42578 Advanced Business Analytics

Interpretable Machine Learning

Stanislav Borysov [stabo@dtu.dk]

Lecture's scope

- **The lecture is not about**
  - Exploratory data analysis
  - Unsupervised learning
  - Feature engineering
  - Performance evaluation
  - Residual analysis

- **The lecture is about**
  - Supervised learning
  - Explaining predictive models

Why Do We Need Interpretable ML?

Why do we need interpretable ML?

1 Fairness

- Suppose we are trying to predict employees' performance in a big company to expedite the appraisal process and identify the best employees. We have data from the last 10 years about the performance reviews of employees. **But what if that company tends to promote more men than women?**

- In this case, the model might learn the **bias** and predict that men tend to perform at a higher scale (and this bias has unfortunately happened in certain real-world scenarios). Now, if there is no way to interpret our model at this stage, the model might end up providing false insights at the cost of compromising on fairness.

https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/

Why do we need interpretable ML?

2 Checking causality and debugging models

- Can you build your trust based on accuracy?

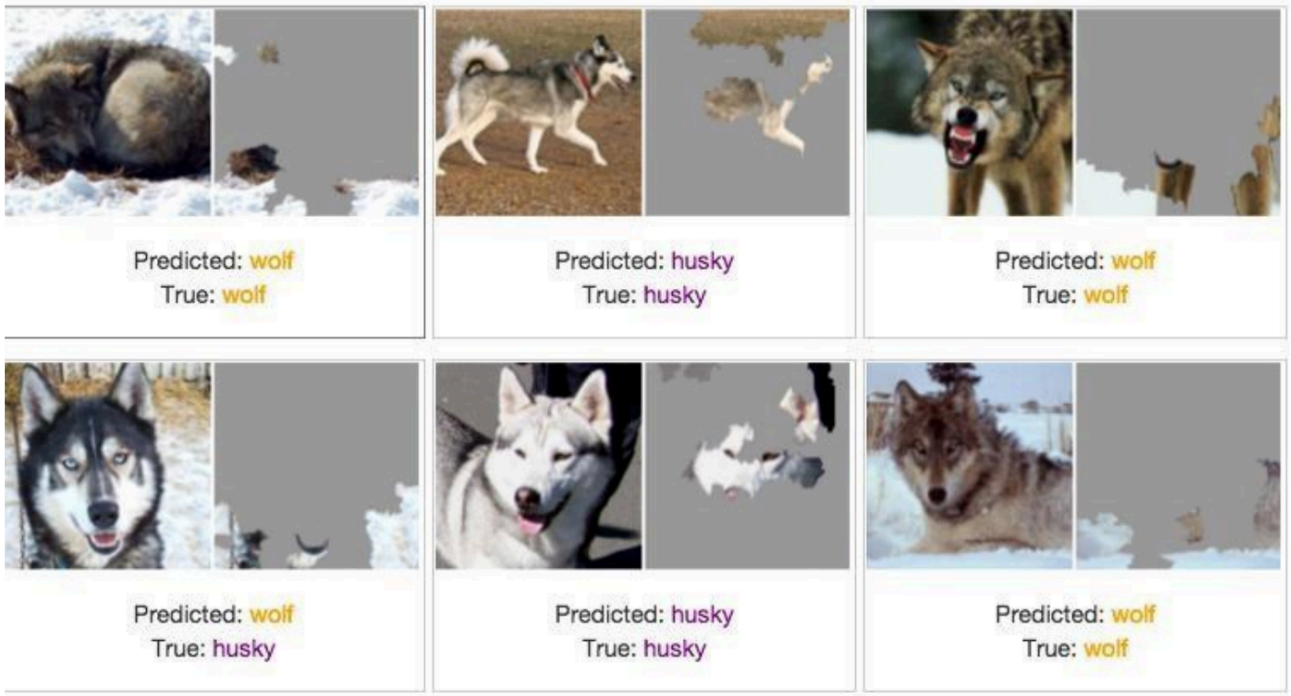
Only 1 mistake!

https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/

Why do we need interpretable ML?

2 Checking causality and debugging models

- … Yes, if you want to build a great snow detector!

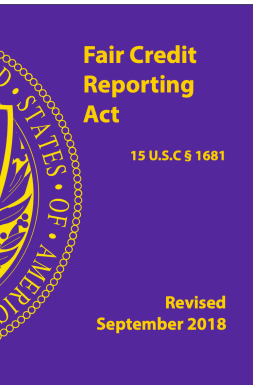


https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/

Why do we need interpretable ML?

3 Regulations

- Based on the latest regulations by the EU, GDPR's article 12 allows individuals to enquire about algorithmic decisions. For example, in the banking and finance industry, questions such as the following can come up and have to be answered by these banks:
  - *Why was my loan rejected?*
  - *Why did I get a low Credit Limit on my credit card? etc.*



- It's required in regulated industries like finance and healthcare to audit the decision process and ensure it's not discriminatory. The US Fair Credit Reporting Act requires that agencies disclose "all of the key factors that adversely affected the credit score of the consumer in the model used, the total number of which shall not exceed four".

https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/

Why do we need interpretable ML?

- The existing machine learning algorithms have overwhelmed the world with their predictive power. But, they often aren't capable of providing explanations for their predictions.

- **ML interpretability is the degree to which a human can understand the reason and rationale behind a model's output.**

- It aims to answer the following questions:
  - *Why a particular prediction was made, as opposed to others?*
  - *When does the model succeed? When to trust a model?*
  - *When does it fail & why? When to disregard the model output?*

https://medium.com/walmartlabs/interpretable-machine-learning-part-i-bd1829b42b3a

Why do we need interpretable ML?

- *What drives model predictions?* We should have the ability to query our model and find out latent feature interactions to get an idea of which features might be important in the decision-making policies of the model. This ensures **fairness** of the model.
- *Why did the model take a certain decision?* We should also be able to validate and justify why certain key features were responsible for driving certain decisions taken by a model during predictions. This ensures **accountability** and **reliability** of the model.
- *How can we trust model predictions?* We should be able to evaluate and validate any data point and how a model takes decisions on it. This should be demonstrable and easy to understand for key stakeholders that the model works as expected. In fields like medical, defence, judiciary, education etc., models must be strictly answerable for their predictions. This ensures **transparency** of the model.
- Interpretable models can help to bridge the gap between data scientists and domain experts and enable effective exchange of **data insights and knowledge**.

https://towardsdatascience.com/human-interpretable-machine-learning-part-1-the-need-and-importance-of-model-interpretation-2ed758f5f476
https://towardsdatascience.com/explainable-artificial-intelligence-part-2-model-interpretation-strategies-75d4afa6b739
https://towardsdatascience.com/explainable-artificial-intelligence-part-3-hands-on-machine-learning-model-interpretation-e8ebe5afc608

Overview of Interpretability in ML

Interpretability dimensions

1 **Model**

- Model-specific vs Model-agnostic

2 **Scope**

- Global vs Local

3 **Data**

- Most important data samples (support vectors in SVM, most crucial training samples, etc.)
- Data acquisition strategies (e.g., active learning)
- Data neighbourhood in K-Nearest Neighbours
- Adversarial examples

Interpretability dimensions: Today's focus

1 **Model**

- Model-specific vs Model-agnostic

2 **Scope**

- Global vs Local

3 **Data**

- Most important data samples (support vectors in SVM, most crucial training samples, etc.)
- Data acquisition strategies (e.g., active learning)
- Data neighbourhood in K-Nearest Neighbours
- Adversarial examples

1 Model: Intrinsic or post hoc?

- **Intrinsic ("white box") interpretability** is all about leveraging a machine learning model which is intrinsically interpretable in nature (like linear models, parametric models or tree-based models).

- **Post hoc interpretability** means selecting and training a black box model (ensemble methods or neural networks) and applying interpretability methods after the training (feature importance, partial dependency plots).

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

1 Model: Model-specific or model-agnostic?

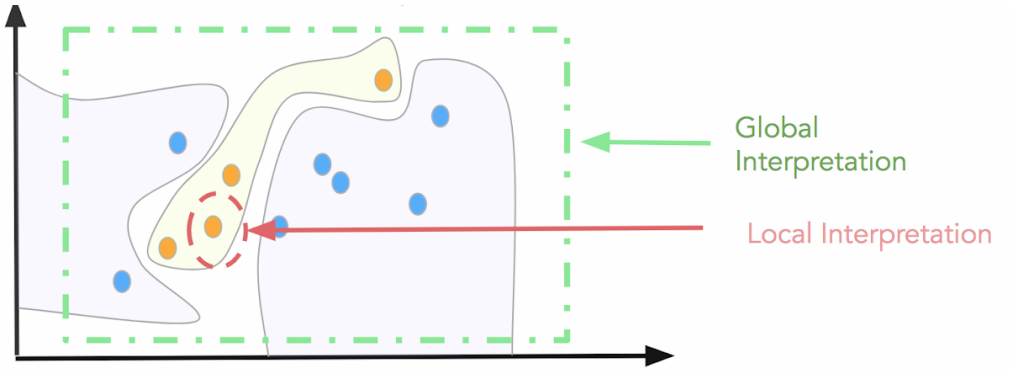
- **Model-specific interpretation tools** are very specific to intrinsic model interpretation methods which depend purely on the capabilities and features on a per-model basis. This can be coefficients, p-values, AIC scores pertaining to a regression model, rules from a decision tree and so on.

- **Model-agnostic interpretation tools** are more relevant to post hoc methods and can be used on any machine learning model. These agnostic methods usually operate by analysing (and perturbations of inputs) feature input and output pairs. By definition, these methods do not have access to any model internals like weights, constraints or assumptions.

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

2 Scope: Local or global?

- *If the interpretation method explains a single prediction or the entire model behaviour? Or if the scope is somewhere in between?*



- **Global interpretability** could illuminate the inner working mechanisms of machine learning models and thus can increase their transparency.
- **Local interpretability** will help uncover the causal relations between a specific input and its corresponding model prediction.

- Those two help users to **trust a model** and **trust a prediction**, respectively.

- **Aggregate local importances for the whole dataset to get the global view**

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"
https://cacm.acm.org/magazines/2020/1/241703-techniques-for-interpretable-machine-learning/fulltext
https://towardsdatascience.com/explainable-artificial-intelligence-part-2-model-interpretation-strategies-75d4afa6b739

2 Scope: Local or global?

- You can zoom in on a single instance and examine what the model predicts for this input, and explain why. If you look at an individual prediction, the behaviour of the otherwise complex model might behave more pleasantly. Locally, the prediction might only depend linearly or monotonically on some features, rather than having a complex dependence on them. For example, the value of a house may depend nonlinearly on its size. But if you are looking at only one particular 100 square meters house, there is a possibility that for that data subset, your model prediction depends linearly on the size. You can find this out by simulating how the predicted price changes when you increase or decrease the size by 10 square meters. Local explanations can, therefore, be more accurate than global explanations. This book presents methods that can make individual predictions more interpretable in the section on model-agnostic methods.

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Examples of the approaches

	Model-specific	Model-agnostic

Local	• **White-box models**
	  • Generalised Linear Models (e.g., linear / logistic / Poisson regression)
	  • Decision Rules, Decision Trees
	  • Naive Bayes
	• **Ensembles of white-box models**
	  • Random Forest, Boosted Trees
	• **Kernel methods**
	  • K-Nearest Neighbours
	  • Support Vector Machines
	• **Neural Networks**
	  • Neuron “firing” patterns
	  • Convolutional feature maps
		• **Local white-box surrogate models**
		  • LIME
		• **Game theory**
		  • Shapley values & SHAP

Global	• **White-box models**
	• **Ensembles of white-box models**
	• **Kernel methods**
	• **Neural Networks**
	  • Weight analysis
	  • Convolutional filter weights
		• **White-box global surrogate models**
		• **Partial dependence plots**
		• **Permutation feature importance**

The rest of the lecture…

	Model-specific	Model-agnostic

Local	• **White-box models**
	  • Generalised Linear Models (e.g., linear / logistic / Poisson regression)
	  • Decision Rules, Decision Trees
	  • Naive Bayes
	• **Ensembles of white-box models**
	  • Random Forest, Boosted Trees
	• **Kernel methods**
	  • K-Nearest Neighbours
	  • Support Vector Machines
	• **Neural Networks**
	  • Neuron “firing” patterns
	  • Convolutional feature maps
		• **Local white-box surrogate models**
		  • LIME
		• **Game theory**
		  • Shapley values & SHAP
		3

Global	• **White-box models**
	• **Ensembles of white-box models**
	• **Kernel methods**
	• **Neural Networks**
	  • Weight analysis
	  • Convolutional filter weights
	1
		• **White-box global surrogate models**
		• **Partial dependence plots**
		• **Permutation feature importance**
		2

Model-Specific Interpretability

Linear Regression

Linear Regression

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon$$

- **Interpretation of numerical features**
  An increase of the feature x_k by one unit increases the prediction for y by β_k units when all other feature values remain fixed.

- **Interpretation of categorical features**
  Changing the feature x_k from the reference category to the other category increases the prediction for y by β_k when all other features remain fixed.

- **Interpretation of the intercept β_0**
  The intercept is the feature weight for the "constant feature", which is always 1 for all instances. For the instance with all numerical feature values at zero and the categorical feature values at the reference categories, the model prediction is the intercept weight. The interpretation of the intercept is usually not relevant because instances with all features values at zero often make no sense. The interpretation is only meaningful when the features have been standardised to have zero mean. Then the intercept reflects the predicted outcome of an instance where all features are at their mean value.

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

Linear Regression: Feature importance

- **Global: Coefficients (weights)** β_k
  - Normalisation of features to compare the relative importance

  OR
  - Calculate t-statistic $t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$

- **Local: Effect of feature *k* for sample *i***
  - $\text{effect}_k^{(i)} = \beta_k x_k^{(i)}$

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

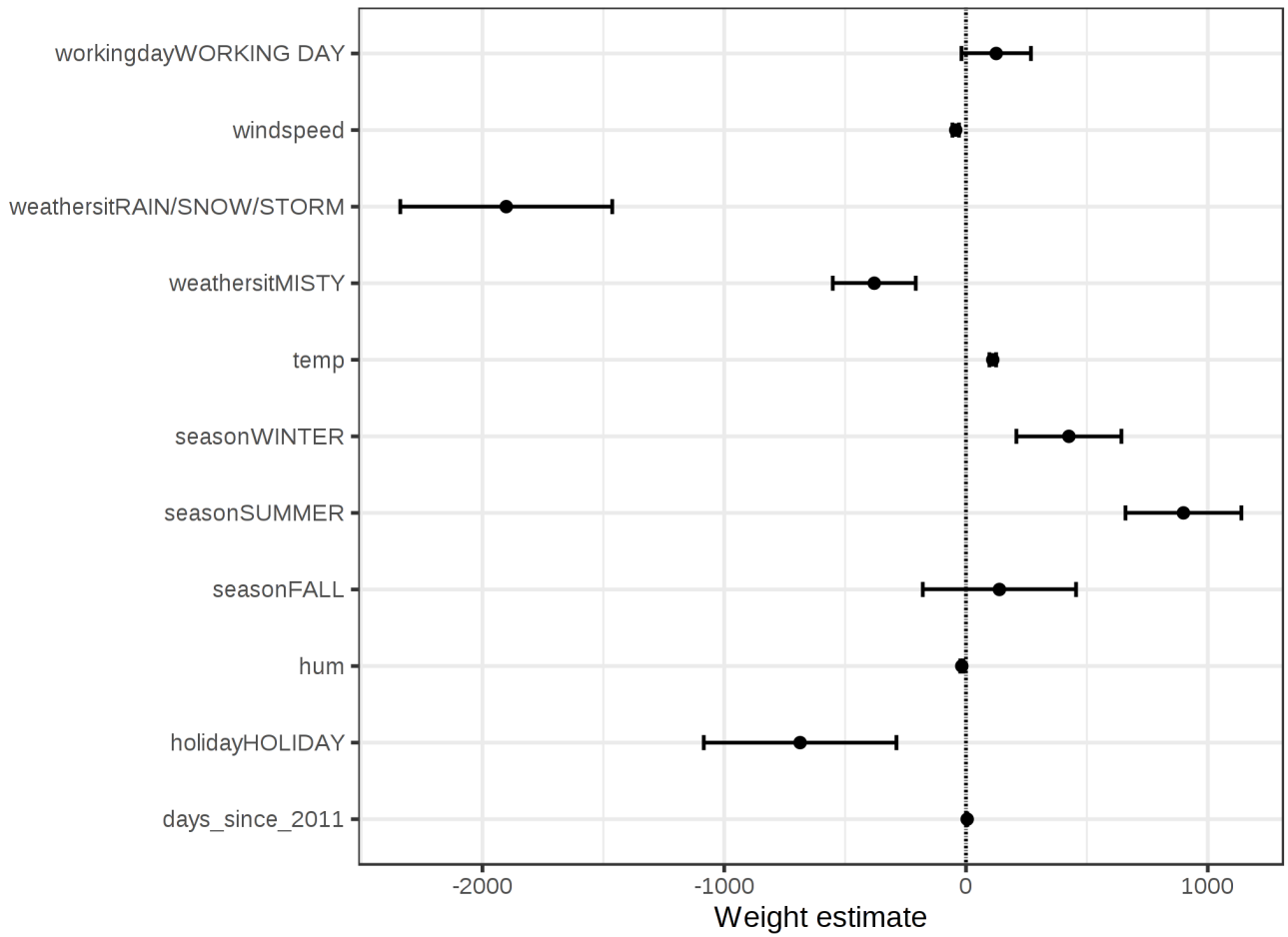
Linear Regression: Example

- Use the linear regression model to predict the number of rented bikes on a particular day, given weather and calendar information.

