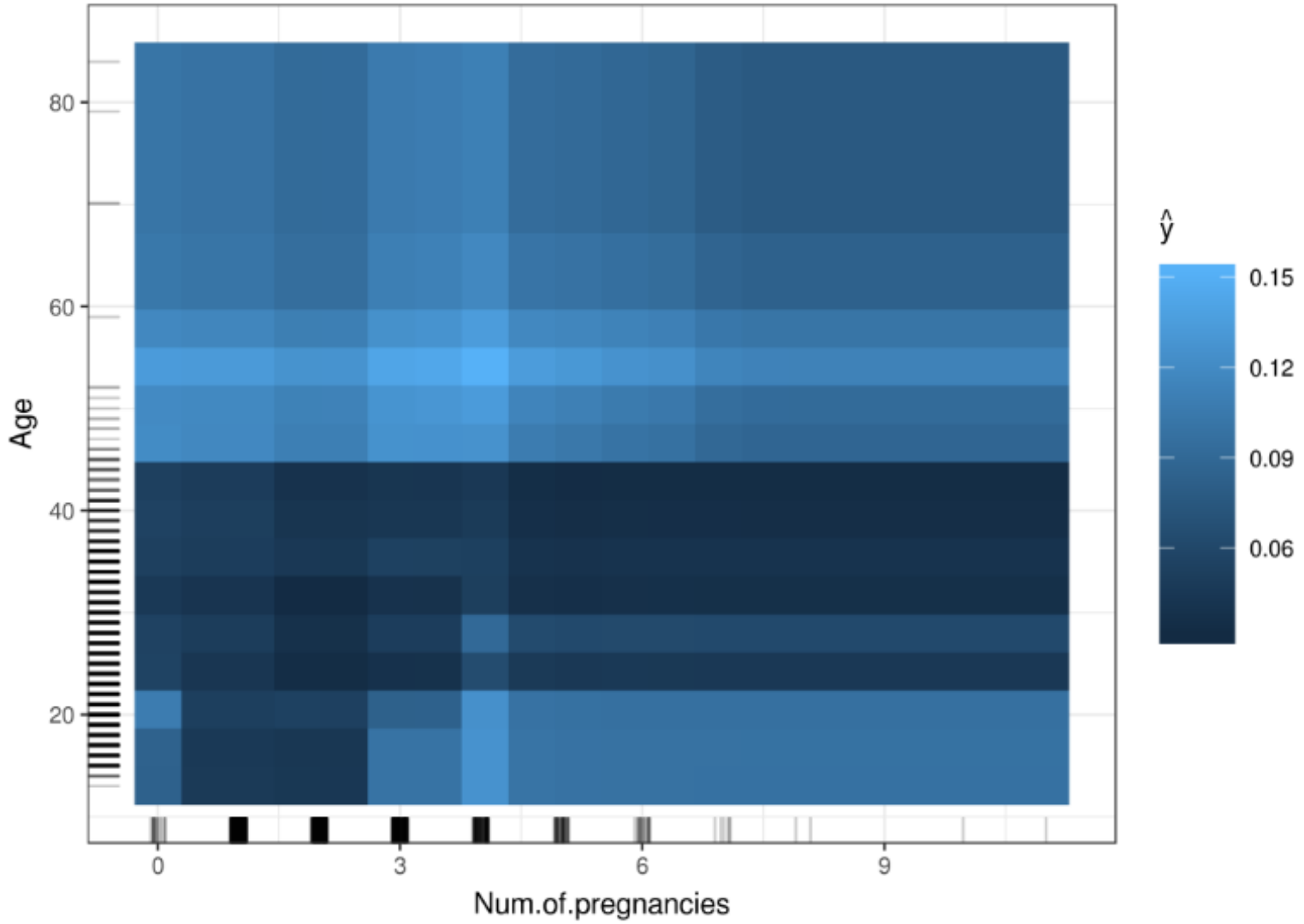
- The partial dependence plot shows the marginal effect of a feature on the predicted outcome of a previously fit model



PDP - Cervical Cancer Vs Age/H Contra



PDP - Cervical Cancer Vs Age/# of Pregnancies



PDP Pro's and Con's

Pro's

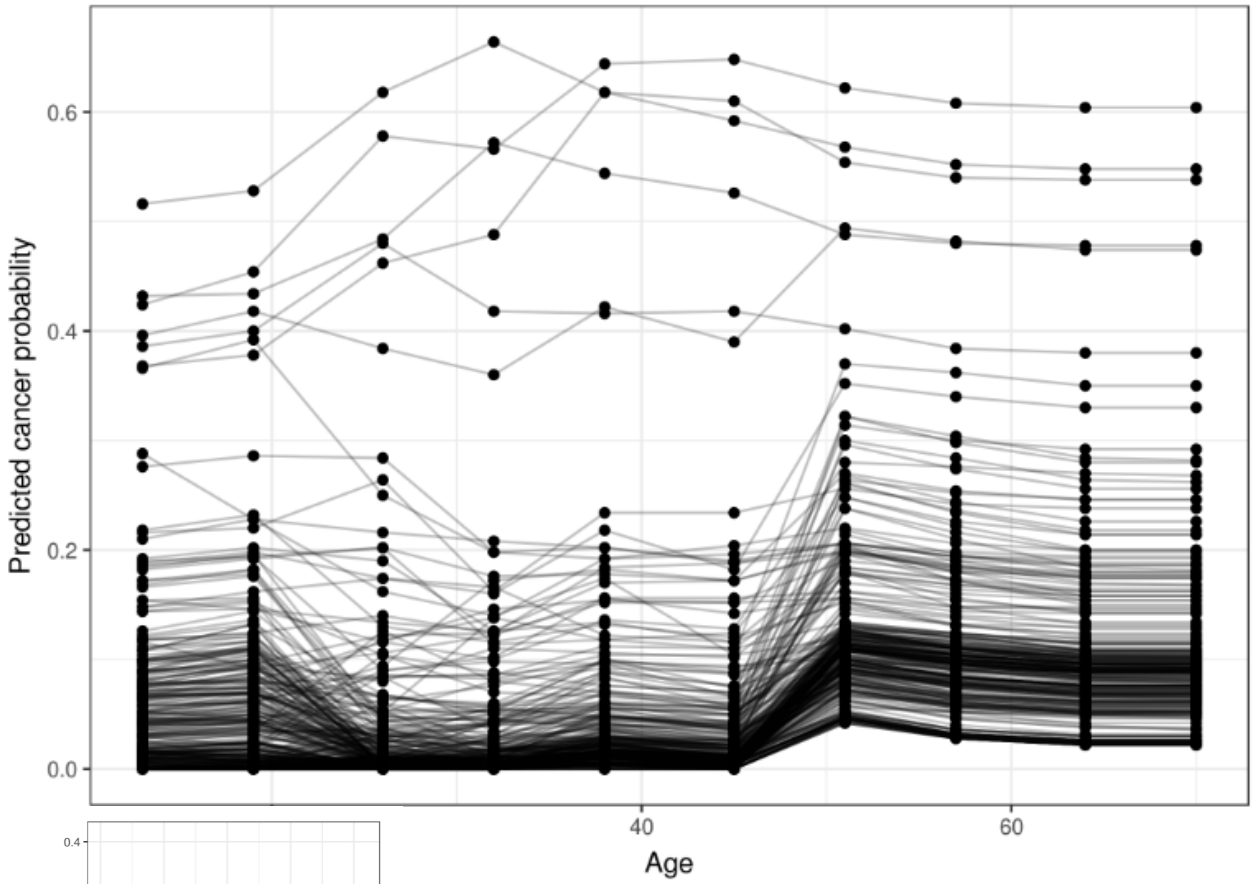
- The computation of partial dependence plots is **intuitive**
- If the feature for which you computed the PDP is uncorrelated with the other model features, then the PDPs are perfectly representing how the feature influences the target on average.
- Partial dependence plots are **simple to implement**.
- **Causal interpretation**

