Responsible Machine Learning

Lecture 6: Model Explainability/Interpretability

CS 4973-05

Fall 2023

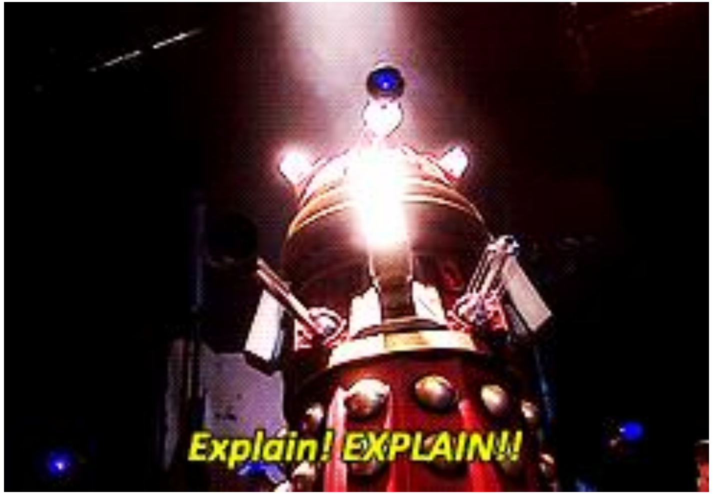
Instructor: Avijit Ghosh

ghosh.a@northeastern.edu

Northeastern University, Boston, MA

Today's lecture slide credits: Sarthak Saha, Northeastern and Giovanni Bruner, University of Dundee

Interpretability in Machine Learning



The current state of machine learning

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Why Interpret ?

Why Interpretability is important…

- Often a model is as good as the **insights** it allows to you to gather on a business problem, other than the prediction itself.

- Being able to be transparent about the output of your model may be required by law…think at GDPR **right of explanation**.

- You may want to make sure that your model is not picking up a racial, gender or religion **bias**. What if your model always refuses a loan to black people?

- Your model might be predicting the **right thing**, for a completely wrong reason! Want an example? Go the the next slide.

Interpretability

- There are several definition of **interpretability** in the context of a Machine Learning model. The one I like the most is Interpretability as trust.

- Trust that the model is predicting a certain value for the "*right reasons*".

- Interpretability is key to ensure the **social acceptance** of Machine Learning algorithms in our everyday life.

Cause nobody wants to deal with Carol Beer, right ?

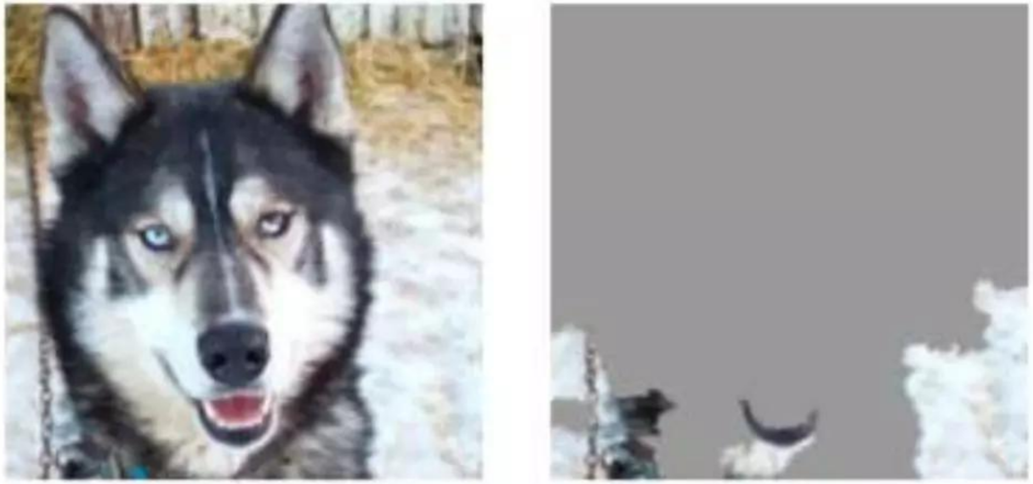
Would you trust this model?

- It’s possible to build a model that is very accurate, but it loses its power if we are unable to explain why a certain prediction was issued.

- In the Husky vs Wolves experiment researchers built an image recognition model that could correctly classify a Husky from a Wolf with very high accuracy.

- However, after using an explanation method researchers found out that this was due to all wolves having a snowy background!

- Would you trust this model?



(a) Husky classified as wolf (b) Explanation

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

Interpretability in practice.

A Machine Learning model works with a set of features in a multi dimensional space with the objective to minimize a function or maximizing a likelihood.