	Weight	SE	|t|
(Intercept)	2399.4	238.3	10.1
seasonSUMMER	899.3	122.3	7.4
seasonFALL	138.2	161.7	0.9
seasonWINTER	425.6	110.8	3.8
holidayHOLIDAY	-686.1	203.3	3.4
workingdayWORKING DAY	124.9	73.3	1.7
weathersitMISTY	-379.4	87.6	4.3
weathersitRAIN/SNOW/STORM	-1901.5	223.6	8.5
temp	110.7	7.0	15.7
hum	-17.4	3.2	5.5
windspeed	-42.5	6.9	6.2
days_since_2011	4.9	0.2	28.5

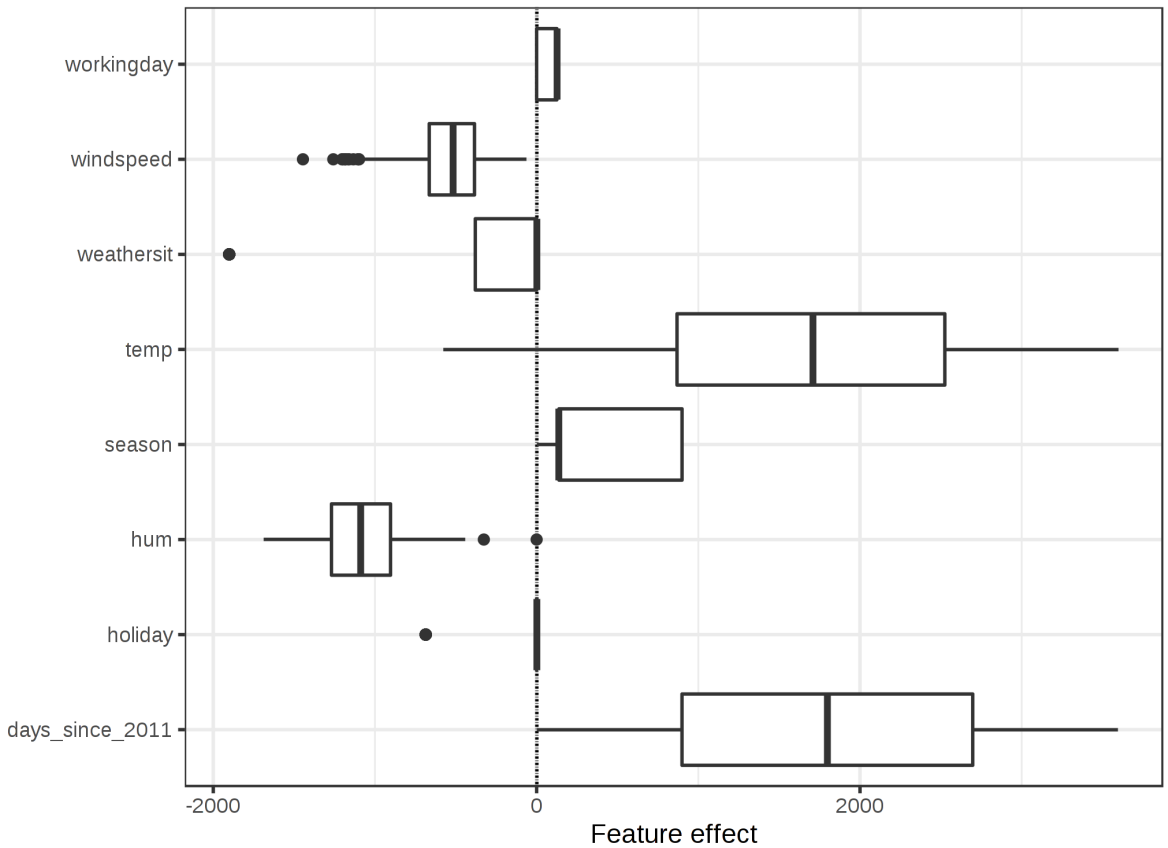
C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Global importance: Weight plot



C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

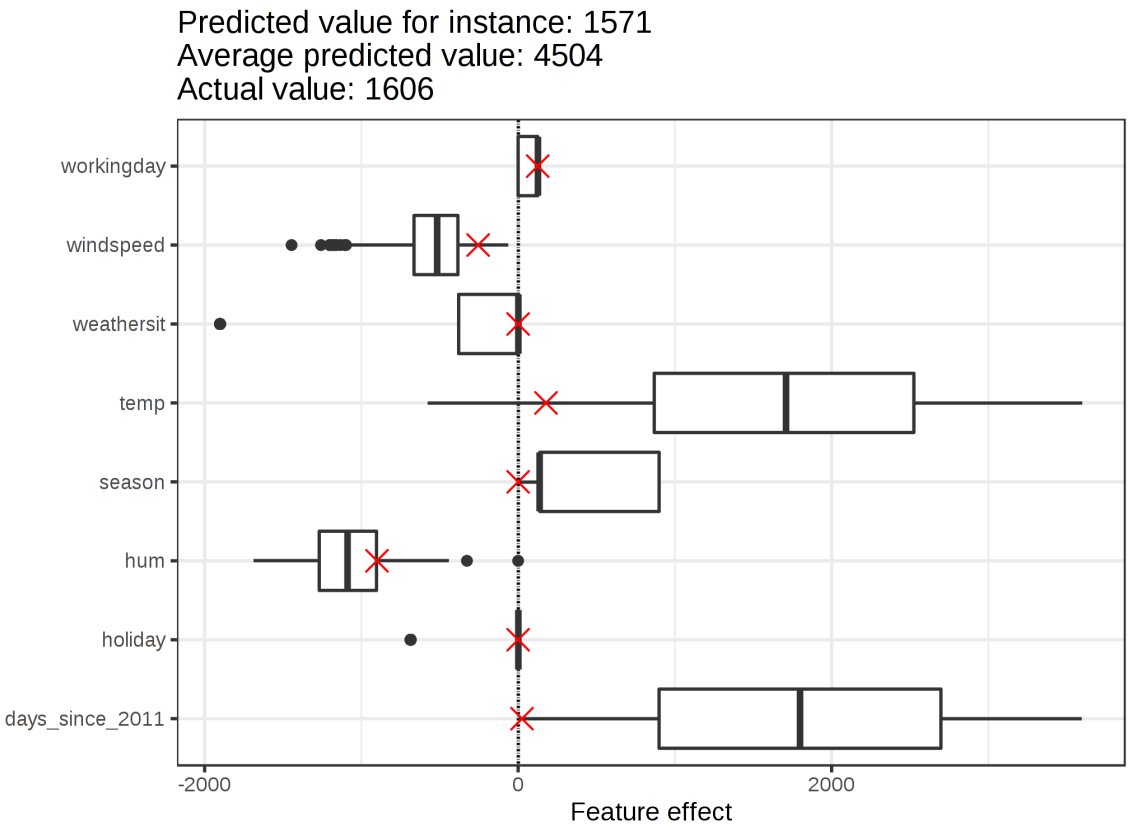
Effect plot (all samples)



C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Local importance: Effect plot (single sample)

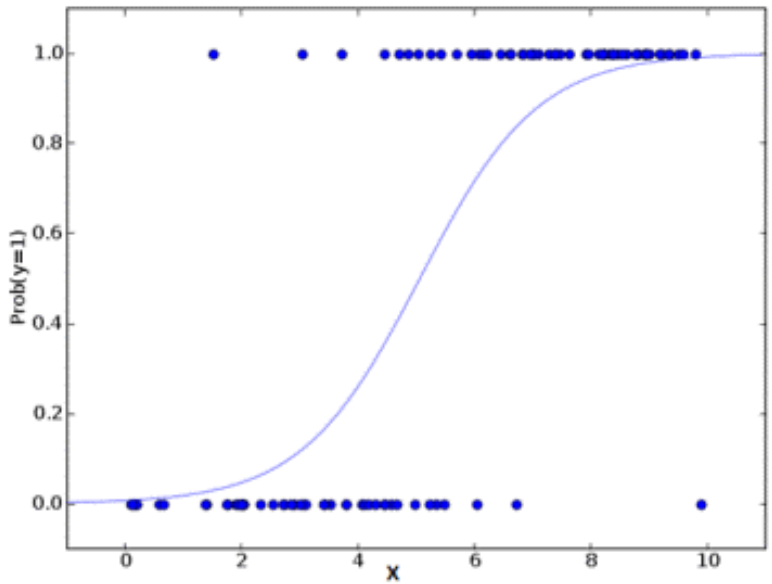
Feature	Value
season	SPRING
yr	2011
mnth	JAN
holiday	NO HOLIDAY
weekday	THU
workingday	WORKING DAY
weathersit	GOOD
temp	1.604356
hum	51.8261
windspeed	6.000868
cnt	1606
days_since_2011	5



C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Logistic Regression

Logistic Regression



$$P(y^{(i)} = 1) = \frac{1}{1 + exp(-(\beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_p x_p^{(i)}))}$$

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Logistic Regression: Interpretation of coefficients

- Let's rewrite the probability expression

$$log\left(\frac{P(y=1)}{1-P(y=1)}\right) = log\left(\frac{P(y=1)}{P(y=0)}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

$$\frac{P(y=1)}{1-P(y=1)} = odds = exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p\right)$$

- What happens when we increase one of the feature values by 1?

$$\frac{odds_{x_j+1}}{odds} = \frac{exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_j(x_j+1) + \ldots + \beta_p x_p\right)}{exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_j x_j + \ldots + \beta_p x_p\right)}$$

$$\frac{odds_{x_j+1}}{odds} = exp\left(\beta_j(x_j+1) - \beta_j x_j\right) = exp\left(\beta_j\right)$$

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

Logistic Regression: Interpretation of coefficients

$$\frac{odds_{x_j+1}}{odds} = exp\left(\beta_j(x_j+1) - \beta_j x_j\right) = exp\left(\beta_j\right)$$

- A change in a feature by one unit changes (multiplicatively) the odds ratio by a factor of $\exp(\beta_j)$
- For example, if you have odds of 2, it means that the probability for y=1 is twice as high as y=0. If you have a weight (= log odds ratio) of 0.7, then increasing the respective feature by one unit multiplies the odds by exp(0.7) (approximately 2) and the odds change to 4.

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Logistic Regression: Example

- Logistic regression to predict cervical cancer based on some risk factors.

	Weight	Odds ratio	Std. Error
Intercept	-2.91	0.05	0.32
Hormonal contraceptives y/n	-0.12	0.89	0.30
Smokes y/n	0.26	1.29	0.37
Num. of pregnancies	0.04	1.04	0.10
Num. of diagnosed STDs	0.82	2.26	0.33
Intrauterine device y/n	0.62	1.85	0.40

- Interpretation of a numerical feature (“Num. of diagnosed STDs”): An increase in the number of diagnosed STDs (sexually transmitted diseases) changes (increases) the odds of cancer vs. no cancer by a factor of 2.26, when all other features remain the same. Keep in mind that correlation does not imply causation.
- Interpretation of a categorical feature (“Hormonal contraceptives y/n”): For women using hormonal contraceptives, the odds for cancer vs. no cancer are by a factor of 0.89 lower, compared to women without hormonal contraceptives, given all other features stay the same.
- Like in the linear model, the interpretations always come with the clause that ‘all other features stay the same’.

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Inherently interpretable models: Generalised Linear Models (GLM)

$$\underbrace{\mathrm{E}(\mathbf{Y}) = \boldsymbol{\mu}}_{1} = \underbrace{g^{-1}}_{2}(\underbrace{\mathbf{X}}_{3}\underbrace{\boldsymbol{\beta}}_{4})$$

1 Target variable

2 Link function

3 Features (independent variables)

4 Parameters

Distribution	Support of distribution	Typical uses	Link name	Link function
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\boldsymbol{\beta} = \mu$
Exponential	real: $(0, +\infty)$	Exponential-response data, scale parameters	Inverse	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-1}$
Gamma				
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-2}$
Poisson	integer: $[0, +\infty)$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\boldsymbol{\beta} = \ln(\mu)$
Bernoulli	integer: $[0, 1]$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$
Binomial	integer: $[0, N]$	count of # of "yes" occurrences out of N yes/no occurrences		
Categorical	integer: $[0, K)$	outcome of single K-way occurrence		
	K-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1			
Multinomial	K-vector of integer: $[0, N]$	count of occurrences of different types (1 .. K) out of N total K-way occurrences		

Linear regression

Logistic regression

https://en.wikipedia.org/wiki/Generalized_linear_model

GLM interpretability

- **Global**
  - β describe both the magnitude and direction of each feature
- **Local**
  - Multiply the coefficient vector by a specific feature vector to see the predicted value, and the contribution of each feature to that prediction
- **Bonuses**
  - Solid statistical foundations
    - p-values
    - Confidence intervals
    - ANOVA
  - Regularisation works as a feature selector
  - …

GLM interpretability note

- The interpretation of a single weight always comes with the footnote that the other input features remain at the same value, which is not the case with many real applications.
- For example, a linear model that predicts the value of a house, that takes into account both the size of the house and the number of rooms, can have a negative weight for the room feature. It can happen because there is already the highly correlated house size feature. In a market where people prefer larger rooms, a house with fewer rooms could be worth more than a house with more rooms if both have the same size. The weights only make sense in the context of the other features in the model.