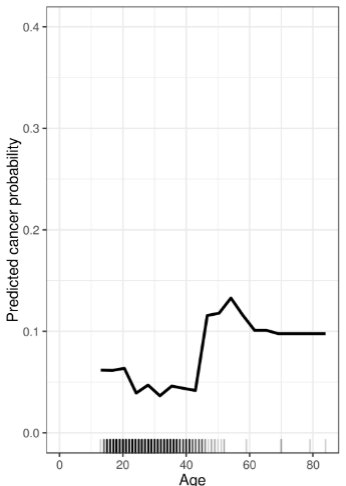
Con's

- The **maximum number of features** you can look at jointly is - realistically - two and - if you are stubborn and pretend that 3D plots on a 2D medium are useful - three.
- Some PD visualizations don't include the **feature distribution**
- The **assumption of independence** poses the biggest issue.
- **Heterogenous effects might be hidden**

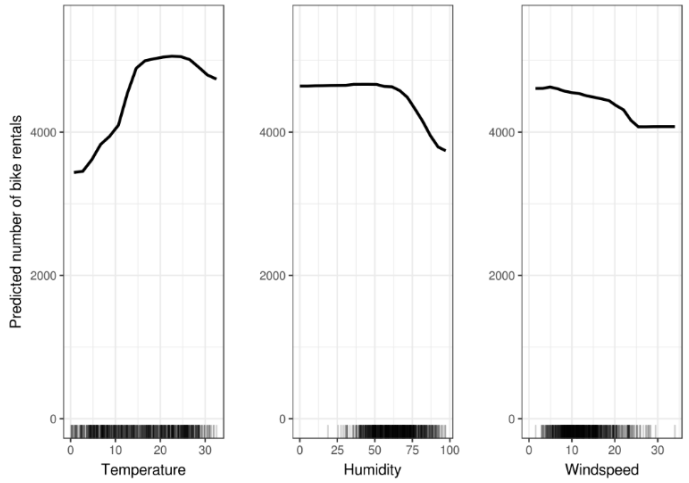
Individual Conditional Expectation (ICE)

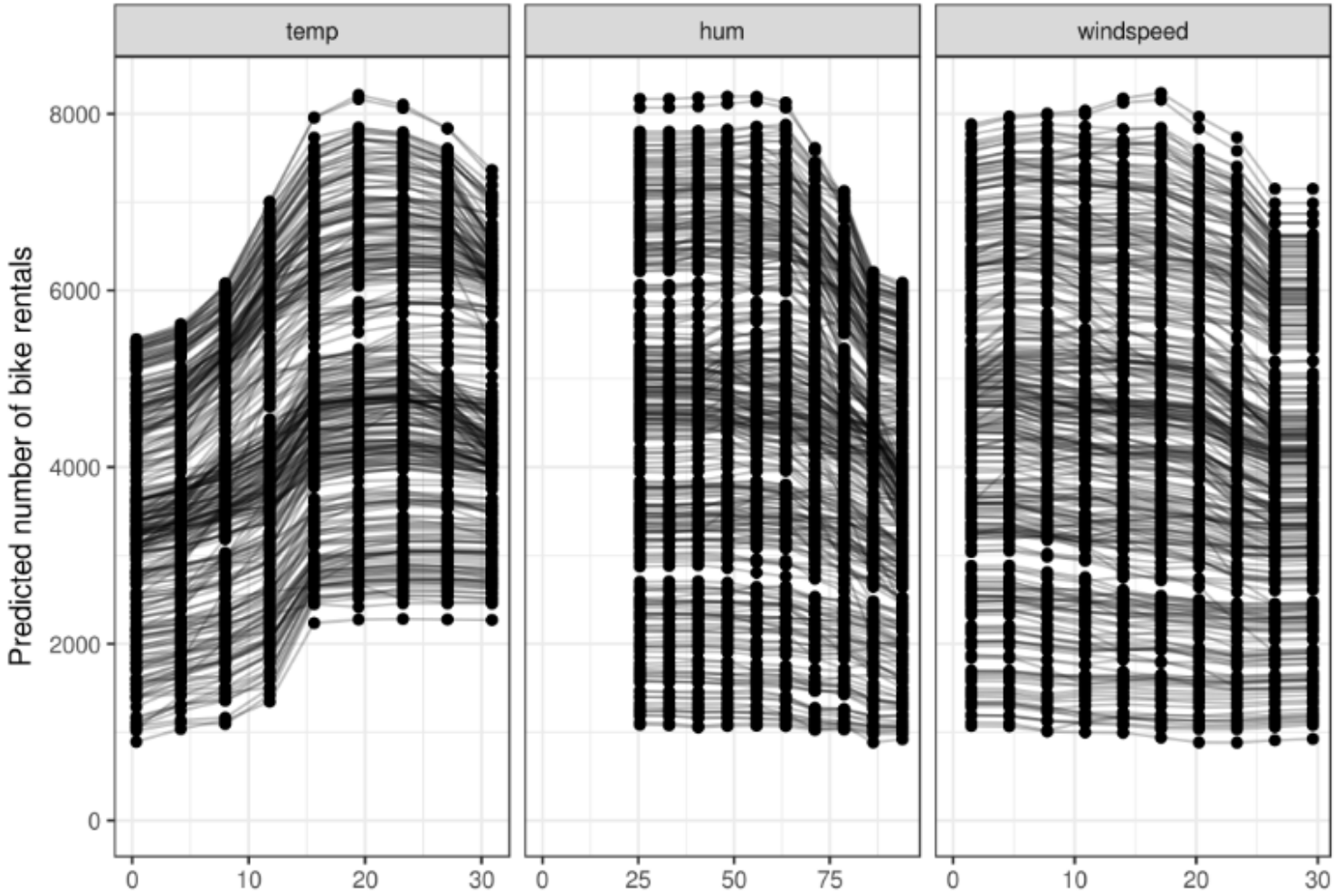
- For a chosen feature, Individual Conditional Expectation (ICE) plots draw one line per instance, representing how the instance's prediction changes when the feature changes.