It’s like a **game**, with a set of **players** (our players) trying to reach an objective (a correct prediction). **We need to able to understand which players contributed the most to the objective.**

In general, it seems like there are few fundamental problems –

- We don’t trust the models
- We don’t know what happens in extreme cases
- Mistakes can be expensive / harmful
- Does the model makes similar mistakes as humans ?
- How to change model when things go wrong ?

Interpretability is one way we try to deal with these problems

Three key characteristics of a good feature attribution model

1) **Consistency*:** If we change our model so that it relies more on a feature, we expect that the importance of this feature does not decrease.

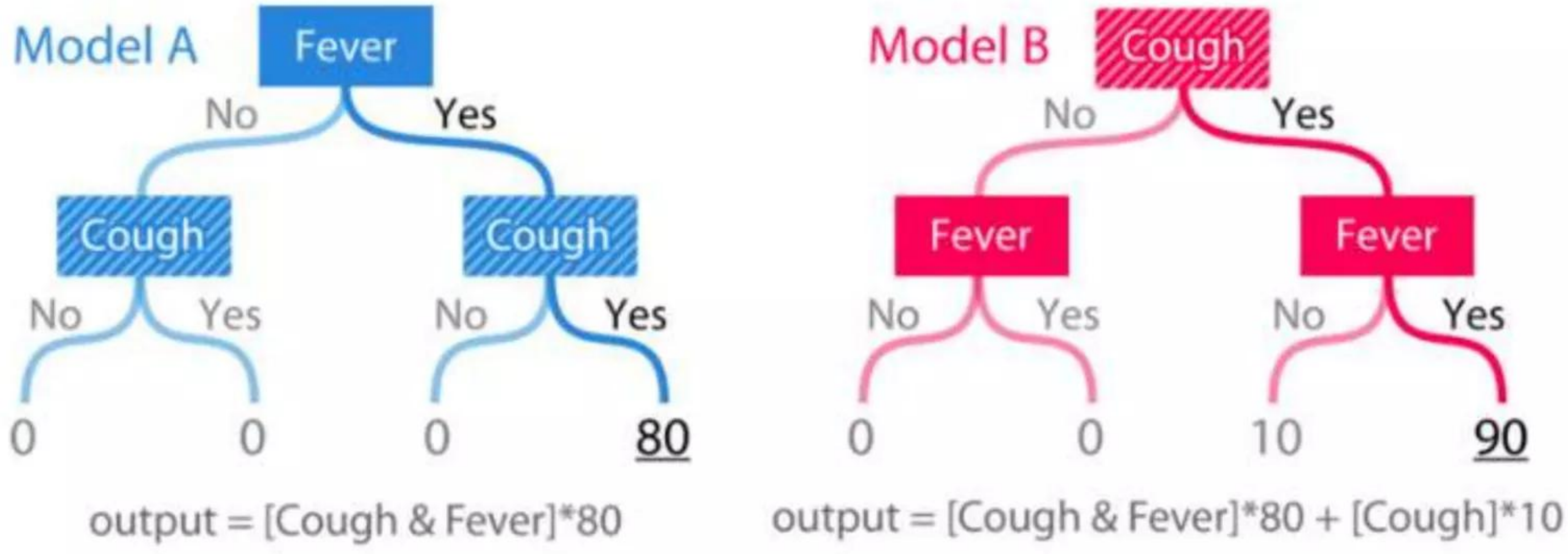
2) **Accuracy*:** If we have chosen a metric to measure the importance of a model, then the attribution of each feature should add up to that metric.

3) **Insightfulness:** Just getting a feature importance ranking is not enough. We need to understand if a feature contributed to lower or increase our model output scores.

*https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

What about consistency?

Let’s take two simple models to estimate if a person has flu symptoms…This model classifies each observation perfectly.

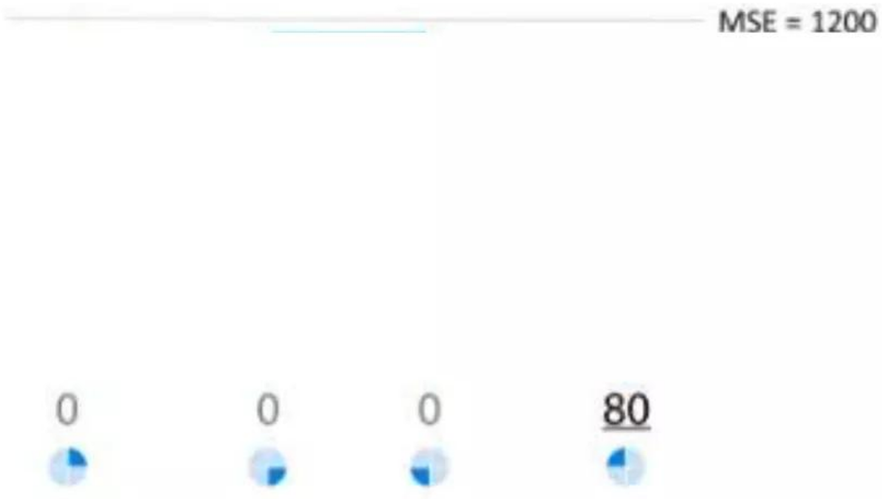


Simple tree models over two features. Cough is clearly more important in model B than model A.