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Inherently interpretable models: Naïve Bayes

Inherently interpretable models: Naïve Bayes

$$p(C_k \mid x_1, \ldots, x_n) = \frac{1}{Z} \underbrace{p(C_k)}_{\text{priors}} \prod_{i=1}^{n} \underbrace{p(x_i \mid C_k)}_{\text{likelihoods}} \qquad \underbrace{Z = p(\mathbf{x})}_{\text{evidence}}$$

Inherently interpretable models: Naïve Bayes

$$p(C_k \mid x_1, \ldots, x_n) = \frac{1}{Z} \underbrace{p(C_k)}_{\text{priors}} \prod_{i=1}^{n} \underbrace{p(x_i \mid C_k)}_{\text{likelihoods}} \qquad \underbrace{Z = p(\mathbf{x})}_{\text{evidence}}$$

Classifier

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

Inherently interpretable models: Naïve Bayes

$$p(C_k \mid x_1, \ldots, x_n) = \frac{1}{Z} \underbrace{p(C_k)}_{\text{priors}} \prod_{i=1}^{n} \underbrace{p(x_i \mid C_k)}_{\text{likelihoods}} \qquad \underbrace{Z = p(\mathbf{x})}_{\text{evidence}}$$

Classifier

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

Continuous

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

Categorical

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

Binary

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

Inherently interpretable models: Naïve Bayes

$$p(C_k \mid x_1, \ldots, x_n) = \frac{1}{Z} \underbrace{p(C_k)}_{\text{priors}} \prod_{i=1}^{n} \underbrace{p(x_i \mid C_k)}_{\text{likelihoods}} \qquad \underbrace{Z = p(\mathbf{x})}_{\text{evidence}}$$

Classifier

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

Global & Local interpretability

Continuous

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

Categorical

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

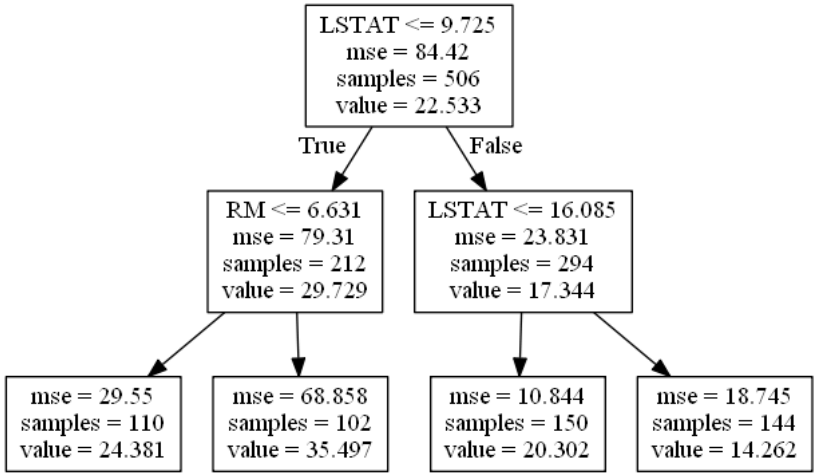
Binary

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

Inherently interpretable models: Decision Tree

Inherently interpretable models: Decision Tree

Regression



- If the target is a continuous value, then for node m, representing a region R_m with N_m observations $X_m = \{X^{(i)}\}_{i \in R_m}$, common criteria to minimise as for determining locations for future splits are

- **Mean Squared Error:**

$$H(X_m) = \frac{1}{N_m} \sum_{i \in R_m} (y_i - \bar{y}_m)^2$$

- **Mean Absolute Error:**

$$H(X_m) = \frac{1}{N_m} \sum_{i \in R_m} |y_i - \bar{y}_m|$$

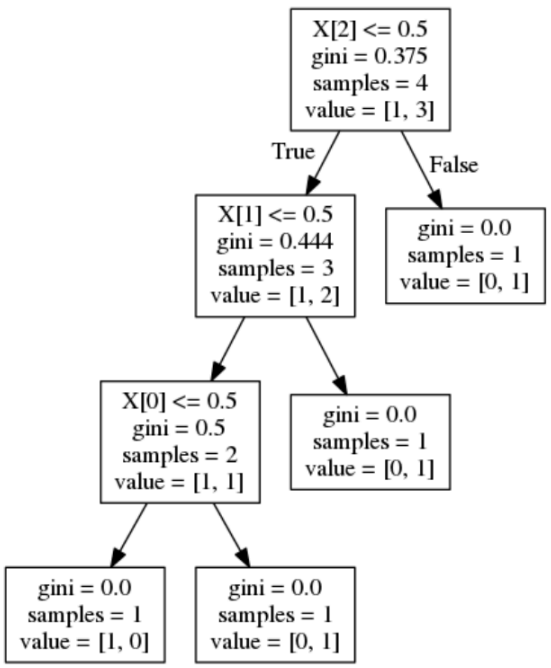
- where

$$\bar{y}_m = \frac{1}{N_m} \sum_{i \in R_m} y_i$$

https://scikit-learn.org/stable/modules/tree.html

Inherently interpretable models: Decision Tree

Classification



- If the target is a classification outcome taking on values 0,1,…,K-1, for node m, representing a region representing a region R_m with N_m observations $X_m = \{X^{(i)}\}_{i \in N_m}$, let

$$p_{mk} = 1/N_m \sum_{x_i \in R_m} I(y_i = k)$$

be the proportion of class k observations in node m. Common measures of impurity are

- **Gini**

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk})$$

- **Entropy**

$$H(X_m) = -\sum_k p_{mk} \log(p_{mk})$$

https://scikit-learn.org/stable/modules/tree.html

Decision Tree: Interpretation

- **DT interpretation:**
  Starting from the root node, you go to the next nodes and the edges tell you which subsets you are looking at. Once you reach the leaf node, the node tells you the predicted outcome. All the edges are connected by ‘AND’.

  Template: *If feature x is [smaller/bigger] than threshold c AND … then the predicted outcome is the mean value of y of the instances in that node.*

https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/

Inherently interpretable models: Decision Tree

- Tree based models split the data multiple times according to certain cutoff values in the features. Through splitting, different subsets of the dataset are created, with each instance belonging to one subset. The final subsets are called terminal or leaf nodes and the intermediate subsets are called internal nodes or split nodes. To predict the outcome in each leaf node, the average outcome of the training data in this node is used. Trees can be used for classification and regression.
- There are various algorithms that can grow a tree. They differ in the possible structure of the tree (e.g. number of splits per node), the criteria how to find the splits, when to stop splitting and how to estimate the simple models within the leaf nodes. The classification and regression trees (CART) algorithm is probably the most popular algorithm for tree induction.

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Inherently interpretable models: Decision Tree

- CART takes a feature and determines which cut-off point minimizes the variance of y for a regression task or the Gini index of the class distribution of y for classification tasks. The variance tells us how much the y values in a node are spread around their mean value. The Gini index tells us how “impure” a node is, e.g. if all classes have the same frequency, the node is impure, if only one class is present, it is maximally pure. Variance and Gini index are minimized when the data points in the nodes have very similar values for y. As a consequence, the best cut-off point makes the two resulting subsets as different as possible with respect to the target outcome. For categorical features, the algorithm tries to create subsets by trying different groupings of categories. After the best cutoff per feature has been determined, the algorithm selects the feature for splitting that would result in the best partition in terms of the variance or Gini index and adds this split to the tree. The algorithm continues this search-and-split recursively in both new nodes until a stop criterion is reached. Possible criteria are: A minimum number of instances that have to be in a node before the split, or the minimum number of instances that have to be in a terminal node.

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Decision Tree: Global importance (“Impurity reduction”, “Gain”)

- Decision trees make splits to maximise the decrease in impurity. We can use this reduction to measure the contribution of each feature.

- **Step 1:** Go through all the splits in which the feature was used
- **Step 2:** Measure the reduction in criterion (Gini/information gain/Variance Reduced) compared to the parent node weighted by the number of samples. These weights are important as they help us incorporate the number of samples being segregated by the given feature as well into the equation. The formula is:

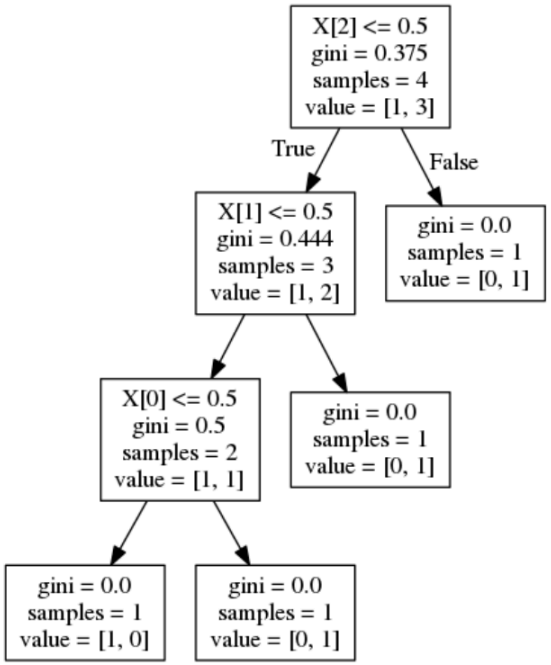
$$\frac{N_{parent}}{N}\left(Gini_{parent} - \frac{N_{Right}}{N_{parent}} \cdot Gini_{Right} - \frac{N_{Left}}{N_{parent}} \cdot Gini_{Left}\right)$$

  - Here,
    - N is the total number of observations
    - N & Gini represents the number of samples & Gini impurity in the parent, left and right node
- **Step 3:** Take the sum for all splits for each feature and compare. (The sum of all importances is scaled to 100. This means that each importance can be interpreted as a share of the overall model importance.)

https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/

Global importance: Example

- Let's say we have a decision tree with 4 samples and Gini impurity values as shown in the below figure. Each feature here appears only once so we can directly use the formula to calculate feature importance:



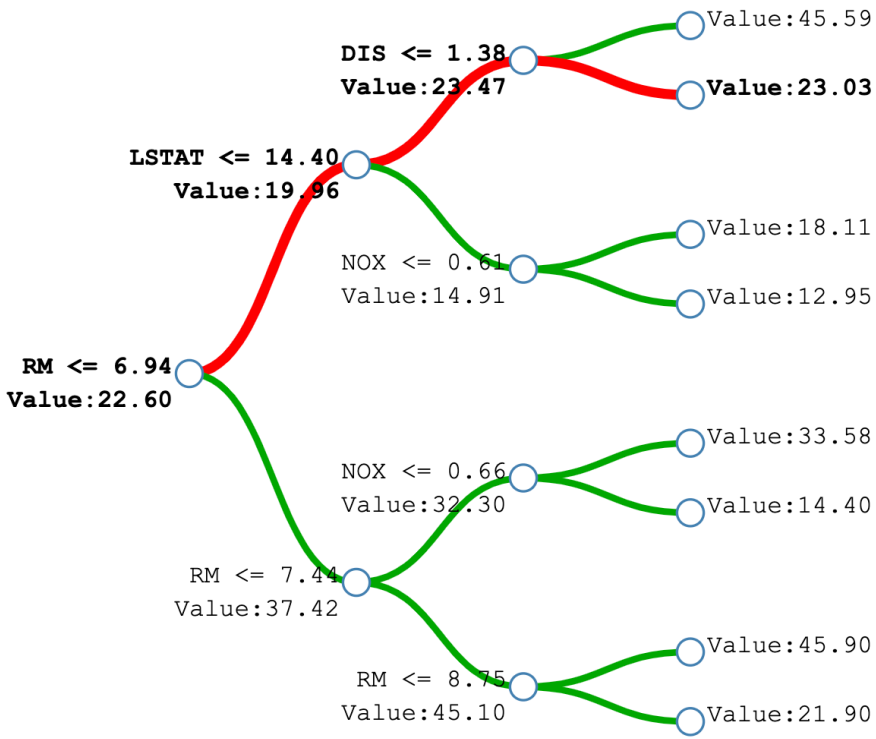
- Since each feature is used once in our case, there is no need to calculate the sum.
- For X[2] :
  = (4 / 4) * (0.375 – (0.75 * 0.444)) = 0.042
- For X[1] :
  = (3 / 4) * (0.444 – (2/3 * 0.5)) = 0.083
- For X[0] :
  = (2 / 4) * (0.5) = 0.25

`sklearn.tree.DecisionTreeRegressor.feature_importances_`

`sklearn.tree.DecisionTreeClassifier.feature_importances_`

https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/

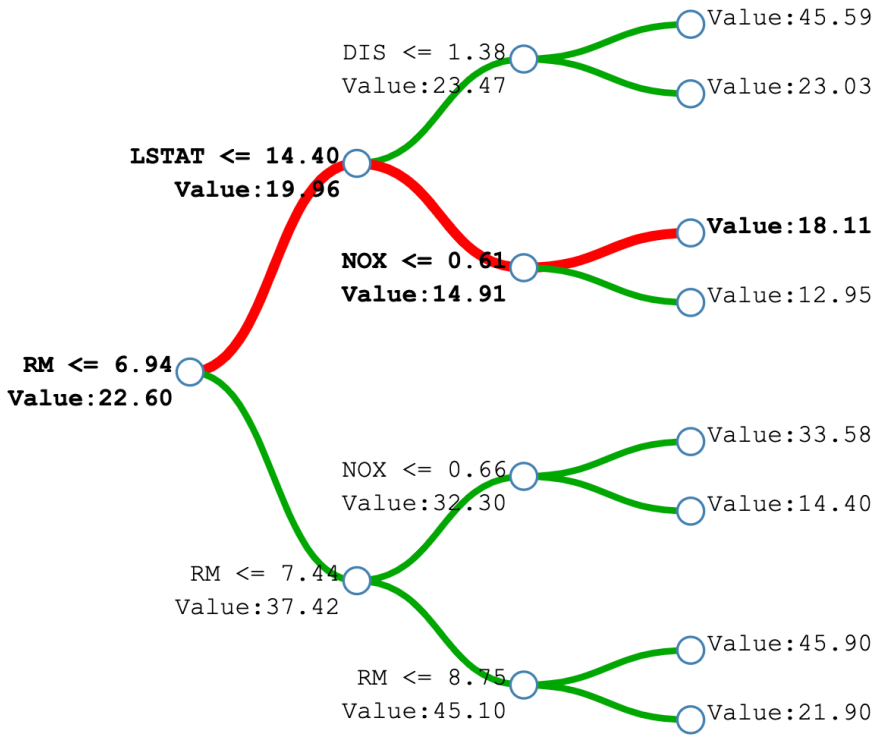
Decision Tree: Local importance



Prediction: 23.03 ≈ 22.60 (trainset mean) - 2.64(loss from RM) + 3.52(gain from LSTAT) - 0.44(loss from DIS)

http://blog.datadive.net/interpreting-random-forests/

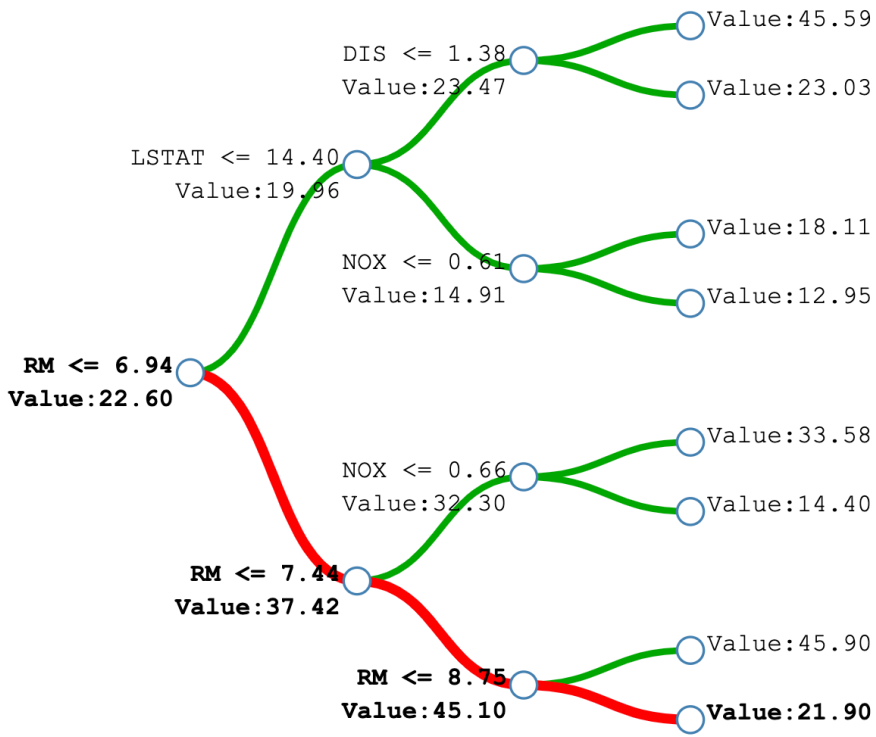
Decision Tree: Local importance



Prediction: 18.11 ≈ 22.60 (trainset mean) - 2.64(loss from RM) - 5.04(loss from LSTAT) + 3.20(gain from NOX)

http://blog.datadive.net/interpreting-random-forests/

Decision Tree: Local importance



Prediction: 21.90 ≈ 22.60 (trainset mean) + 14.82(gain from RM) + 7.68(gain from RM) - 23.2(loss from RM)

http://blog.datadive.net/interpreting-random-forests/

Decision Tree: Local importance

- Individual predictions of a decision tree can be explained by decomposing the decision path into one component per feature. We can track a decision through the tree and explain a prediction by the contributions added at each decision node.
- The root node in a decision tree is our starting point. If we were to use the root node to make predictions, it would predict the mean of the outcome of the training data. With the next split, we either subtract or add a term to this sum, depending on the next node in the path. To get to the final prediction, we have to follow the path of the data instance that we want to explain and keep adding to the formula.

$$\hat{f}(x) = \bar{y} + \sum_{d=1}^{D} \text{split.contrib(d,x)} = \bar{y} + \sum_{j=1}^{p} \text{feat.contrib(j,x)}$$

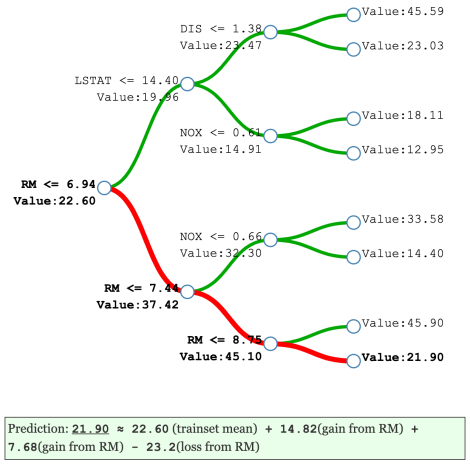
- The prediction of an individual instance is the mean of the target outcome plus the sum of all contributions of the D splits that occur between the root node and the terminal node where the instance ends up. We are not interested in the split contributions though, but in the feature contributions. A feature might be used for more than one split or not at all. We can add the contributions for each of the p features and get an interpretation of how much each feature has contributed to a prediction.