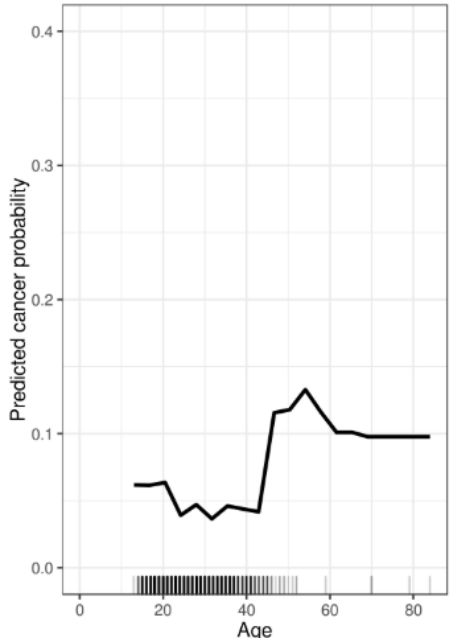


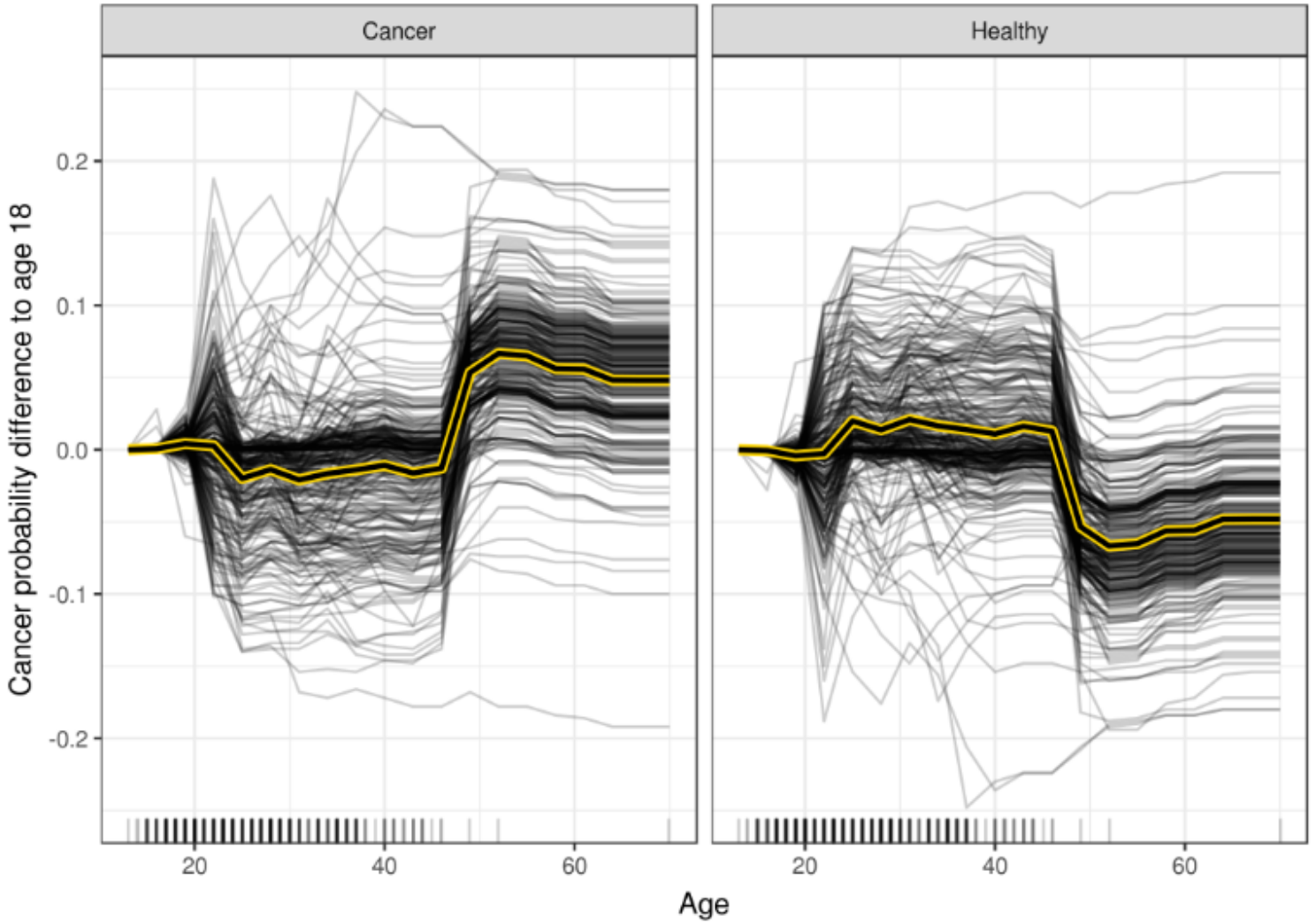
ICE – Bike Rental



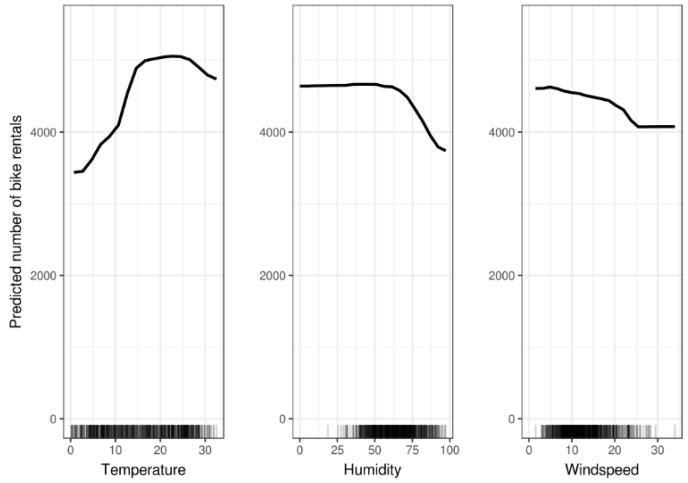


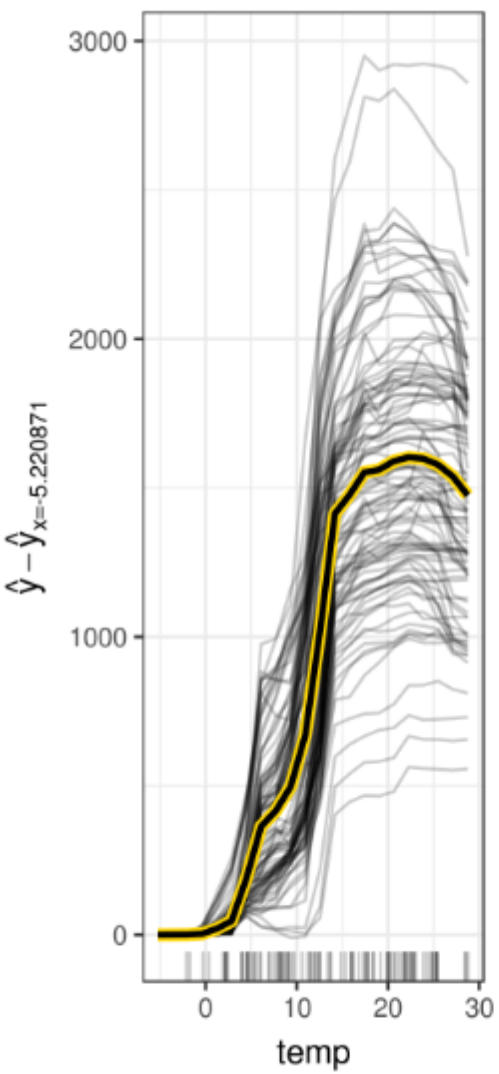
Centered ICE Plot (c-ICE)