*https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

Consistency

Imagine that we have 4 observations and that they all finish in the correct leaf. We use Mean Squared Error as a metric.

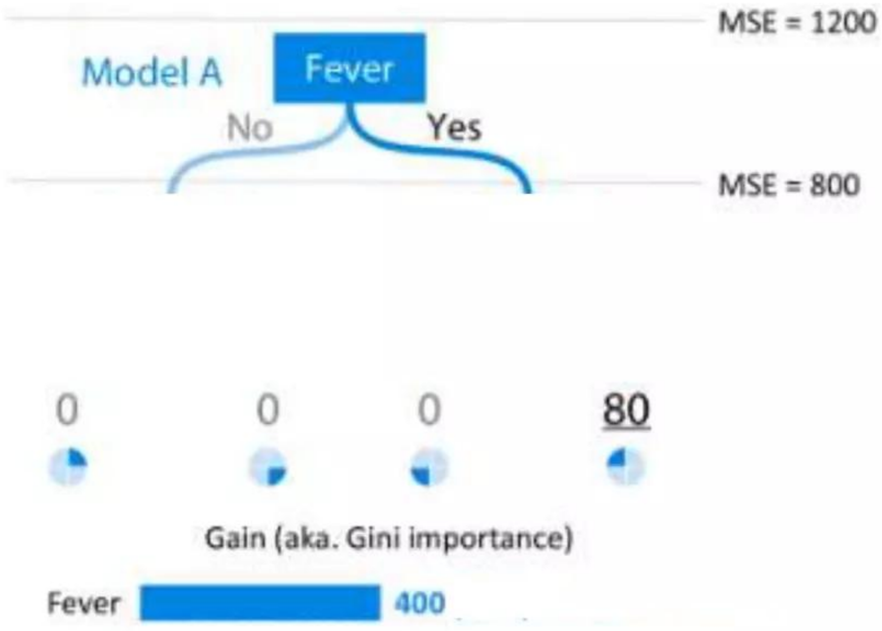


Step 1: Before doing any split we could assign a mean score of 20 to each of the 4 observations.
MSE = (((0-20)**2) + ((0-20)**2) + ((0-20)**2) + ((80-20)**2)) = 1200

*https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

Consistency

Imagine that we have 4 observations and that they all finish in the correct leaf. We use Mean Squared Error as a metric.



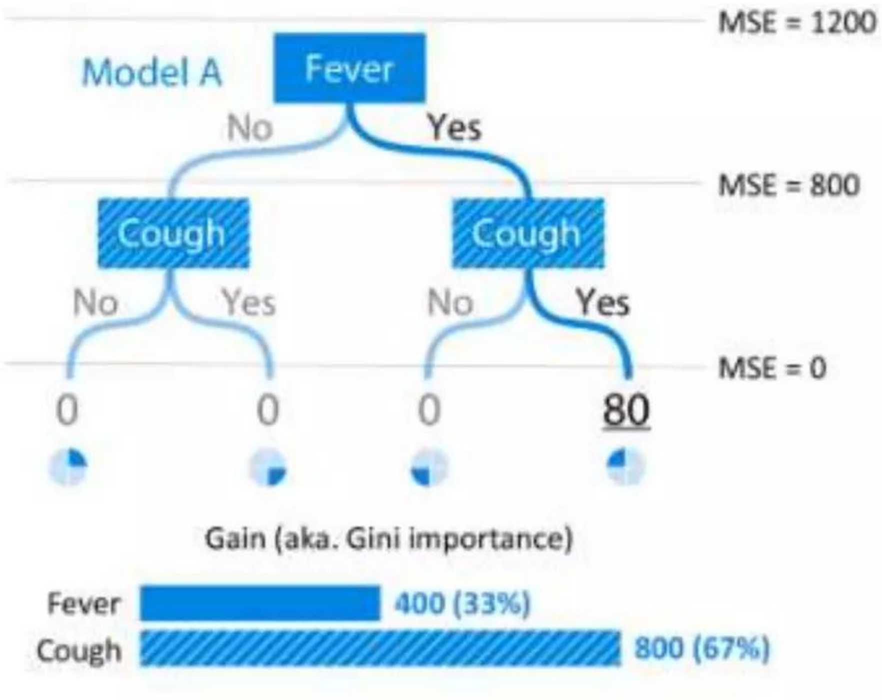
Step 2: We use ‘Fever’ to split the data, two observations go to the right, two to the left.
MSE = (((0-0)**2) + ((0-0)**2) + ((0-40)**2) + ((80-40)**2))
= 800

MSE has dropped from 1200 to 800. We attribute 400 to feature Fever.