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Decision Tree: Local importance

- Practically, features contribution computation has been made possible thanks to a new capability of scikit-learn 0.17: allowing to store values for all nodes instead of just leaves for decision trees.

- This approach has been described here http://blog.datadive.net/random-forest-interpretation-with-scikit-learn/ and is available in the Treeinterpreter package https://github.com/andosa/treeinterpreter
  `pip install treeinterpreter`



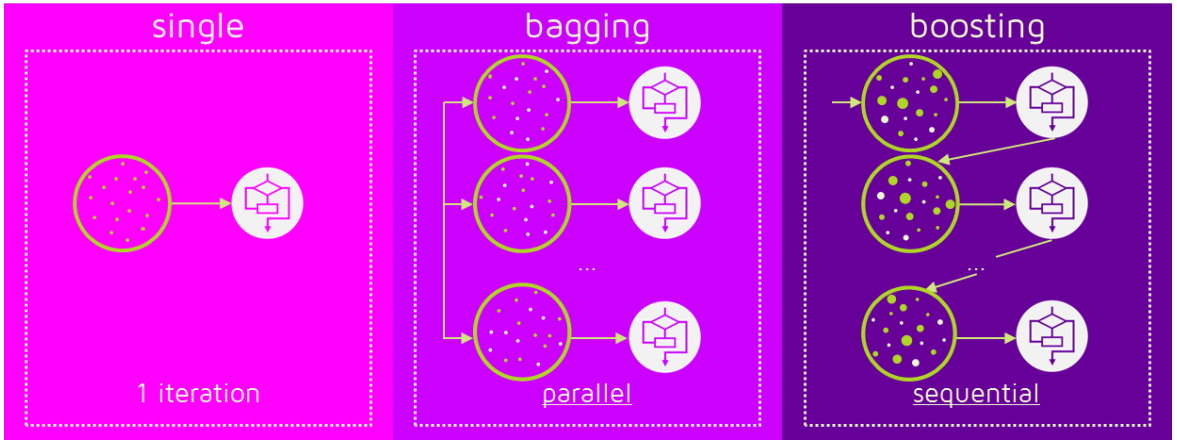
https://medium.com/@Zelros/a-brief-history-of-machine-learning-models-explainability-f1c3301be9dc

Ensembles of interpretable models: Random Forest & Boosted Trees

Ensembles of interpretable models: Random Forest & Boosted Trees

- **Random Forest**
  An unordered set of decision trees fitted on subsets of features and data (“bagging”). The final prediction is average / majority voting.

- **Boosted Trees**
  An ordered set of decision trees, where each subsequent tree is fitted to the prediction errors (residuals / incorrectly classified instances) of the previous trees (“boosting”). The final prediction is a weighted sum of the predictions.



https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

Random Forest & Boosted Trees: Feature Importance

- **Global impurity reduction / gain / Gini importance**
  Take average over trees

- **Local importance**
  Take average over trees

- Other global importance measures
  - **Weight.** The number of times a feature is used to split the data across all trees.
  - **Cover.** The number of times a feature is used to split the data across all trees weighted by the number of training data points that go through those splits.

Problems with Decision Trees & Ensembles

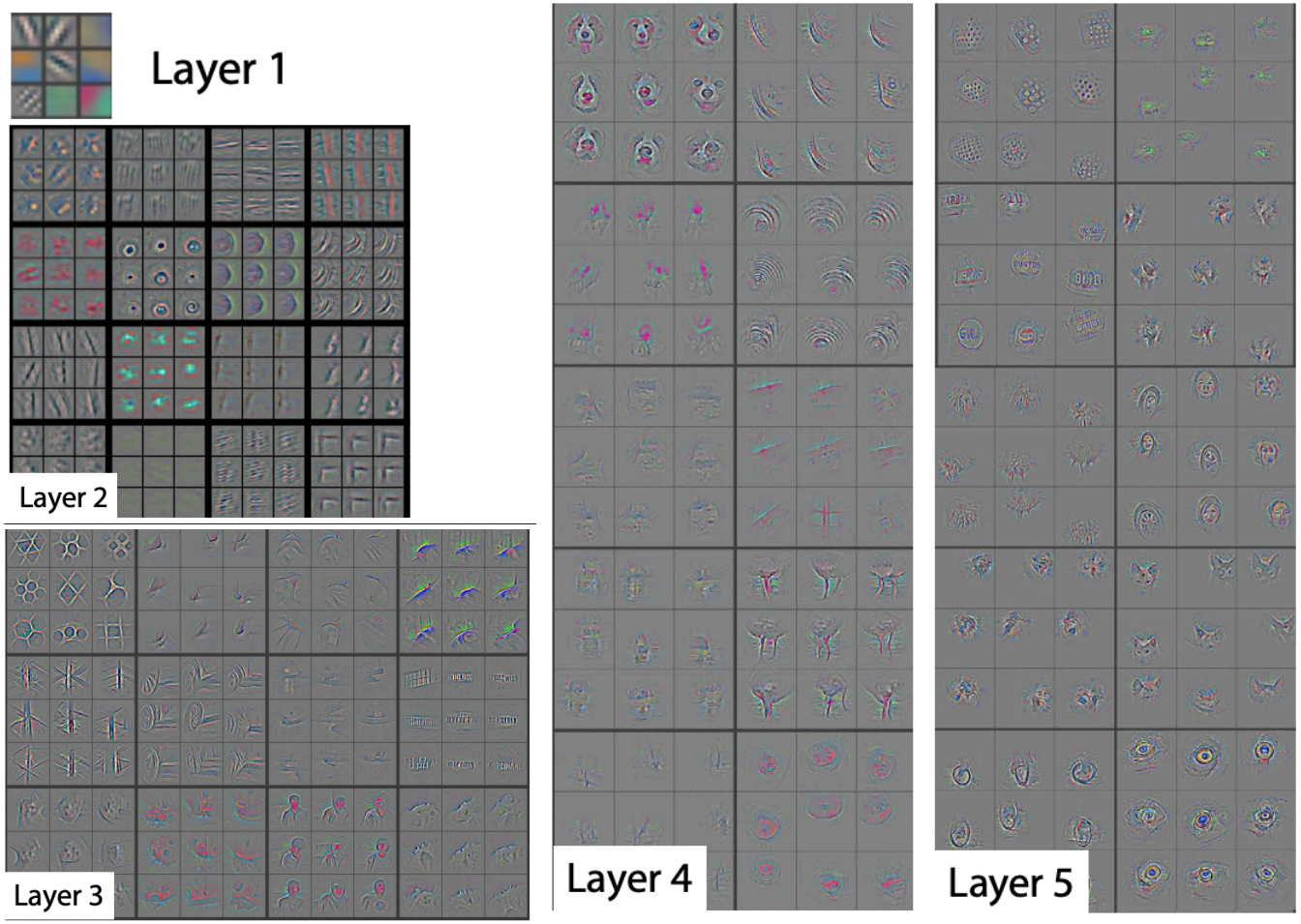
- **Impurity decrease**
  Categorical variables with many levels and continuous variables will dominate. Especially this is prominent when the importance is calculated on the train set (what sklearn does)
  https://explained.ai/rf-importance/
  - Overfitting to a random noise
- **Inconsistency between the methods, low stability**
  https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27
- …

- **Bottom line: The methods are useful but be cautious!**

- There are better / other (model-agnostic) methods:
  - Permutation importance (“shuffle column”)
  - Feature removal (“drop column”)

Post hoc approaches for neural networks

Post hoc approaches for neural networks: Global: Activation patterns (CNN)



https://arxiv.org/pdf/1311.2901.pdf

Post hoc approaches for neural networks: Global: Activation patterns (RNN)

“URL” neuron

```
ttp://www.ynetnews.com/] English-language website of Israel's lar
tp://www.bachets.com/  -xglish linguagesairsite of tslaelis sing
 d :xne.waea..awatoa. s &ntiaca-sardeelh oan tbisanfanreif ' aatd
mw-2 piiisoessis./ern.c](dceen epesaaiki ieeledh,irthraonse ,cose
dr.<:ahb-nptwt.xi ghf/ma)Tvdryzi couedlsu:tha-oo tu,stuif lvepery
stp,tcoa2drulwoclensr]p.llvaod,,eytc-n dm-oibuvs]bb imsultta tlybn
```

“[ ]” neuron

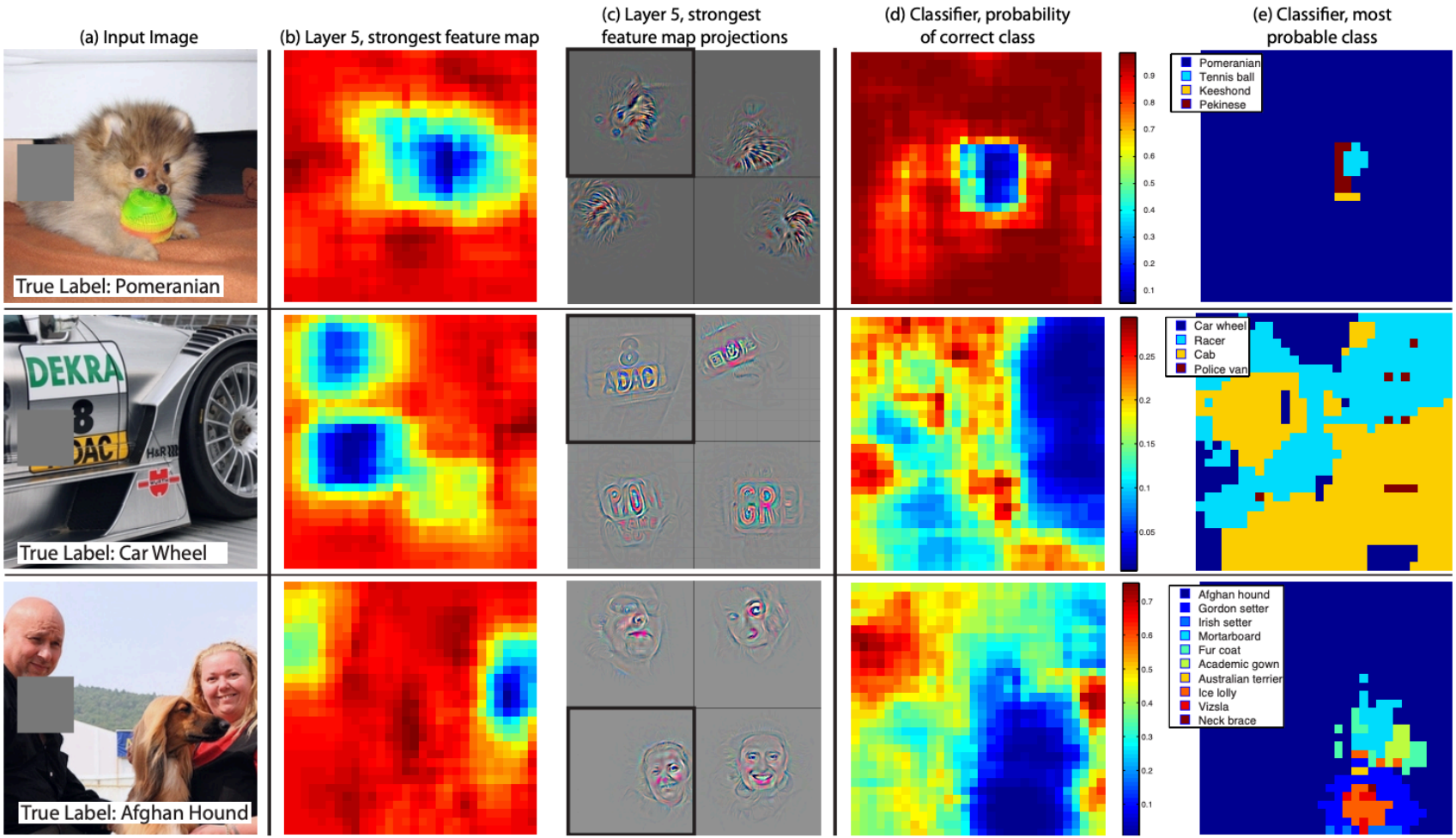
```
glish [[weekly newspaper]] * ' ' [[YNetNews]] ' ' [http://www.ynetnews.c
lish c[Caakly]cawspaper]] * ' [hTaA at ]' ' (http://www.bachets.co
iaci-lhSoip]i sec ]enp]s . ' ' [Co *wess] s a[a d :xne.waea..awatoa
eena,pCciietnedlox]gici|| s'[sAmFeSahon]t':: ,imomw-2 piiisoessis./er
syz .sfpenn alruel|rra . '#*: oDuFreiuep , :b1edr.<:ahb-nptwt.xi gh
a dpeamArbdeorpitee]dts-| T{[BaAvTpoSwao,. . .oacstp,tcoa2drulwoclens
```

“IF statement” neuron

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
    siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

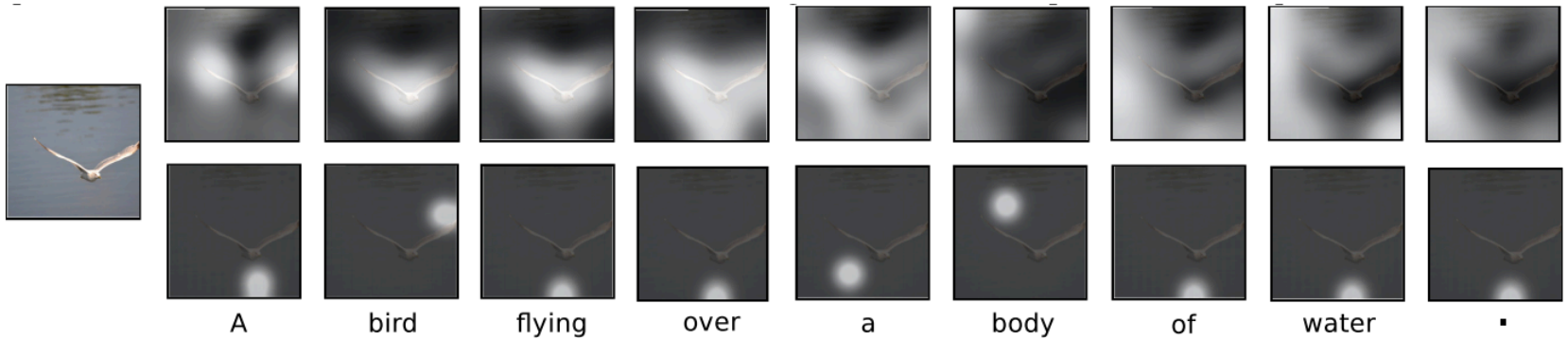
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Post hoc approaches for neural networks: Local: Input occlusion (CNN)



https://arxiv.org/pdf/1311.2901.pdf

Post hoc approaches for neural networks: Local: Attention (CNN, RNN)