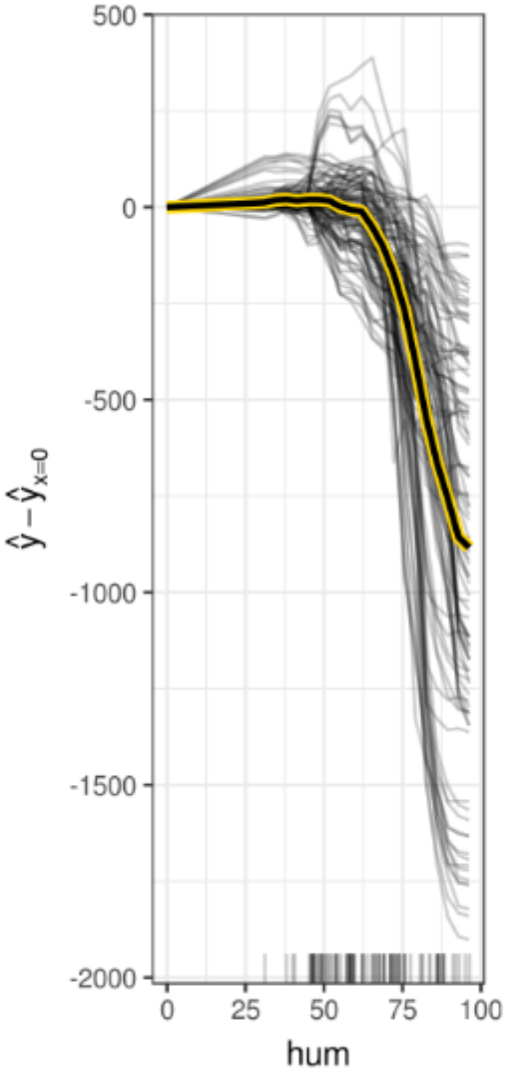


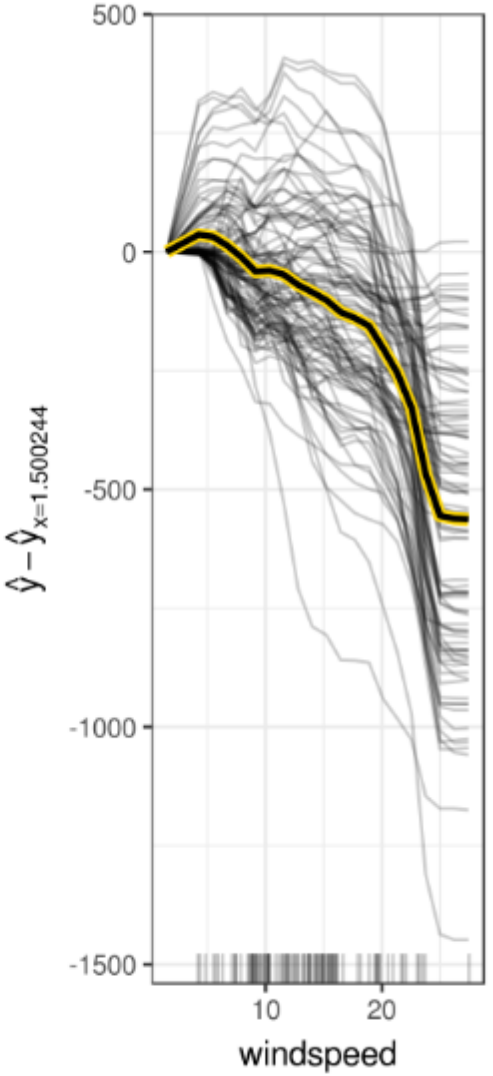


c-ICE – Bike Rental









ICE Pro's and Con's

Pro's

- Individual conditional expectation curves are **even more intuitive to understand** than partial dependence plots.
- In contrast to partial dependence plots they can **uncover heterogeneous relationships**.

Con's

- ICE curves **can only display one feature** meaningfully, because two features would require drawing multiple, overlaying surfaces and there is no way you would still see anything in the plot.
- ICE curves suffer from the same problem as PDPs: When the feature of interest is correlated with the other features, then **not all points in the lines might be valid data points** according to the joint feature distribution.
- When many ICE curves are drawn the plot **can become overcrowded** and you don't see anything any more.
- In ICE plots it might not be easy to **see the average**.

Feature Interaction

- When features in a prediction model interact with each other, then the influence of the features on the prediction is not additive but more complex
- When a machine learning model makes a prediction based on two features, we can decompose the prediction into four terms:
  - a constant term,
  - one term for the first feature
  - one for the second feature and
  - one for the interaction effect between the two features.

- Friedman's H-statistic
  - Between two features

$$H_{jk}^2 = \sum_{i=1}^{n} \left[PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)}) \right] / \sum_{i=1}^{n} PD_{jk}^2(x_j^{(i)}, x_k^{(i)})$$

  - Between one and rest

$$H_j^2 = \sum_{i=1}^{n} \left[\hat{f}(x^{(i)}) - PD_j(x_j^{(i)}) - PD_{-j}(x_{-j}^{(i)}) \right] / \sum_{i=1}^{n} \hat{f}^2(x^{(i)})$$

Feature Interaction

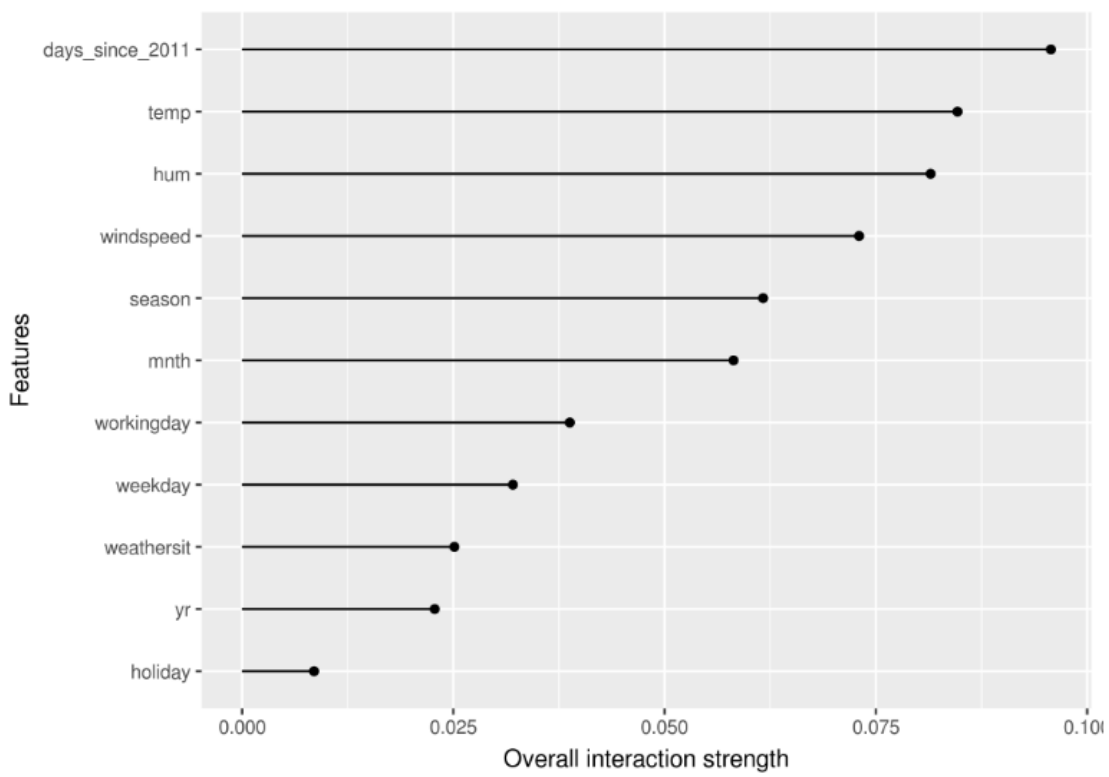
- Constant – 150k
- First Feature (Location) – 50k if good
- Second Feature (Size) – 100k if big
- Interaction – None