*https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

Consistency

Imagine that we have 4 observations and that they all finish in the correct leaf. We use Mean Squared Error as a metric.

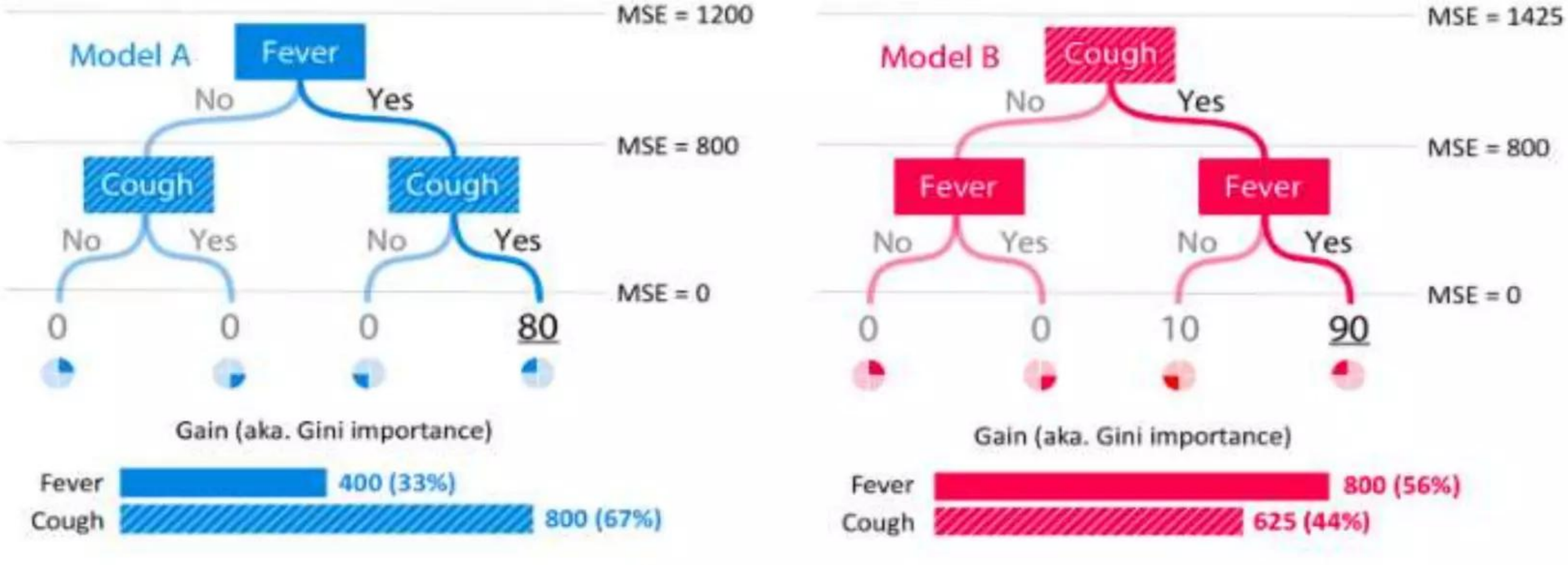


Step 3: We introduce the feature 'Cough' and we finally assign each observation to the correct leaf.
MSE = (((0-0)**2) + ((0-0)**2) + ((0-0)**2) + ((80-80)**2)) = 0

MSE has dropped from 800 to 0. We attribute 800 to feature Cough.

*https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

Consistency, where is the problem?



Computation of the gain (aka. Gini importance) scores for model A and model B.

Features **near the root of tree should be more important**, for the greedy way trees are built. When Cough is promoted to a upper level in model B importance actually decreases! Hence the **inconsistency** in the method.