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

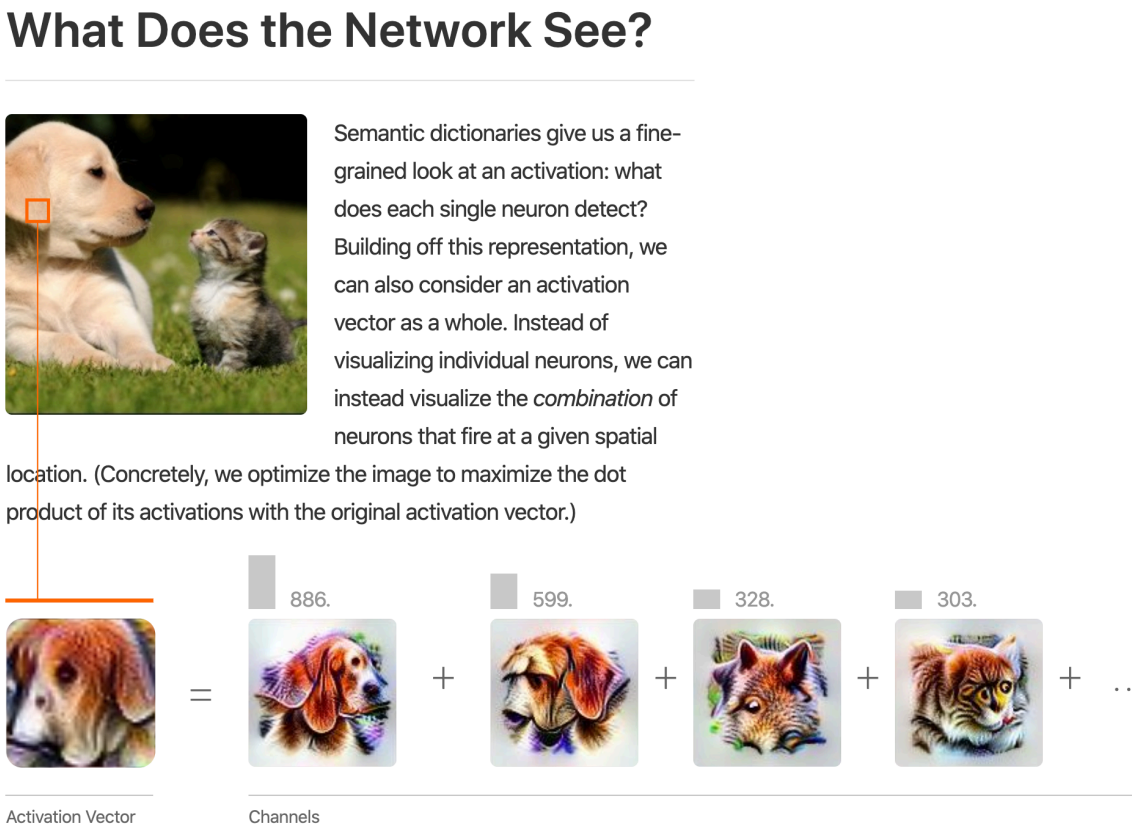
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

https://arxiv.org/pdf/1311.2901.pdf

Interactive visualisations from Google Research

- **The Building Blocks of Interpretability**
  https://distill.pub/2018/building-blocks/

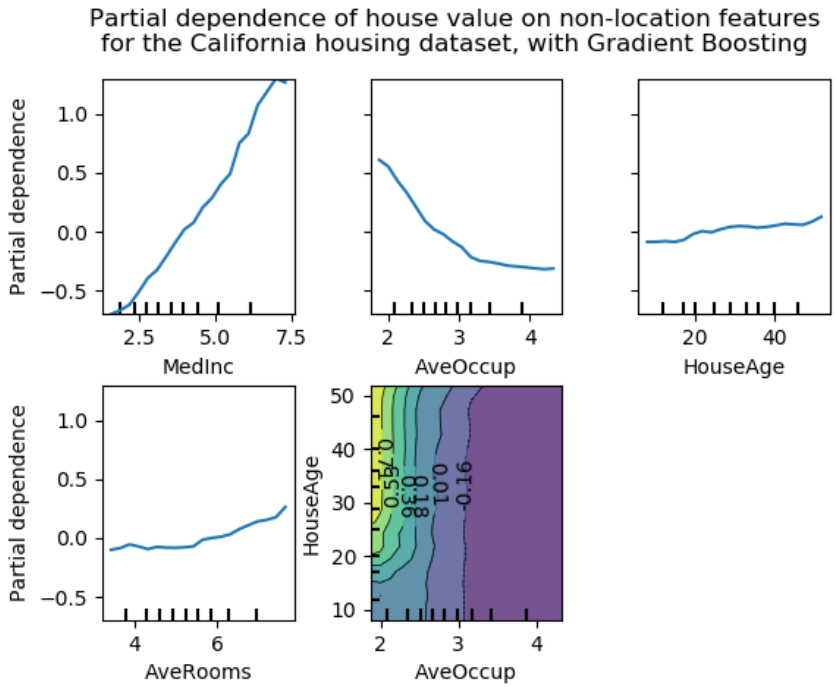


Global Model-Agnostic Methods

Partial dependence plots

Partial dependence plots

- Partial dependence plots (PDP) show the dependence between the model prediction and a set of features, marginalising over the values of all other features. Due to the limits of human perception the size of the plotted feature set must be small (usually, one or two) thus the plotted features are usually chosen among the most important features.



https://scikit-learn.org/stable/modules/partial_dependence.html

Permutation importance

Permutation importance

Data

Feature 1
Feature 2
Feature 3

→ **Model** → **Predictions** → **Performance metrics reference value** (e.g., MSE or F1)

Data

Randomly shuffle values

Feature 1
Feature 2
Feature 3

→ **Model** → **Predictions** → **Performance metrics new value** (e.g., MSE or F1)

Feature 2 importance = new value - reference value
(OR = new value / reference value)

Permutation importance

- The permutation feature importance measurement was introduced for Random Forests by Breiman (2001). Based on this idea, Fisher, Rudin, and Dominici (2018) proposed a model-agnostic version of the feature importance — they called it Model Reliance.

- Calculate on a validation set

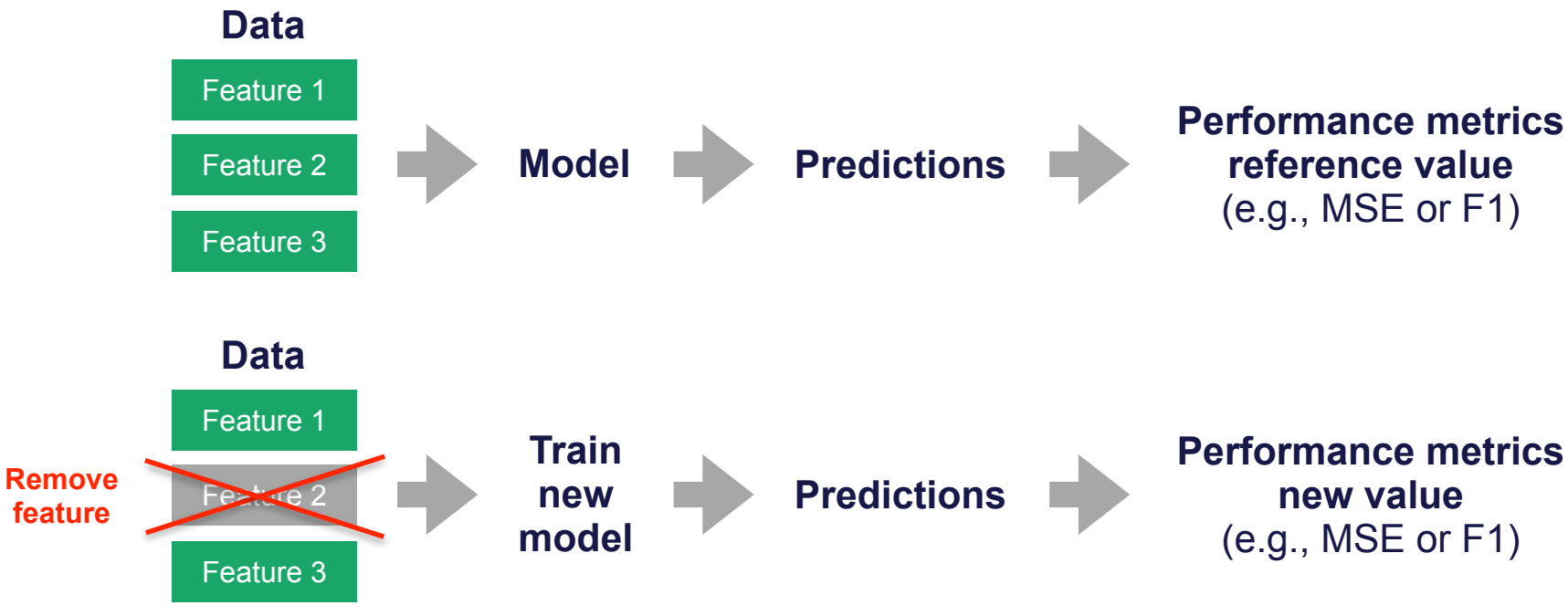
- Better than model-specific random forest importances https://explained.ai/rf-importance/

- Problems with multicollinear or correlated features https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance_multicollinear.html

https://towardsdatascience.com/human-interpretable-machine-learning-part-1-the-need-and-importance-of-model-interpretation-2ed758f5f476
https://towardsdatascience.com/explainable-artificial-intelligence-part-2-model-interpretation-strategies-75d4afa6b739
https://towardsdatascience.com/explainable-artificial-intelligence-part-3-hands-on-machine-learning-model-interpretation-e8ebe5afc608

Drop feature importance

Drop feature importance



Feature 2 importance = new value - reference value
(OR = new value / reference value)

Drop feature importance

- More accurate than permutation importance (less overfitting)
- Requires retraining the model for each feature (slow)

- Retrain on the training set and calculate on a validation set

- The same problems with multicollinear or correlated features as for permutation importance

Global surrogate model

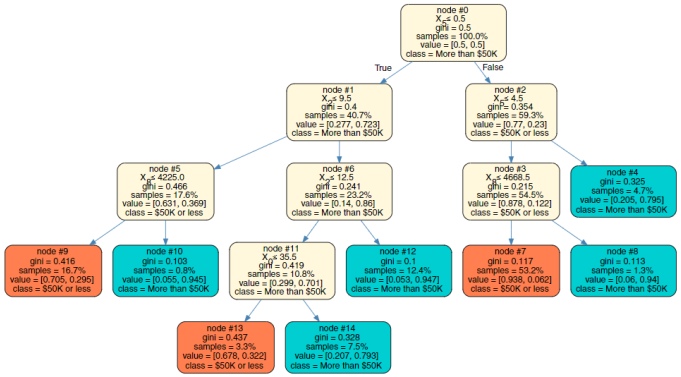
Global surrogate model

- A global surrogate model is an interpretable model that is trained to approximate the predictions of a black box model.
- The purpose of (interpretable) surrogate models is to approximate the predictions of the underlying model as closely as possible while being interpretable.
- Fit an interpretable proxy model (like logistic regression or decision tree) on the predictions of the model (either locally or globally). While the main model provides the exact predictions, the proxy model provides the corresponding approximate explanations and helps to understand the main model.

https://towardsdatascience.com/human-interpretable-machine-learning-part-1-the-need-and-importance-of-model-interpretation-2ed758f5f476
https://towardsdatascience.com/explainable-artificial-intelligence-part-2-model-interpretation-strategies-75d4afa6b739
https://towardsdatascience.com/explainable-artificial-intelligence-part-3-hands-on-machine-learning-model-interpretation-e8ebe5afc608

Global surrogate model: Algorithm

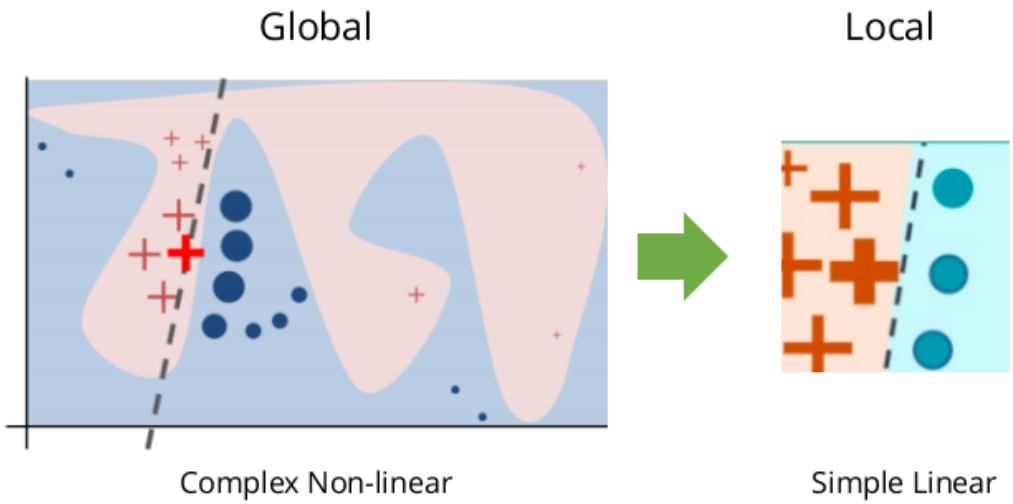
1. Choose a dataset: This could be the same dataset that was used for training the black box model or a new dataset from the same distribution. You could even choose a subset of the data or a grid of points, depending on your application.
2. For the chosen dataset, get the predictions of your base black box model.
3. Choose an interpretable surrogate model (linear model, decision tree, …).
4. Train the interpretable model on the dataset and the black-box predictions.
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 / visualise the surrogate model.



C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Local Model-Agnostic Methods

Intuition:
Complex functions are simple locally



$y = f(x)$

$y = f(x_0) + (x-x_0)\, df/dx + O(x^2)$

https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

Intuition:
Complex functions are simple locally

- In linear regression models, Beta coefficients explain the prediction for all the data points (if variable value increases by 1, prediction increases by Beta for every data point). This is global fidelity. In a causal analysis, it is called “average” causal analysis. However, this does not explain individual differences (heterogeneity). The effect of a variable for one user could be different from another user. This is local fidelity (explanation for individual data point or local subsection of the joint distribution of independent variables). Local function explanations (expectedly) could be more accurate than global explanations because, in a local region, the function (higher probability) is linear and monotonic. Even local regions could be highly non-linear and local explainability would fail — limitations of such methods. (This also is a part of smoothness assumption — without this, it is a cannot do many situations as many optimisation models would fail). LIME and SHAP explore and use the property of local explainability to build surrogate models to black-box machine learning models to provide them interpretability.

https://towardsdatascience.com/idea-behind-lime-and-shap-b603d35d34eb

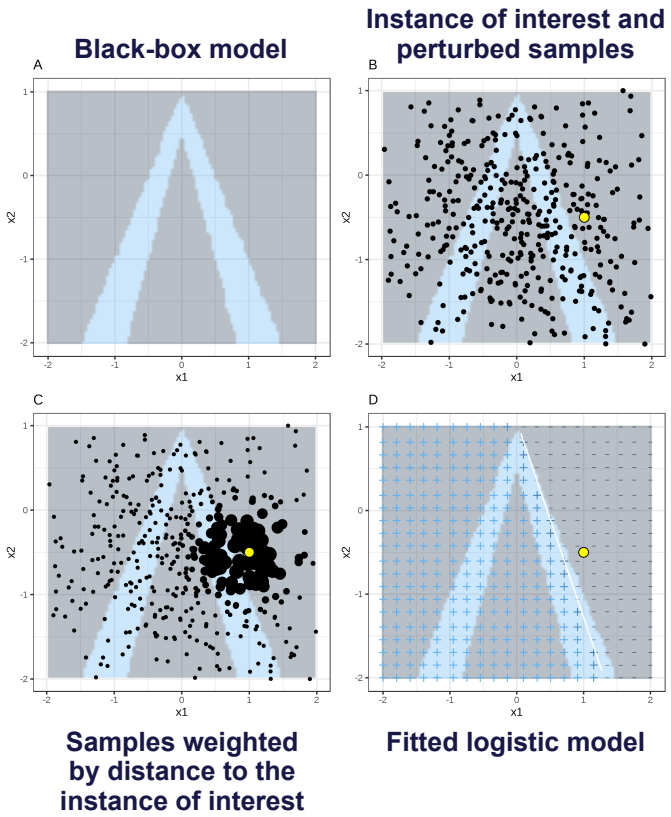
Local Interpretable Model agnostic Explanations (LIME)

LIME: Algorithm

1. Select your instance of interest for which you want to have an explanation of its black box prediction.
2. Perturb your instance and get the black box predictions for these new points.
3. Weight the new samples according to their proximity (distance) to the instance of interest (e.g., using the exp kernel)
4. Fit a weighted*, interpretable** model on the perturbed dataset.
5. Explain the prediction by interpreting the local model.