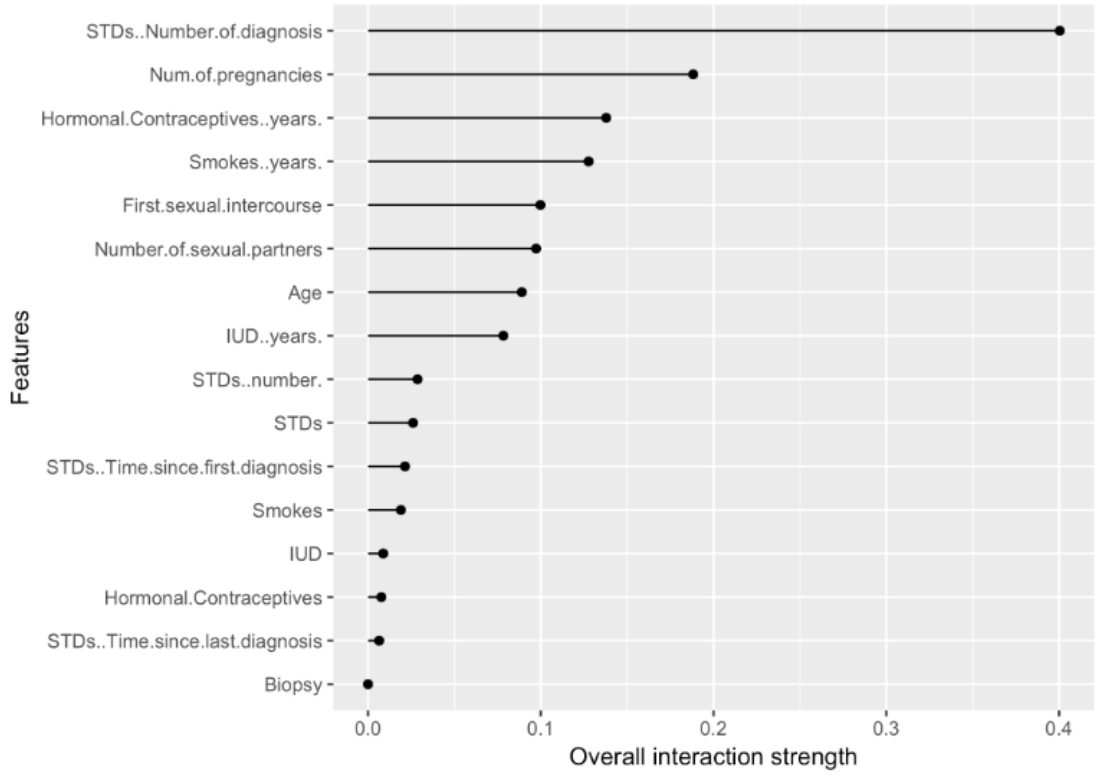
Location	Size	Predicted value
good	big	300,000
good	small	200,000
bad	big	250,000
bad	small	150,000

- Constant – 150k
- First Feature (Location) – 50k if good
- Second Feature (Size) – 100k if big
- Interaction (Location & Size) – 100k if Good and Big

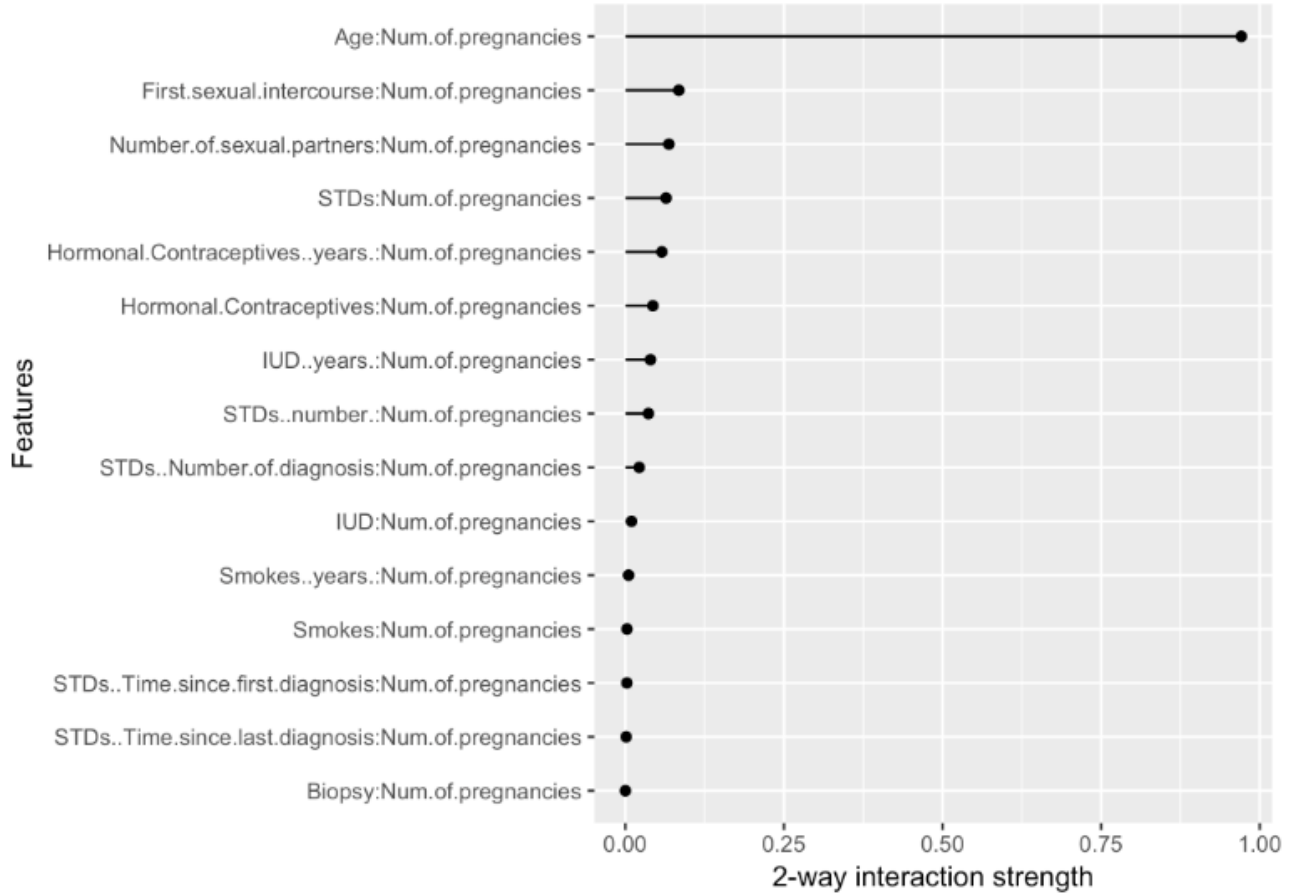
Location	Size	Predicted value
good	big	400,000
good	small	200,000
bad	big	250,000
bad	small	150,000

Feature Interaction – Examples (One with Other)





Two Way Feature Interaction



Feature Interactions – Pro's

The interaction statistic has an **underlying theory** through the partial dependence decomposition.

The H-statistic has a **meaningful interpretation**: The interaction is defined as the portion of variance explained by the interaction.

Since the statistic is **dimensionless and always between 0 and 1**, it is comparable across features and even across models.

The statistic **detects all kinds of interactions**, regardless of a specific form.

With the H-statistic it is also possible to analyze arbitrary **higher interactions**: For example the interaction strength between 3 or more features.

Feature Interactions – Con's

The interaction H-statistic takes a long time to compute, because it's **computationally expensive**.