*https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

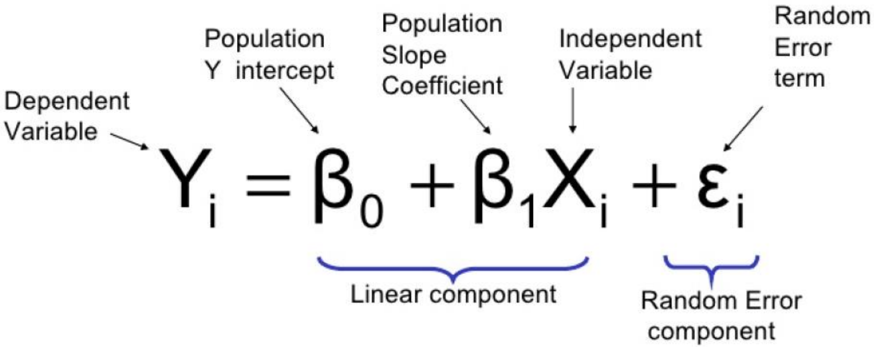
And more …

- Interactive feedback - can model learn from human actions in online setting ? (Can you tell a model to not repeat a specific mistake ?)

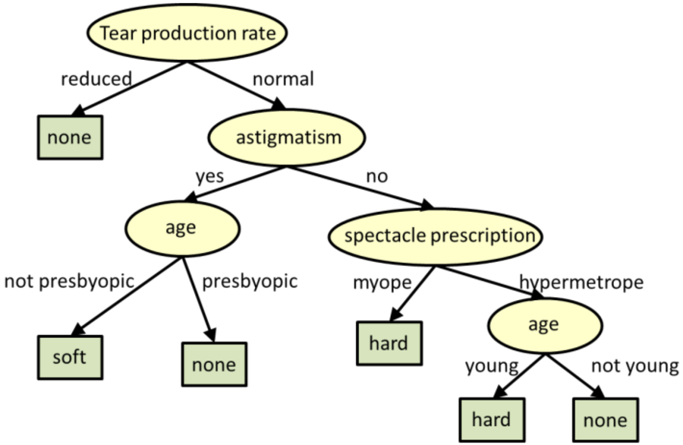
- Recourse – Can a model tell us what actions we can take to change its output ? (For example, what can you do to improve your credit score?)

What does interpretation looks like ?

- In pre-deep learning models, some models are considered “interpretable”



$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$



What does interpretation look like ?

- Heatmap Visualization

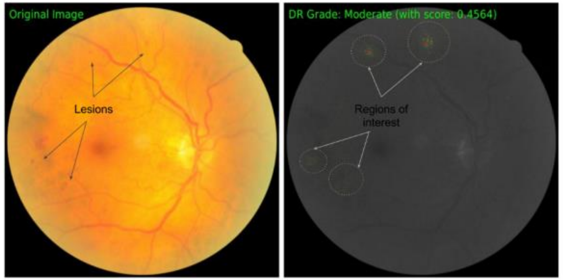
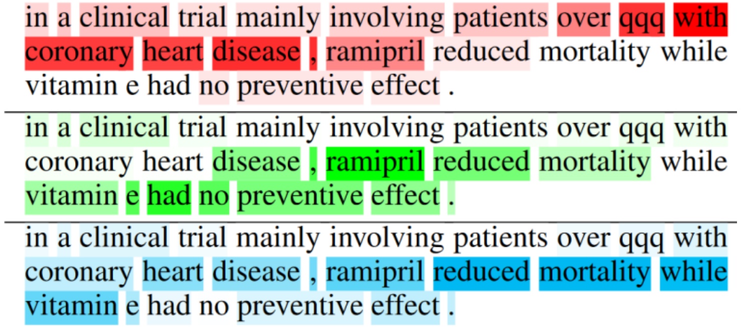


Figure 3. **Attribution for Diabetic Retinopathy grade prediction from a retinal fundus image.** The original image is show on the left, and the attributions (overlayed on the original image in gray scaee) is shown on the right. On the original image we annotate lesions visible to a human, and confirm that the attributions indeed point to them.

[Sundarajan 2017]



in a clinical trial mainly involving patients over qqq with coronary heart disease , ramipril reduced mortality while vitamin e had no preventive effect .

in a clinical trial mainly involving patients over qqq with coronary heart disease , ramipril reduced mortality while vitamin e had no preventive effect .

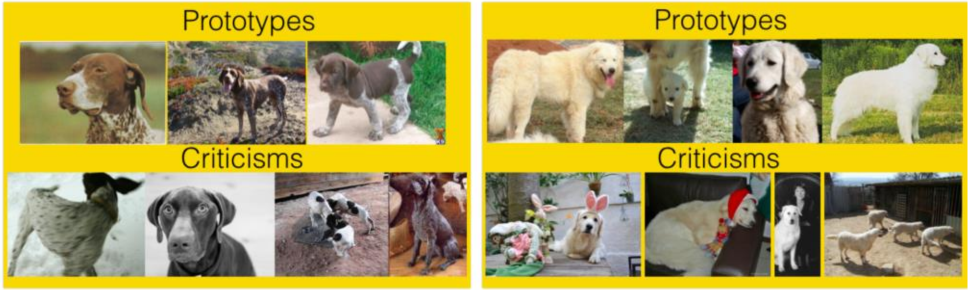
in a clinical trial mainly involving patients over qqq with coronary heart disease , ramipril reduced mortality while vitamin e had no preventive effect .