** the loss for each data point is weighted by its distance to the original instance*

*** e.g., linear regression (possibly regularised), decision tree, etc.*



C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

LIME: Examples

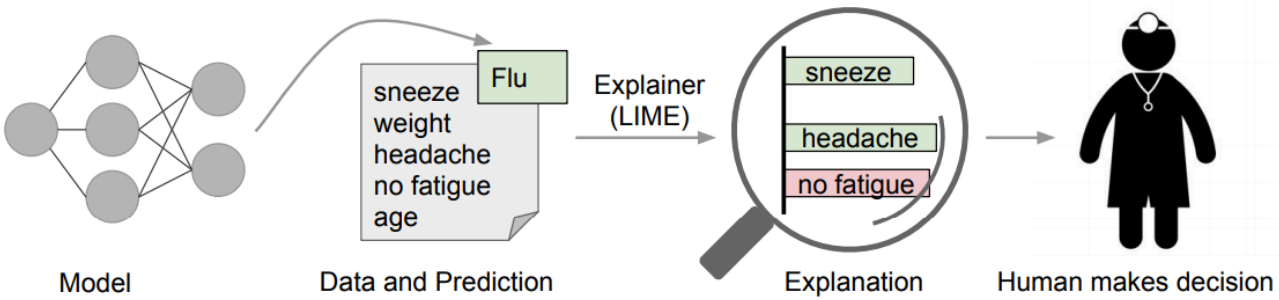


Figure 1: Explaining individual predictions. A model predicts that a patient has the flu, and LIME highlights the symptoms in the patient's history that led to the prediction. Sneeze and headache are portrayed as contributing to the "flu" prediction, while "no fatigue" is evidence against it. With these, a doctor can make an informed decision about whether to trust the model's prediction.

https://arxiv.org/abs/1602.04938

LIME: Examples

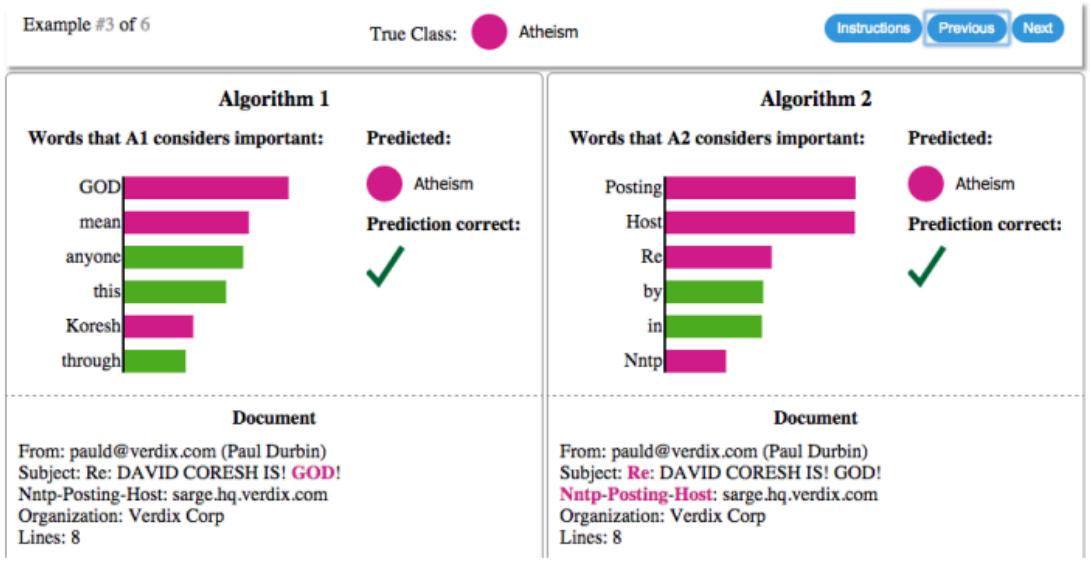
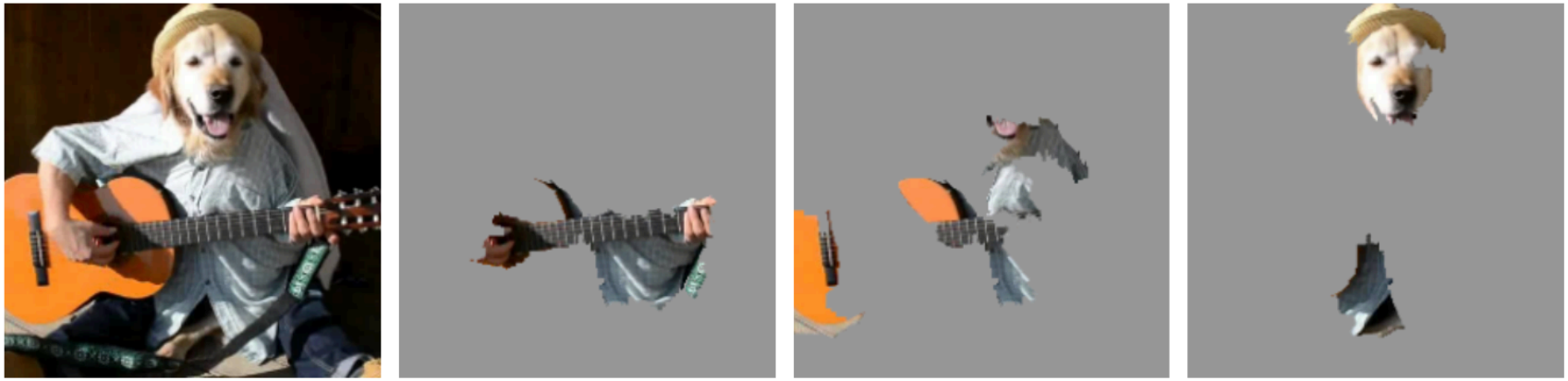


Figure 2: Explaining individual predictions of competing classifiers trying to determine if a document is about "Christianity" or "Atheism". The bar chart represents the importance given to the most relevant words, also highlighted in the text. Color indicates which class the word contributes to (green for "Christianity", magenta for "Atheism").

https://arxiv.org/abs/1602.04938

LIME: Examples



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

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

https://arxiv.org/abs/1602.04938

LIME: Examples

(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

https://arxiv.org/abs/1602.04938

LIME Package

Python package

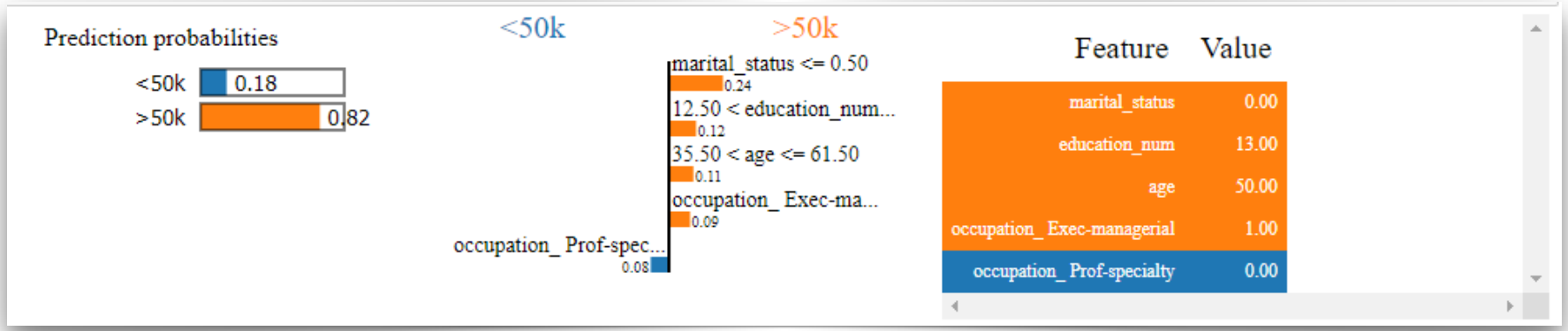
https://github.com/marcotcr/lime

```
pip install lime
```

LIME Package: Local explanations

```
marital_status                    Married
education_num                   Bachelors
hours_per_week                         40
fnlwgt                             167065
sex                                  Male
age                                    50
random                          0.0412146
workclass_ Private                      1
occupation_ Exec-managerial             1
race_ White                             1
Name: 23706, dtype: object

Label:  >50K
Prediction:  >50K
```

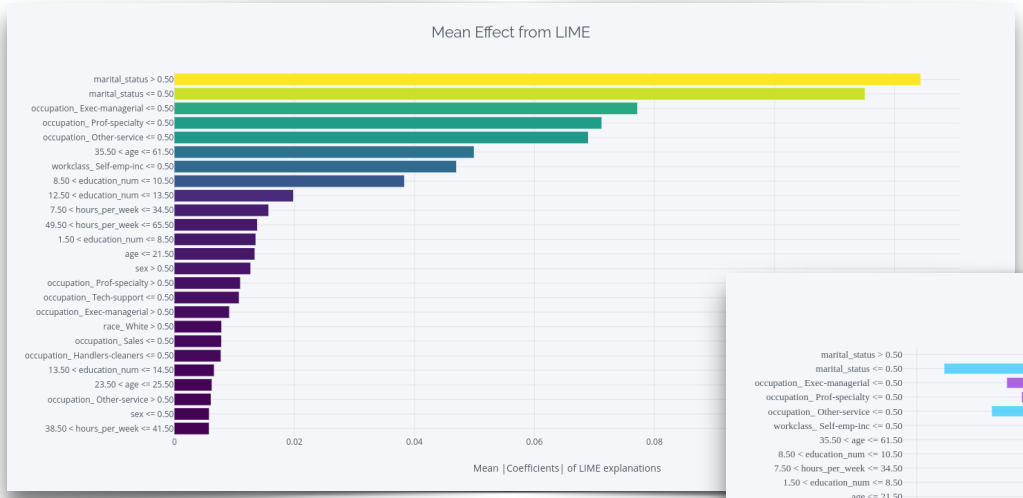
`lime.lime_tabular.LimeTabularExplainer.explain_instance`

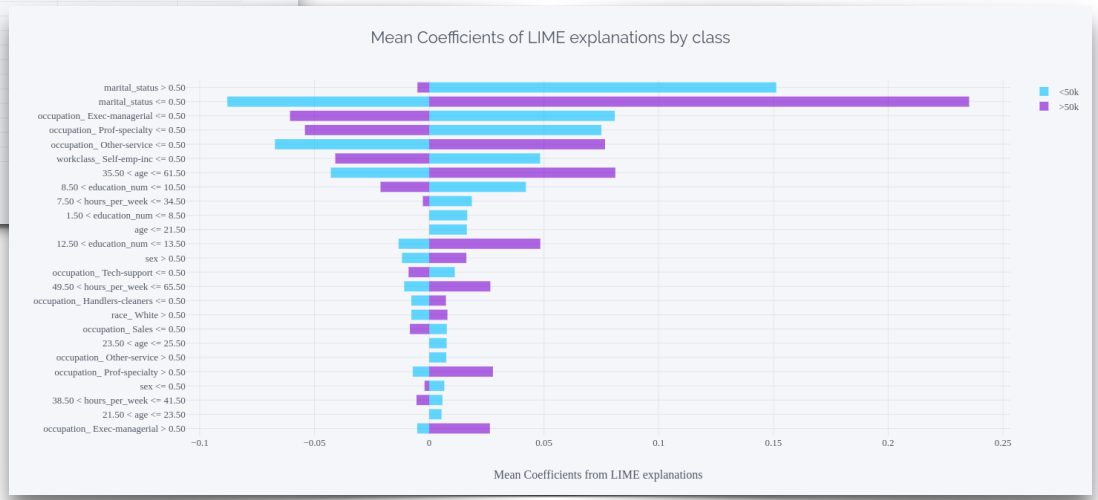


https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

LIME Package: Aggregated local (~global) view

`lime.submodular_pick.SubmodularPick`





https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

LIME: Summary

- **Advantages**
  + Easy to use and understand
  + Computationally fast

- **Disadvantages**
  - How to define neighbourhood and perturb original data point?
  - How to define distance between the original and perturbed samples?
    Exp kernel with h=0.75 is used in the original code by default
  - Instability of explanations

- **Paper**
  "Why Should I Trust You?": Explaining the Predictions of Any Classifier
  https://arxiv.org/abs/1602.04938

- **Python package**
  https://github.com/marcotcr/lime
  `pip install lime`

Shapley values & SHAP

Shapley values

Payout function

$$v(c) = \begin{cases} 80, & \text{if } c = \{A\} \\ 56, & \text{if } c = \{B\} \\ 70, & \text{if } c = \{C\} \\ 80, & \text{if } c = \{A, B\} \\ 85, & \text{if } c = \{A, C\} \\ 72, & \text{if } c = \{B, C\} \\ 90, & \text{if } c = \{A, B, C\} \end{cases}$$

Shapley values

Payout function

$$v(c) = \begin{cases} 80, & \text{if } c = \{A\} \\ 56, & \text{if } c = \{B\} \\ 70, & \text{if } c = \{C\} \\ 80, & \text{if } c = \{A, B\} \\ 85, & \text{if } c = \{A, C\} \\ 72, & \text{if } c = \{B, C\} \\ 90, & \text{if } c = \{A, B, C\} \end{cases}$$

Problem:
A, B, and C work together and earn 90.
How to distribute 90 among A, B, and C?

Shapley values

Payout function

$$v(c) = \begin{cases} 80, & \text{if } c = \{A\} \\ 56, & \text{if } c = \{B\} \\ 70, & \text{if } c = \{C\} \\ 80, & \text{if } c = \{A, B\} \\ 85, & \text{if } c = \{A, C\} \\ 72, & \text{if } c = \{B, C\} \\ 90, & \text{if } c = \{A, B, C\} \end{cases}$$

Problem:
A, B, and C work together and earn 90.
How to distribute 90 among A, B, and C?

Solution:
Marginal contribution of each player
averaged over all possible coalitions

Shapley values

Payout function

$$v(c) = \begin{cases} 80, & \text{if } c = \{A\} \\ 56, & \text{if } c = \{B\} \\ 70, & \text{if } c = \{C\} \\ 80, & \text{if } c = \{A, B\} \\ 85, & \text{if } c = \{A, C\} \\ 72, & \text{if } c = \{B, C\} \\ 90, & \text{if } c = \{A, B, C\} \end{cases}$$

Marginal contributions

π	δ_π^G
(A, B, C)	$(80, 0, 10)$
(A, C, B)	$(80, 5, 5)$
(B, A, C)	$(24, 56, 10)$
(B, C, A)	$(18, 56, 16)$
(C, A, B)	$(15, 5, 70)$
(C, B, A)	$(18, 2, 70)$

Problem:
A, B, and C work together and earn 90.
How to distribute 90 among A, B, and C?