The computation involves estimating marginal distributions. These **estimates also have some variance**, when we don't use all of the data points. This means when we sample points, the estimates will also vary from run to run and the results might **become unstable**.

The H-statistic tells us how strong the interactions are, but it doesn't tell us how the interaction is shaped.

The H-statistic can't be meaningfully applied if the inputs are pixels

The interaction statistic works under the assumption that we can independently shuffle features

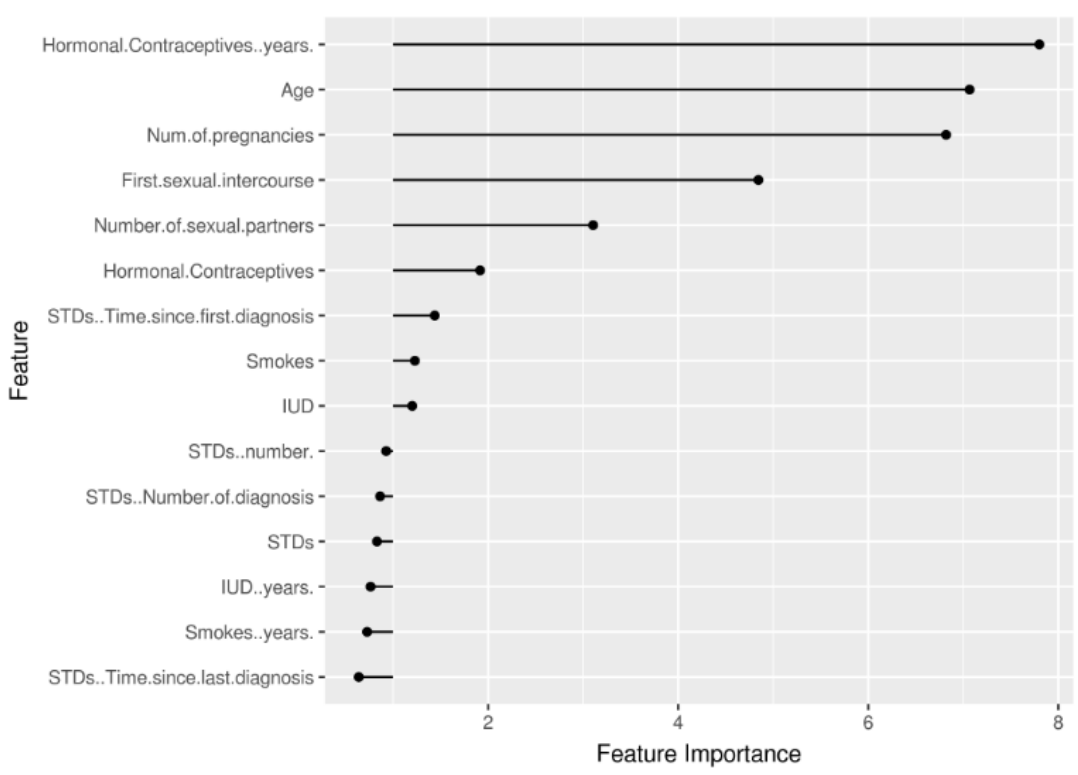
Feature Importance

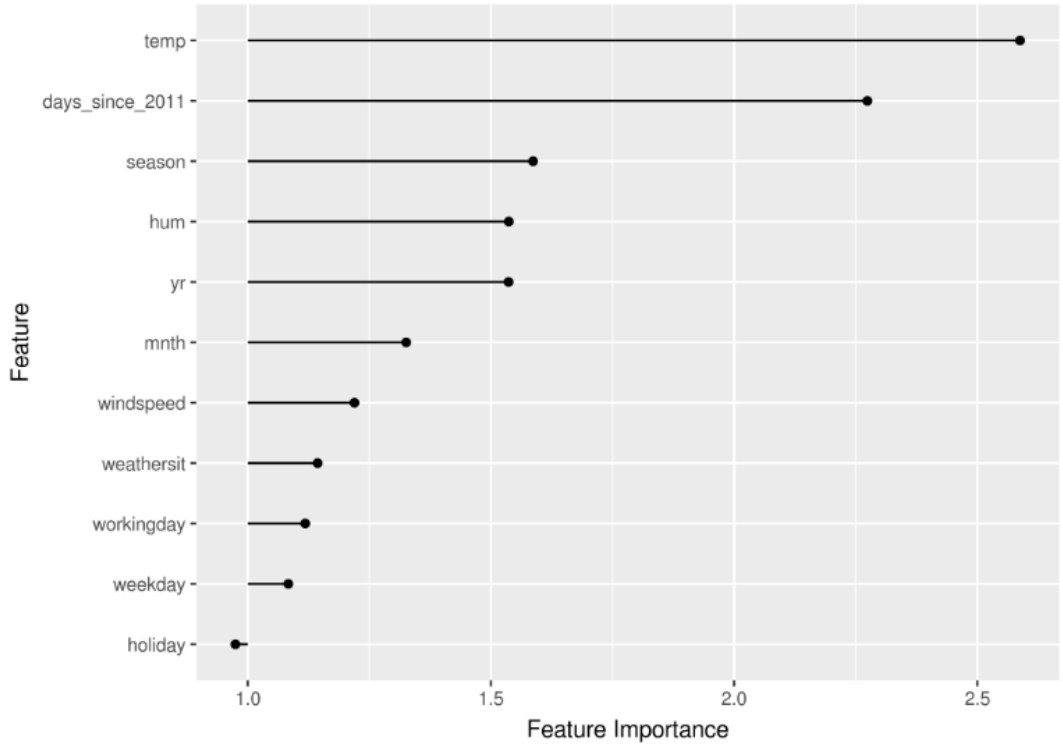
Input: Trained model $\hat{f}$, feature matrix X, target vector Y, error measure $L(Y, \hat{Y})$

1. Estimate the original model error $e_{orig}(\hat{f}) = L(Y, \hat{f}(X))$ (e.g. mean squared error)
2. For each feature $j \in 1, \ldots, p$ do
   - Generate feature matrix X_{perm_j} by permuting feature X_j in X. This breaks the association between X_j and Y.
   - Estimate error $e_{perm} = L(Y, \hat{f}(X_{perm_j}))$ based on the predictions of the permuted data.
   - Calculate permutation feature importance $FI_j = e_{perm}(\hat{f}) / e_{orig}(\hat{f})$. Alternatively, the difference can be used: $FI_j = e_{perm}(\hat{f}) - e_{orig}(\hat{f})$
3. Sort variables by descending FI.

- We measure a feature's importance by calculating the increase of the model's prediction error after permuting the feature. A feature is "important" if permuting its values increases the model error, because the model relied on the feature for the prediction.

Feature Importance Example





Feature Importance Pro's Vs Con's

Pro's	Con's
• Nice interpretation: Feature importance is the increase of model error when the feature's information is destroyed. • Feature importance provides a highly compressed, global insight into the model's behavior. • A positive aspect of using the error ratio instead of the error difference is that the feature importance measurements are comparable across different problems.	• You need access to the actual outcome target. • When features are correlated, the permutation feature importance measure can be biased by unrealistic data instances. • Adding a correlated feature can decrease the importance of the associated feature, by splitting up the importance on both features.