Table 2: Gate activations for each aspect in a PIC abstract. Note that because gates are calculated a the final convolution layer, activations are not i exact 1-1 correspondence with words.

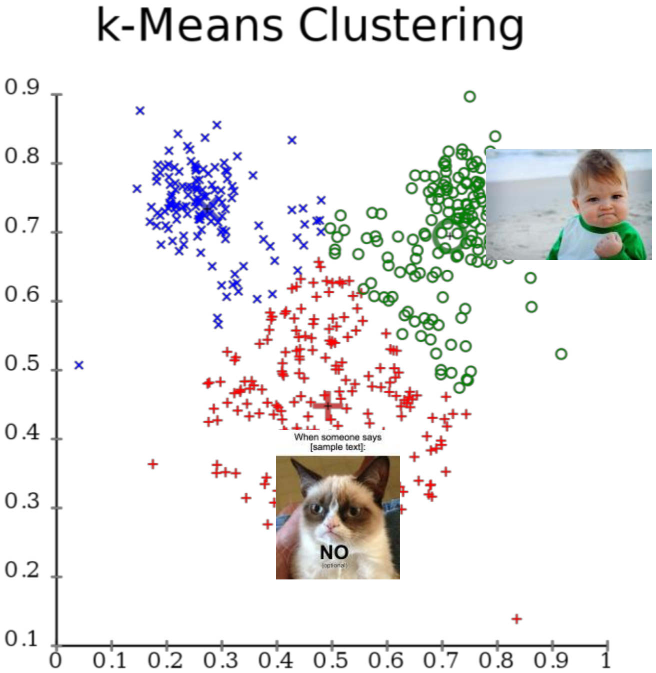
[Jain 2018]

What does interpretation looks like ?

- Give prototypical examples



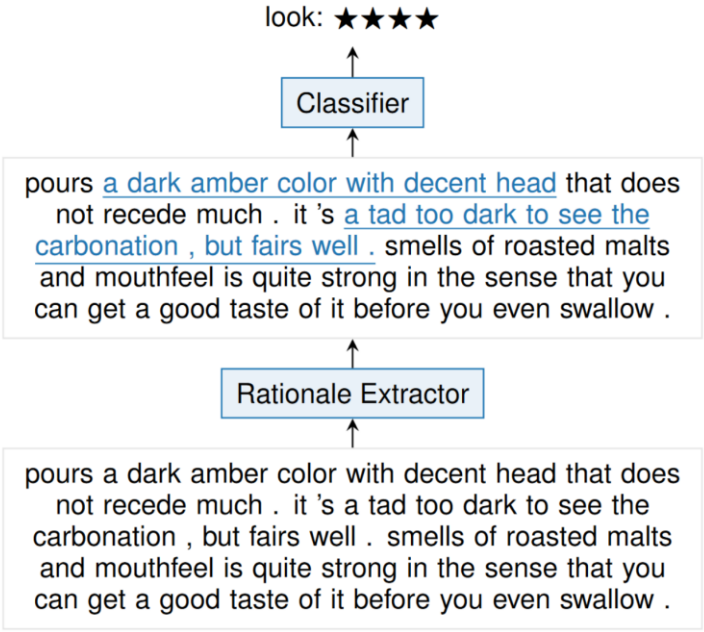
[Kim 2016]



By Chire - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=11765684

What does interpretation look like ?

- Bake it into the model



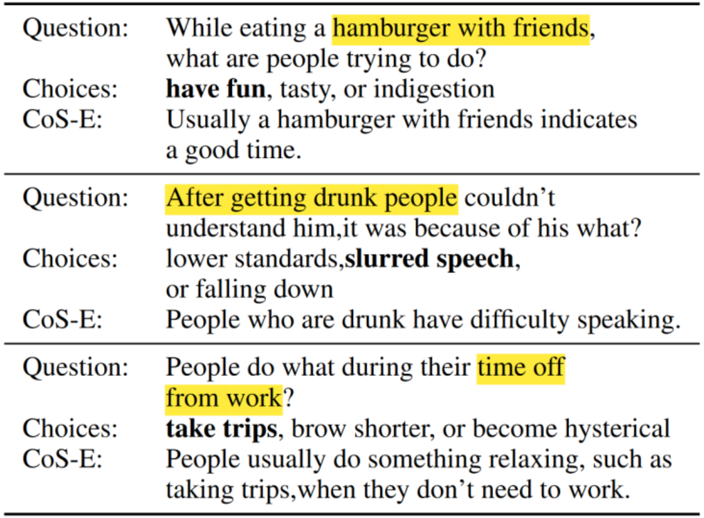
pours a dark amber color with decent head that does not recede much . it 's a tad too dark to see the carbonation , but fairs well . smells of roasted malts and mouthfeel is quite strong in the sense that you can get a good taste of it before you even swallow .