Solution:
Marginal contribution of each player averaged over all possible coalitions

Shapley values

Payout function

$$v(c) = \begin{cases} 80, & \text{if } c = \{A\} \\ 56, & \text{if } c = \{B\} \\ 70, & \text{if } c = \{C\} \\ 80, & \text{if } c = \{A, B\} \\ 85, & \text{if } c = \{A, C\} \\ 72, & \text{if } c = \{B, C\} \\ 90, & \text{if } c = \{A, B, C\} \end{cases}$$

Marginal contributions

π	δ_π^G
(A, B, C)	$(80, 0, 10)$
(A, C, B)	$(80, 5, 5)$
(B, A, C)	$(24, 56, 10)$
(B, C, A)	$(18, 56, 16)$
(C, A, B)	$(15, 5, 70)$
(C, B, A)	$(18, 2, 70)$

$$\phi_A = (80 + 80 + 24 + 18 + 15 + 18)/6 = 39.17$$
$$\phi_B = (0 + 5 + 56 + 56 + 5 + 2)/6 = 20.67$$
$$\phi_C = (10 + 5 + 10 + 16 + 70 + 70)/6 = 30.17$$

Shapley values

Payout function

$$v(c) = \begin{cases} 80, & \text{if } c = \{A\} \\ 56, & \text{if } c = \{B\} \\ 70, & \text{if } c = \{C\} \\ 80, & \text{if } c = \{A, B\} \\ 85, & \text{if } c = \{A, C\} \\ 72, & \text{if } c = \{B, C\} \\ 90, & \text{if } c = \{A, B, C\} \end{cases}$$

Marginal contributions

π	δ_π^G
(A, B, C)	$(80, 0, 10)$
(A, C, B)	$(80, 5, 5)$
(B, A, C)	$(24, 56, 10)$
(B, C, A)	$(18, 56, 16)$
(C, A, B)	$(15, 5, 70)$
(C, B, A)	$(18, 2, 70)$

Shapley values

$$\phi_A = (80 + 80 + 24 + 18 + 15 + 18)/6 = 39.17$$
$$\phi_B = (0 + 5 + 56 + 56 + 5 + 2)/6 = 20.67$$
$$\phi_C = (10 + 5 + 10 + 16 + 70 + 70)/6 = 30.17$$

Shapley values as local feature importance

- A model predicts apartment price, *price = f(apartment characteristics)*
- The average prediction for all apartments is €310,000
- Explain *f(park-nearby, cat-banned, area-50, floor-2nd)* = €300,000

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

Shapley values as local feature importance

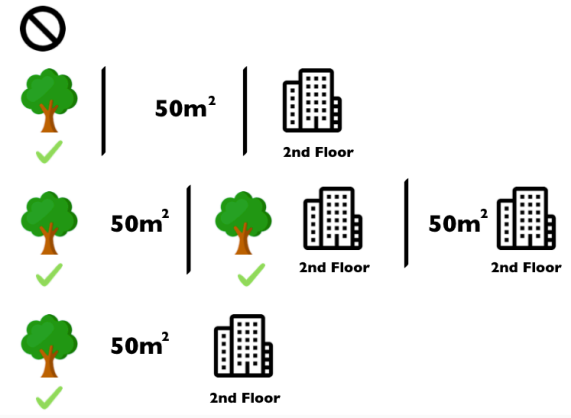
- A model predicts apartment price, *price = f(apartment characteristics)*
- The average prediction for all apartments is €310,000
- Explain *f(park-nearby, cat-banned, area-50, floor-2nd)* = €300,000

- Can be viewed as a game where features "collaborate" together to get the difference between the prediction and the average prediction (-€10,000)
  - For example, the *park-nearby* contributed €30,000; *size-50* contributed €10,000; *floor-2nd* contributed €0; *cat-banned* contributed -€50,000. The contributions add up to -€10,000, the final prediction minus the average predicted apartment price.
  - So we can calculate Shapley values for the feature values

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

Shapley values as local feature importance

- For example, to calculate Shapley value for *cat-banned,* we need to calculate the marginal contribution of *cat-banned* to the following coalitions:
  - No feature values
  - *park-nearby*
  - *size-50*
  - *floor-2nd*
  - *park-nearby* & *size-50*
  - *park-nearby* & *floor-2nd*
  - *size-50* & *floor-2nd*
  - *park-nearby* & *size-50* & *floor-2nd*



- For each of these coalitions, we compute the predicted apartment price with and without the feature value cat-banned and take the difference to get the marginal contribution. The Shapley value is the average of marginal contributions weighted by the number of ways the coalition can be obtained.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right]$$

C. Molnar, “Interpretable Machine Learning, A Guide for Making Black Box Models Explainable”

Shapley values as feature importance: Problems

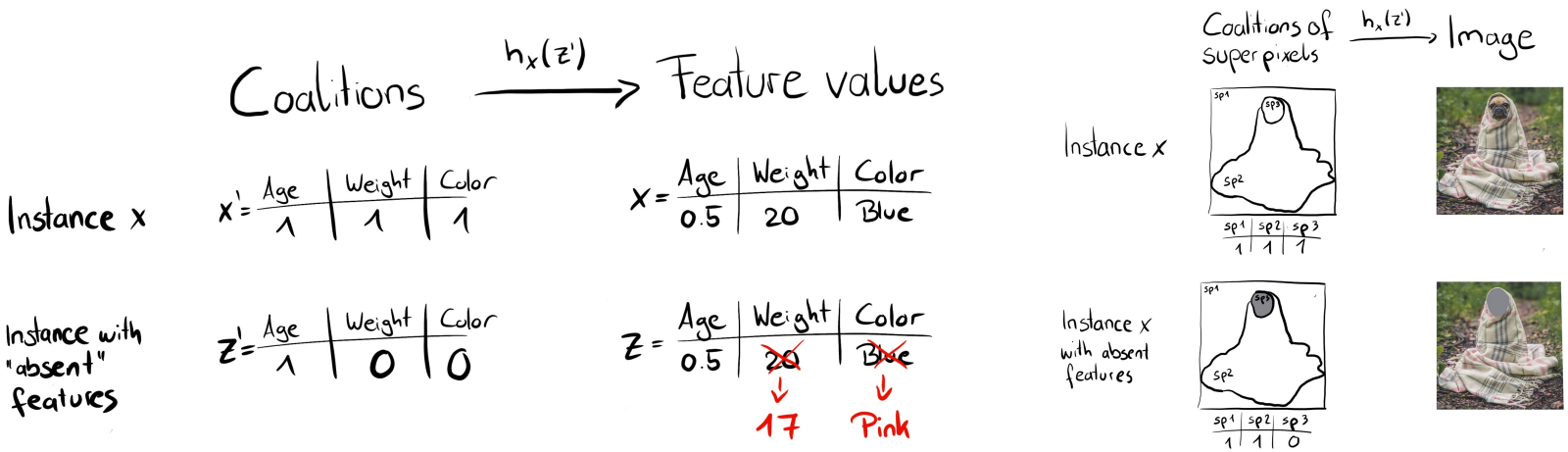
- **How to calculate a model's prediction for an incomplete coalition?**

- **Computational complexity: Number of possible coalitions is ~** 2^N

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

Shapley values as feature importance: Problems

- **How to calculate a model's prediction for an incomplete coalition?**

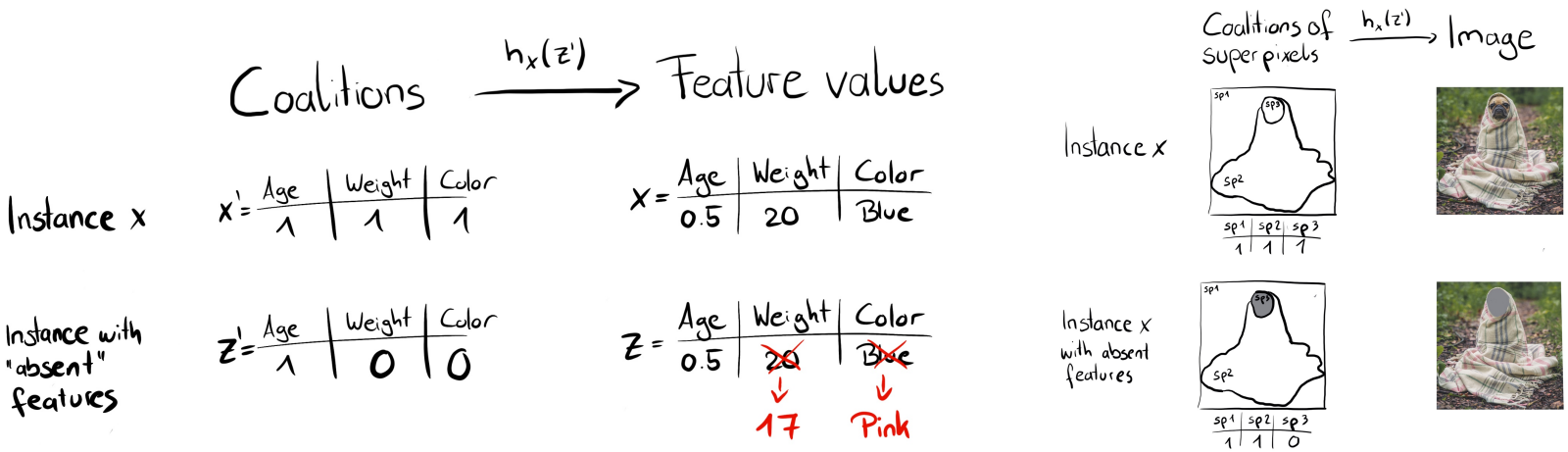


- **Computational complexity: Number of possible coalitions is ~** 2^N

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

Shapley values as feature importance: Problems

- **How to calculate a model's prediction for an incomplete coalition?**
  - Use mean values
  - Sampling from marginals (better but slower)

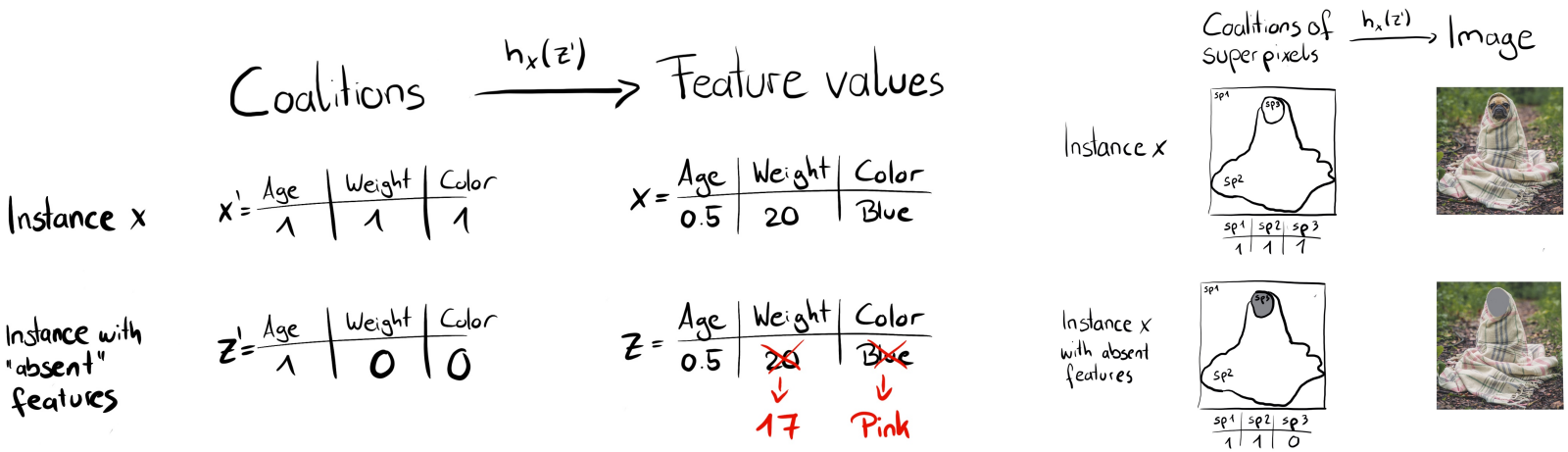


- **Computational complexity: Number of possible coalitions is ~** 2^N

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

Shapley values as feature importance: Problems

- **How to calculate a model's prediction for an incomplete coalition?**
  - Use mean values
  - Sampling from marginals (better but slower)



- **Computational complexity: Number of possible coalitions is ~** 2^N
  - Approximations, assumptions, model-specific tricks, and Monte Carlo Sampling → **SHAP (SHapley Additive exPlanations)**

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

SHAP (SHapley Additive exPlanations)

- **The main idea**
  - Represent nonlinear function around a sample as a linear combination of the feature contributions (~ LIME)
  - The coefficients of the linear model (feature contributions) are the approximated Shapley values for the features ("SHAP values”)
  - Computationally efficient approximation

- **Model-agnostic**
  - Kernel SHAP (LIME + Shapley values)

- **Model-specific**
  - Linear SHAP (Linear & Logistic Regression)
  - Tree SHAP (Decision Trees and Random Forests)
  - Gradient SHAP (Gradient Boosting models)
  - Deep SHAP (Neural Network models)

Kernel SHAP: Algorithm

1. Select your instance of interest ***z*** for which you want to have an explanation of its black box prediction.
2. Sample coalitions ***z'*** by randomly** excluding features.
3. Get a prediction for each coalition by first projecting the coalition back to the data space, for example, by sampling excluded values from the marginals, and then applying the black-box model.
4. Compute the weight for each coalition with the SHAP kernel* (~ how frequent the coalition is).
5. Fit weighted linear model to these new data points.
6. Shapley values are the coefficients from the linear model.

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

** LIME weights the instances according to how close they are to the original instance. The more 0's in the coalition vector, the smaller the weight in LIME. SHAP weights the sampled instances according to the weight the coalition would get in the Shapley value estimation. Small coalitions (few 1's) and large coalitions (i.e. many 1's) get the largest weights.*

*** We can be a bit smarter about the sampling of coalitions: The smallest and largest coalitions take up most of the weight. We get better Shapley value estimates by using some of the sampling budget K to include these high-weight coalitions instead of sampling blindly.*

C. Molnar, "Interpretable Machine Learning, A Guide for Making Black Box Models Explainable"

SHAP: Notes

- Shapley value guarantees a fair distribution of contribution for each of the variables (LIME do not provide the guarantee).
- LIME assumes that the local model is linear, SHAP does not have any such assumptions.
- SHAP value calculation is very time expensive (as it checks all the possible combinations: it does it through monte Carlo simulations rather than brute force).
- SHAP value is NOT the difference between the prediction with and without a variable, rather it is a contribution of a variable to the difference between the actual prediction and the mean prediction.
- The variable importance at a global level is given by adding the absolute value of the SHAP values for each individual data point.
- Although SHAP uses all the variables, we can select some variables with higher variable importance, drop other variables and rerun SHAP if we want to reduce the number of variables (because of the property like consistency, the order of variable importance will not change so less important variables can be ignored).

https://towardsdatascience.com/idea-behind-lime-and-shap-b603d35d34eb

SHAP Package

Python package

https://github.com/slundberg/shap

```
pip install shap
```

SHAP Package: Local explanations

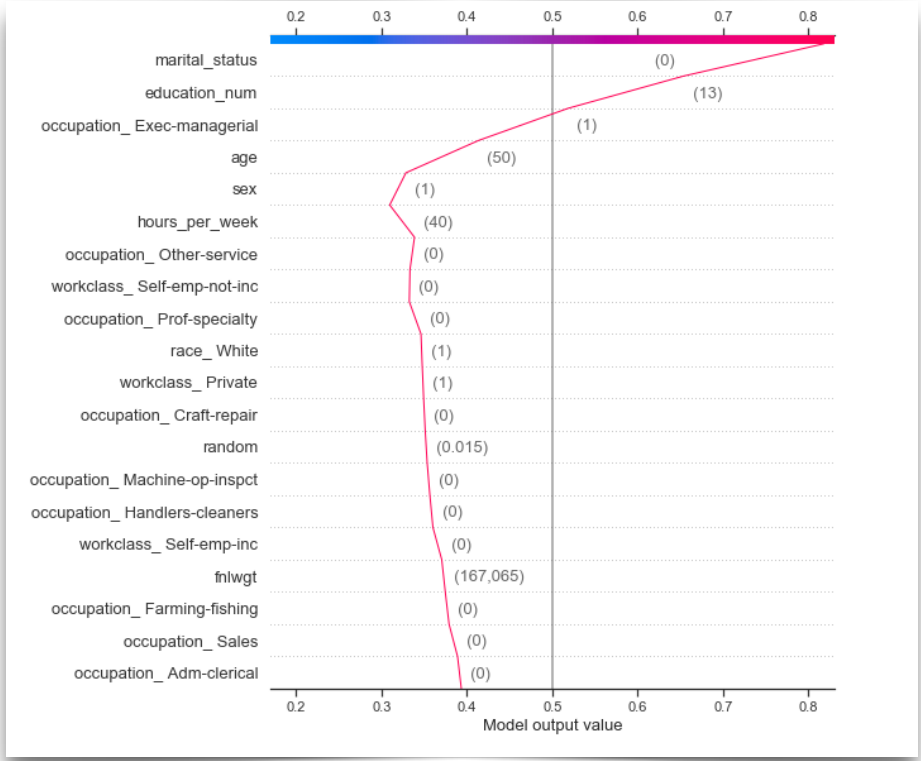
```
marital_status                    Married
education_num                   Bachelors
hours_per_week                         40
fnlwgt                             167065
sex                                  Male
age                                    50
random                          0.0412146
workclass_ Private                      1
occupation_ Exec-managerial             1
race_ White                             1
Name: 23706, dtype: object

Label:  >50K
Prediction:  >50K
```

`shap.TreeExplainer.force_plot`

https://github.com/slundberg/shap
https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