Global Surrogate Models

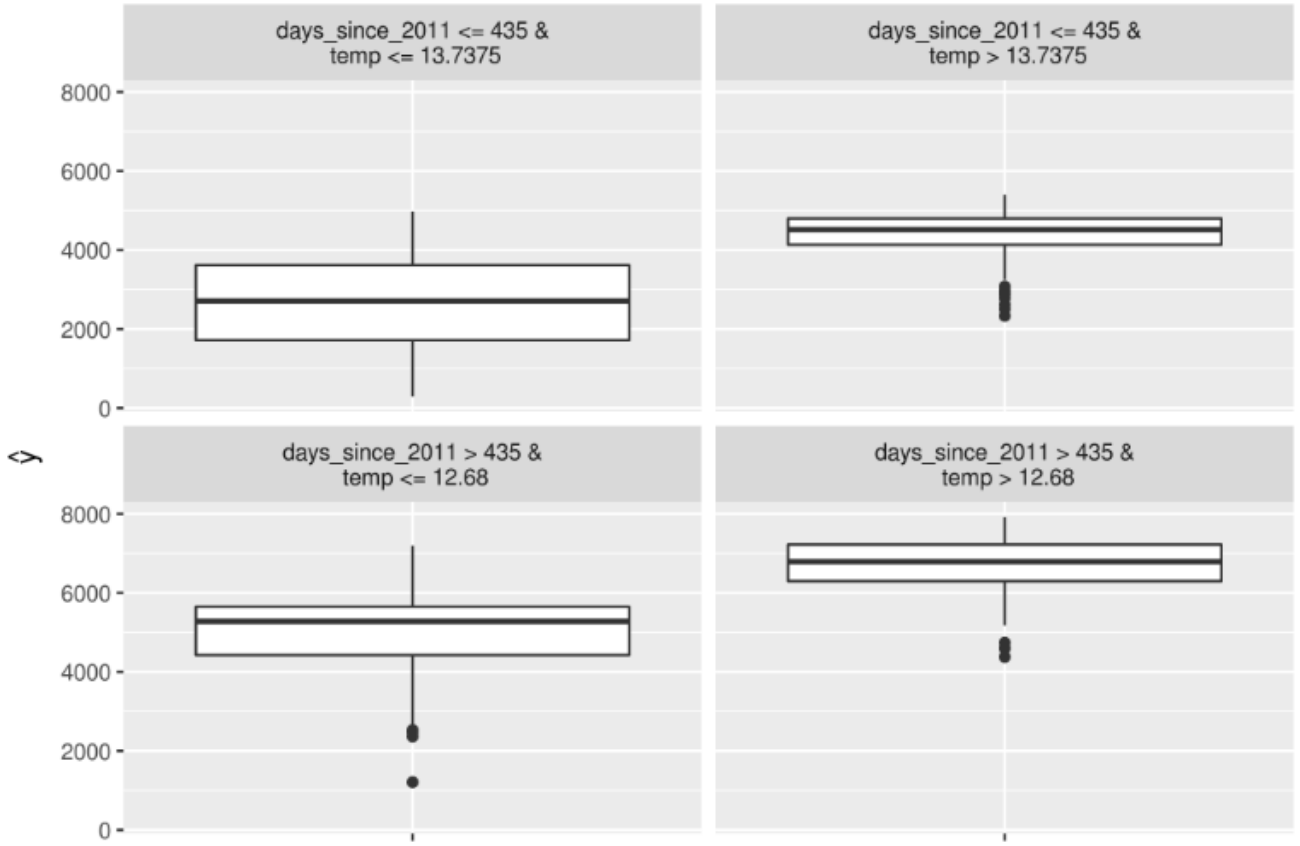
- A global surrogate model is an interpretable model that is trained to approximate the predictions of a black box model. We can draw conclusions about the black box model by interpreting the surrogate model.

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i^* - \hat{y}_i)^2}{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}$$

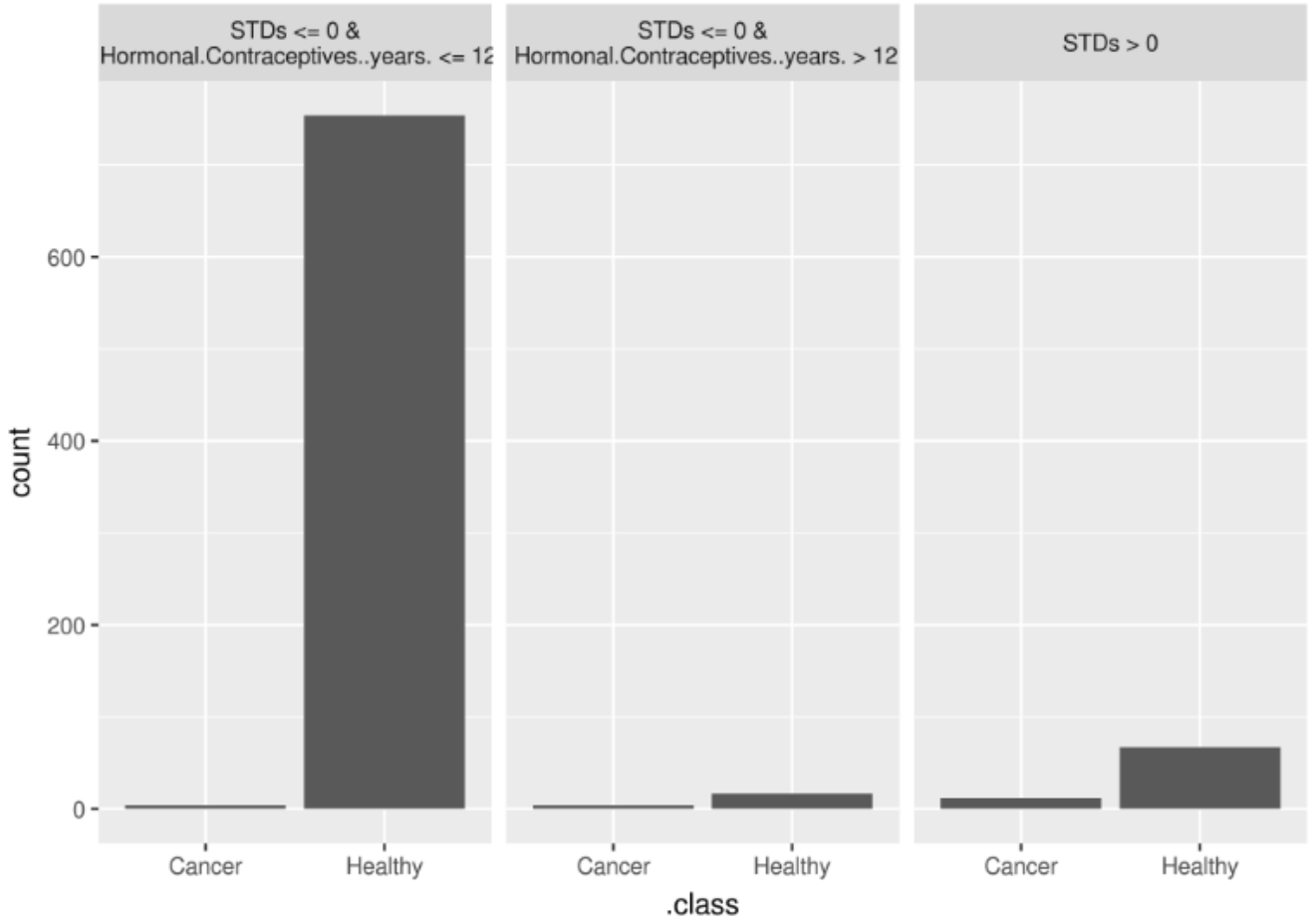
1. Choose a dataset X. This could be the same dataset that was used for training the black box model or a new dataset from the same distribution. You could even choose a subset of the data or a grid of points, depending on your application.
2. For the chosen dataset X, get the predictions $\hat{y}$ of the black box model.
3. Choose an interpretable model (linear model, decision tree, ...).
4. Train the interpretable model on the dataset X and its predictions $\hat{y}$.
5. Congratulations! You now have a surrogate model.
6. Measure how well the surrogate model replicates the prediction of the black box model.
7. Interpret / visualize the surrogate model.