pours a dark amber color with decent head that does not recede much . it 's a tad too dark to see the carbonation , but fairs well . smells of roasted malts and mouthfeel is quite strong in the sense that you can get a good taste of it before you even swallow .

[Bastings et al 2019]

What does interpretation looks like ?

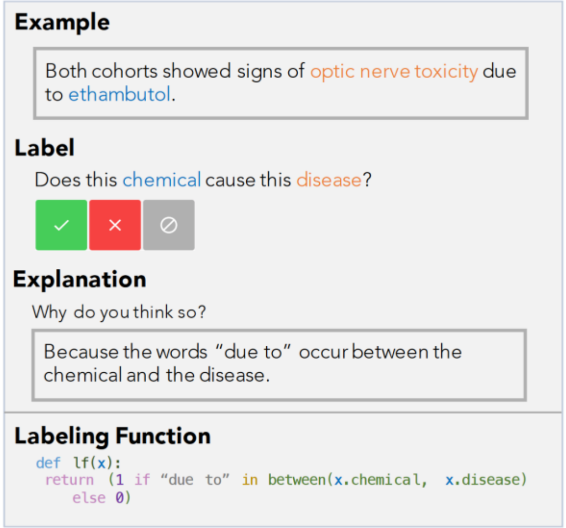
- Provide explanation as text



Question:	While eating a hamburger with friends, what are people trying to do?
Choices:	**have fun**, tasty, or indigestion
CoS-E:	Usually a hamburger with friends indicates a good time.
Question:	After getting drunk people couldn't understand him,it was because of his what?
Choices:	lower standards,**slurred speech**, or falling down
CoS-E:	People who are drunk have difficulty speaking.
Question:	People do what during their time off from work?
Choices:	**take trips**, brow shorter, or become hysterical
CoS-E:	People usually do something relaxing, such as taking trips,when they don't need to work.

Table 1: Examples from our CoS-E dataset.

[Rajani et al 2019]



Example

Both cohorts showed signs of optic nerve toxicity due to ethambutol.

Label

Does this chemical cause this disease?

✓ ✗ ⊘

Explanation

Why do you think so?

Because the words "due to" occur between the chemical and the disease.

Labeling Function

```
def lf(x):
  return (1 if "due to" in between(x.chemical, x.disease)
     else 0)
```

Figure 1: In BabbleLabble, the user provides a natural language explanation for each labeling decision. These explanations are parsed into labeling functions that convert unlabeled data into a large labeled dataset for training a classifier.

[Hancock et al 2018]

Evaluating Interpretability [Doshi-Velez 2017]

- Application level evaluation – Put the model in practice and have the end users interact with explanations to see if they are useful .
- Human evaluation – Set up a Mechanical Turk task and ask non-experts to judge the explanations
- Functional evaluation – Design metrics that directly test properties of your explanation.

How to “interpret” ? Some definitions

Global vs Local

- **Do we explain individual prediction ?**

Example –

Heatmaps
Rationales

- **Do we explain entire model ?**

Example –

Prototypes
Linear Regression
Decision Trees

Inherent vs Post-hoc

- **Is the explainability built into the model ?**

Example –

Rationales
Linear Regression
Decision Trees
Natural Language Explanations

- **Is the model black-box and we use external method to try to understand it ?**

Example –

Heatmaps (Some forms)
Prototypes

Model based vs Model Agnostic

- **Can it explain only few classes of models ?**

Example –

Rationales
LR / Decision Trees
Attention
Gradients (Differentiable
Models only)

- **Can it explain any model ?**

Example –

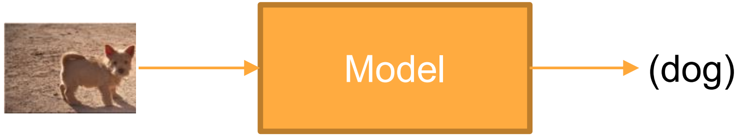
LIME – Locally Interpretable
Model Agnostic Explanations