SHAP Package: Local explanations

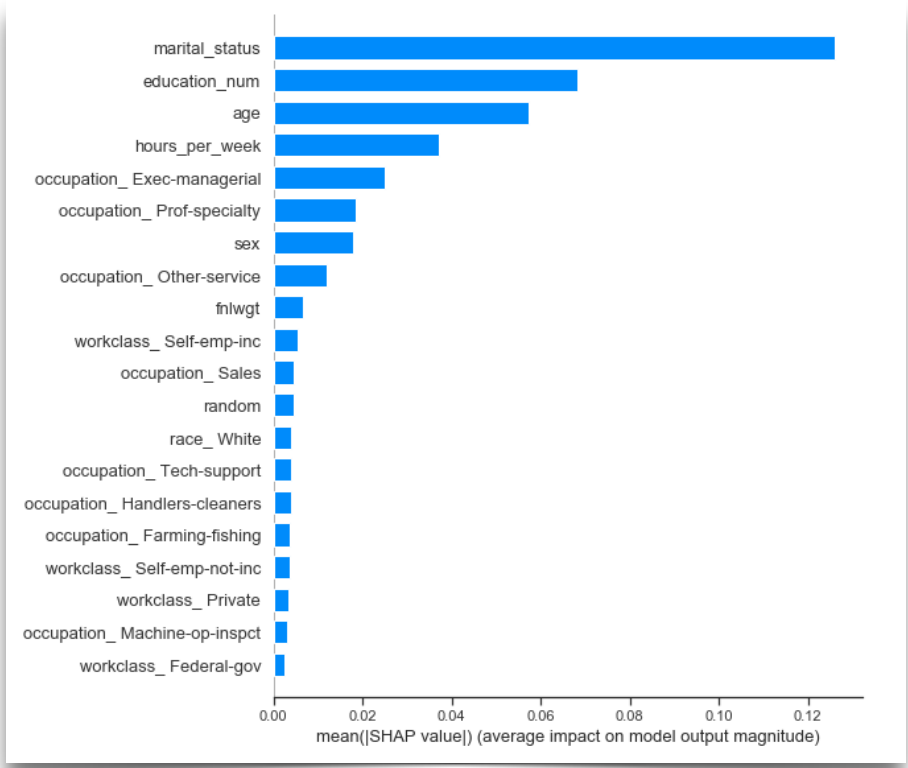
`shap.TreeExplainer.decision_plot`



https://github.com/slundberg/shap
https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

SHAP Package: Aggregated local (~global) view

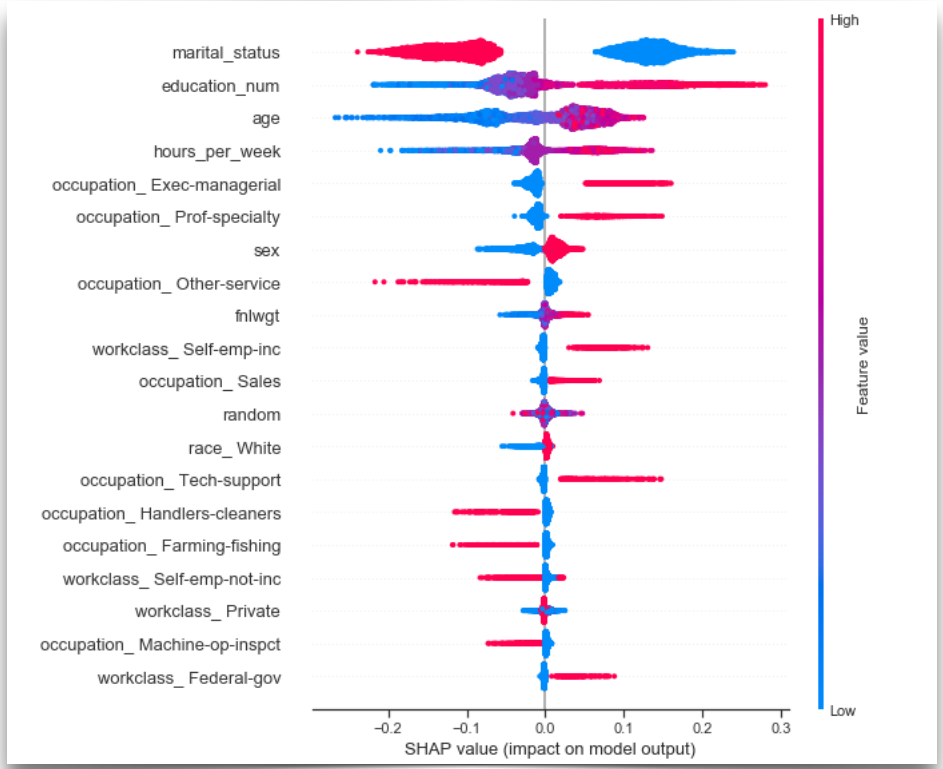
```
shap.TreeExplainer.summary_plot(plot_type='bar')
```



https://github.com/slundberg/shap
https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

SHAP Package: Aggregated local (~global) view

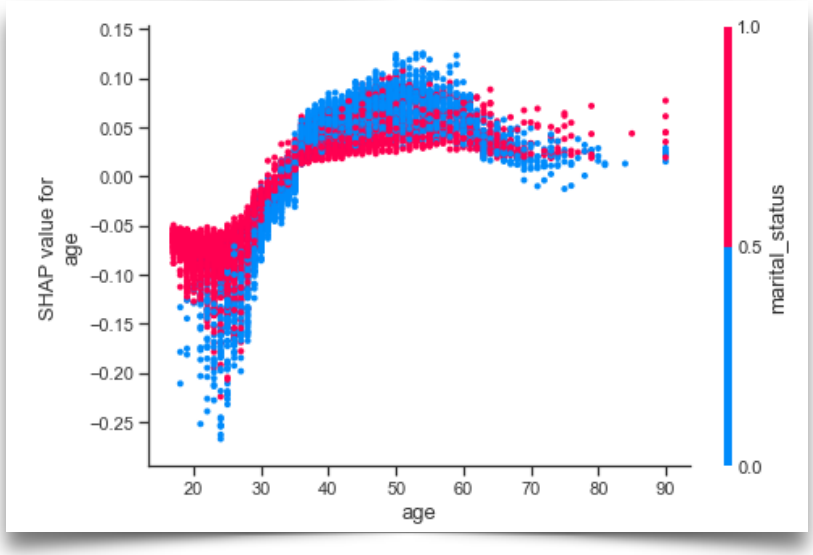
```
shap.TreeExplainer.summary_plot(plot_type='dot')
```

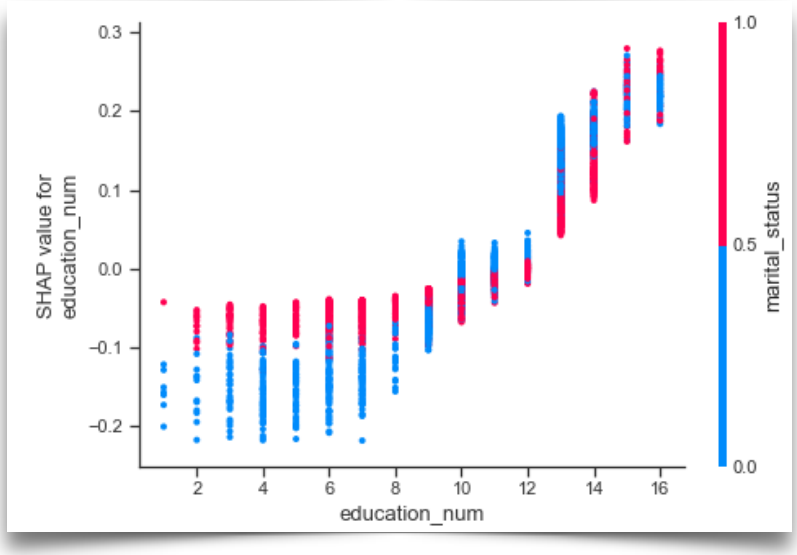


https://github.com/slundberg/shap
https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

SHAP Package: Aggregated local (~global) view

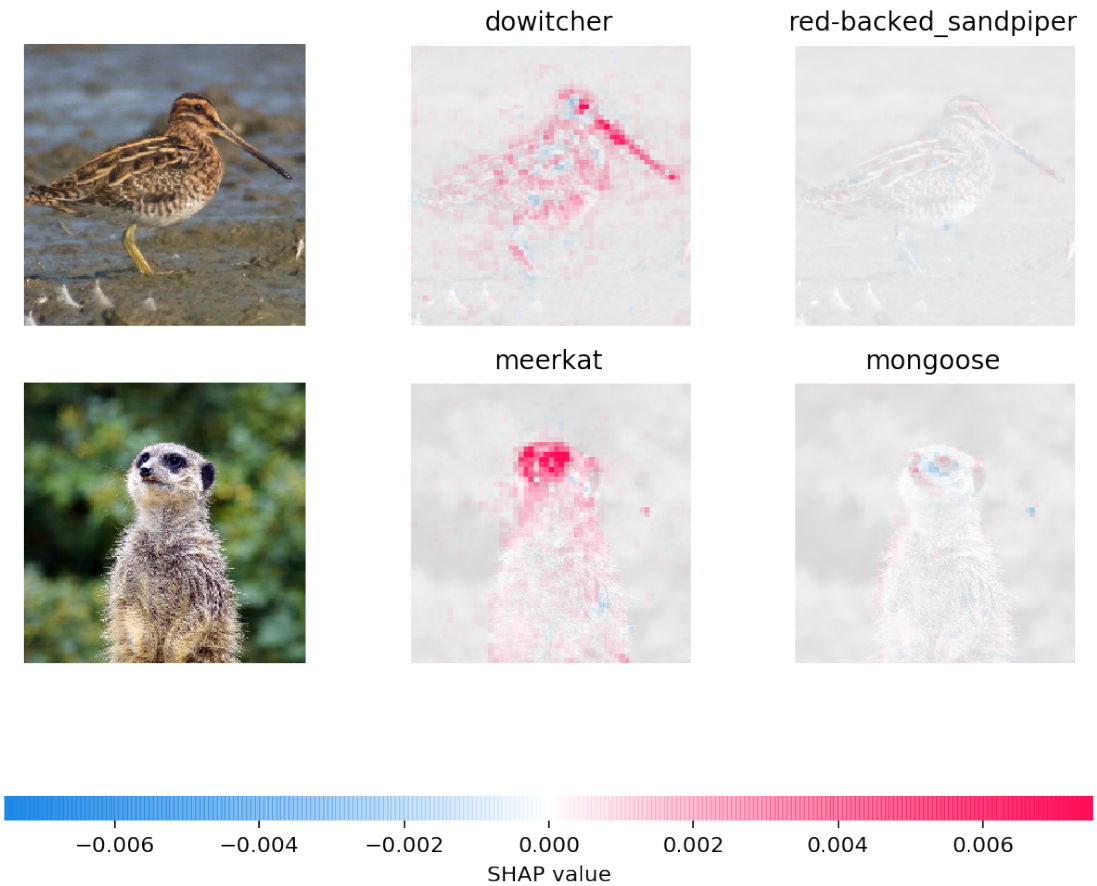
```
shap.TreeExplainer.dependence_plot
```





https://github.com/slundberg/shap
https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

SHAP Package:
Local explanations for images, text, …



https://github.com/slundberg/shap
https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html

SHAP: Summary

- **Advantages**
  + Solid theoretical foundations
  + Built on top of LIME
  + More robust than LIME

- **Disadvantages**
  - Computationally demanding (especially, Kernel SHAP)

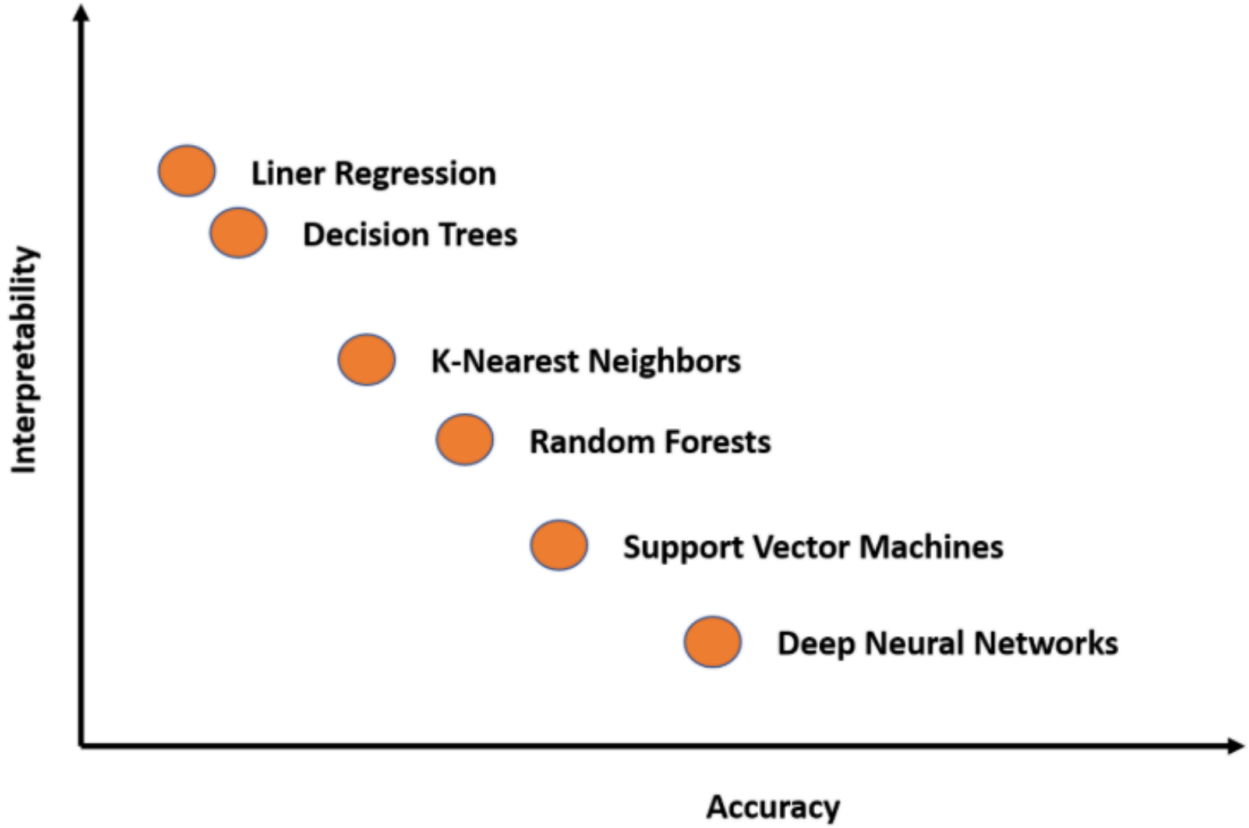
- **Theory on Shapley values**
  https://www.youtube.com/watch?v=w9O0fkfMkx0
  https://www.youtube.com/watch?v=qcLZMYPdpH4

- **Paper**
  A Unified Approach to Interpreting Model Predictions
  https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

- **Python package**
  https://github.com/slundberg/shap
  `pip install shap`

Final Remarks

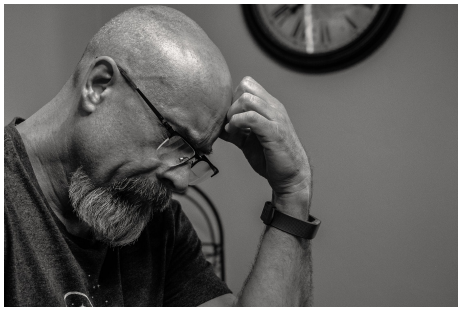
Accuracy-interpretability tradeoff

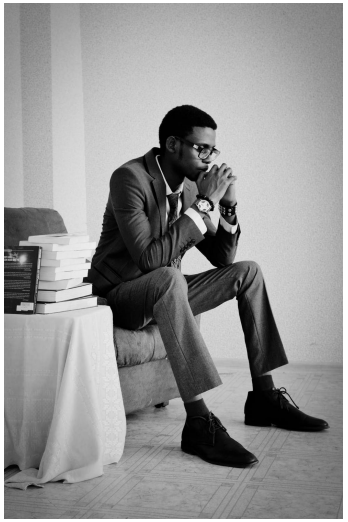


https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/

Finding the right balance is difficult…

- **Two alternatives:**
  - An autonomous car driven by a black-box AI with the fatal incident risk of 0.001%
  - An autonomous car which can kill you with 10% probability, however you *will* know exactly why





Exercises

- “Part 1 - Basics - RandomForest.ipynb”
- “Part 2 - Advanced - xgboost.ipynb”
- [optional] https://www.kaggle.com/learn/machine-learning-explainability

Recommended reading

- C. Molnar, “Interpretable Machine Learning” https://christophm.github.io/interpretable-ml-book/
- Other articles and posts cited in the presentation