Global Surrogate - Example

- Based on a random forest based model predictions, we train a surrogate model using decision tree



Global Surrogate - Example



Global Surrogate Models – Pro's vs Con's

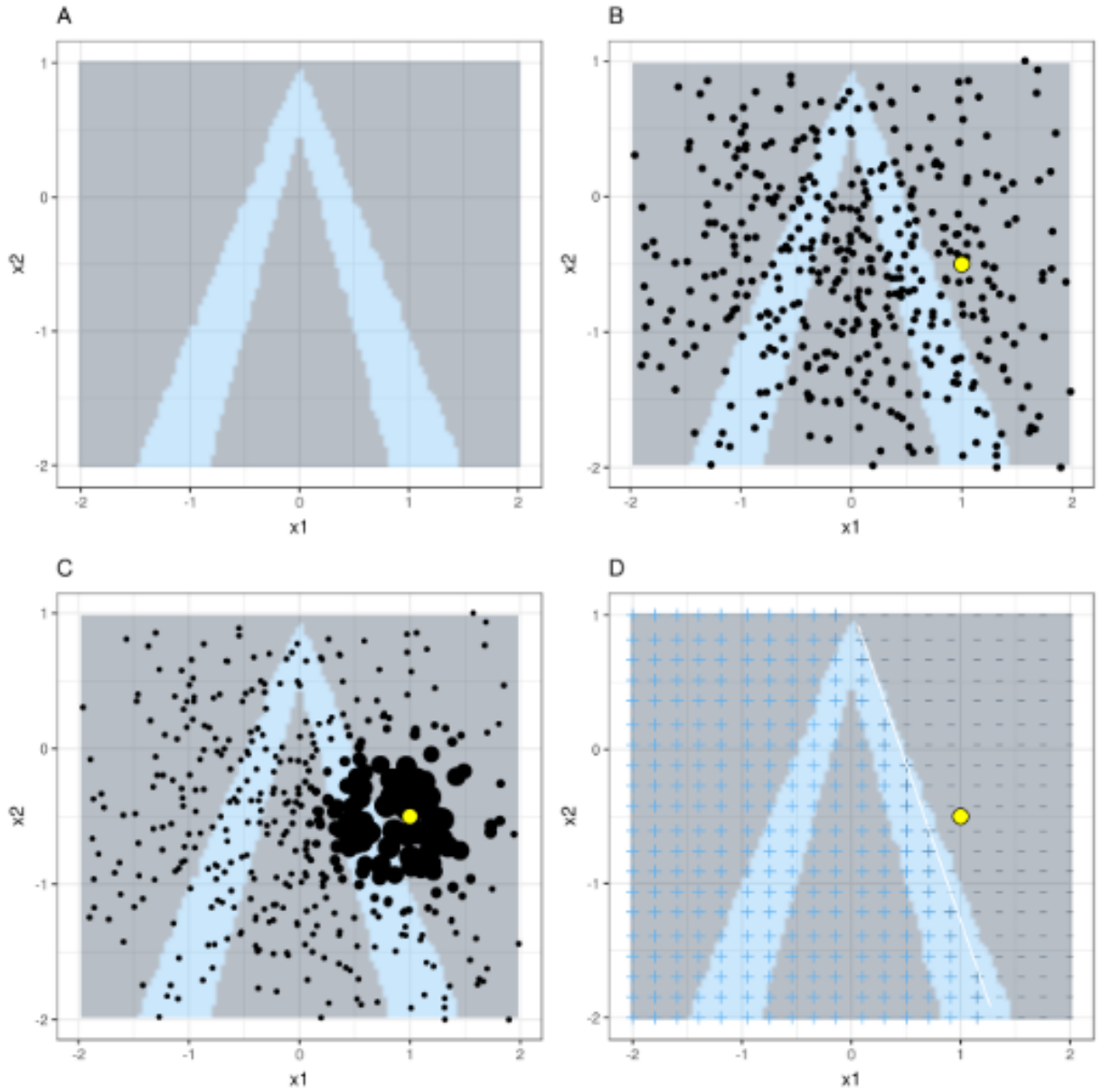
Pro's	Con's
• The surrogate model method is **flexible**: Any interpretable models can be used. This also means that you can swap not only the interpretable model, but also the underlying black box model. • The approach is very **intuitive** and straightforward. • With the R squard measure, we can easily **measure** how good our surrogate models are in terms of approximation of the black box predictions.	• Be careful to draw **conclusions about the model, not the data**, since the surrogate model never sees the real outcome. • It's not clear what the best **cut-off for R squared** is in order to be confident that the surrogate model is close enough to the black box model. 80% of variance explained? 50%? 99%? • The interpretable model you choose as a surrogate **comes with all its advantages and disadvantages**.

Local Surrogate Model Explainability (LIME)

- Local interpretable model-agnostic explanations (LIME) is a method for fitting local, interpretable models that can explain single predictions of any black-box machine learning model. LIME explanations are local surrogate models.
  - Choose your instance of interest for which you want to have an explanation of its black box prediction.
  - Perturb your dataset and get the black box predictions for these new points.
  - Weight the new samples by their proximity to the instance of interest.
  - Fit a weighted, interpretable model on the dataset with the variations.
  - Explain prediction by interpreting the local model.

https://arxiv.org/pdf/1602.04938.pdf

LIME for Tabular Data



LIME for Text Data

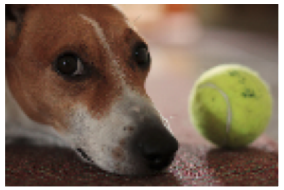
	CONTENT	CLASS
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

	For	Christmas	Song	visit	my	channel!	;)	prob	weight
2	1	0	1	1	0	0	1	0.09	0.57
3	0	1	1	1	1	0	1	0.09	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	0.09	0.57

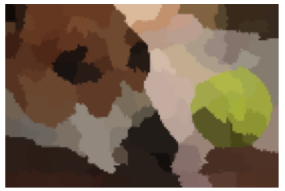
case	label_prob	feature	feature_weight
1	0.0872151	good	0.000000
1	0.0872151	a	0.000000
1	0.0872151	PSY	0.000000
2	0.9939759	channel!	6.908755
2	0.9939759	visit	0.000000
2	0.9939759	Christmas	0.000000

The word “channel” points to a high probability of spam.

LIME for Image Data

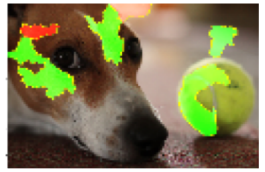


Segment the image



Creates several variants of the original image, each with a different combination of “fudged” super pixels and original pixels

- It computes its distance from the original image
- It uses the classifier to classify this variant.
- Create an optimal linear model of the “importance” of each image segment to the eventual class
- We can ask for the “top segment” or “top X segments” which explains the classification of this image as a tennis ball (corresponding to the highest coefficient in our linear model)



Shapley Value

- Predictions can be explained by assuming that each feature is a ‘player’ in a game where the prediction is the payout. The Shapley value - a method from coalitional game theory - tells us how to fairly distribute the ‘payout’ among the features.

Thanks You & Questions