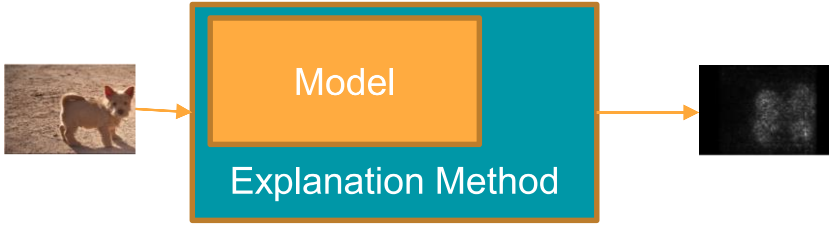
SHAP – Shapley Values

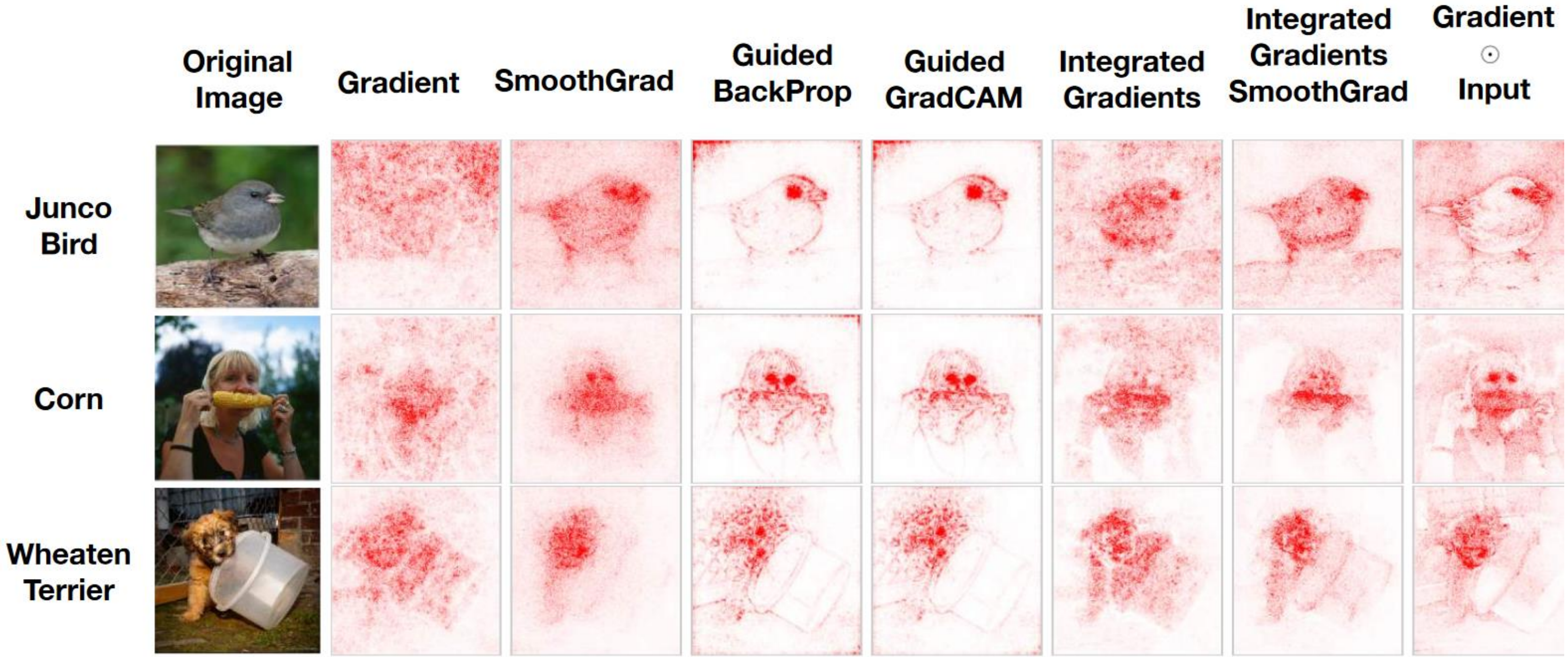
Some
Locally Interpretable,
Post-hoc
methods

Saliency Based Methods

- Heatmap based visualization
- Need differentiable model in most cases
- Normally involve gradient







[Adebayo et al 2018]

Saliency Example - Gradients

$$f(x): R^d \to R$$

$$E(f)(x) = \frac{df(x)}{dx}$$

How do we take gradient with respect to words ?

Take gradient with respect to embedding of the word .

Saliency Example – Leave-one-out

$$f(x)\colon R^{d} \to R$$

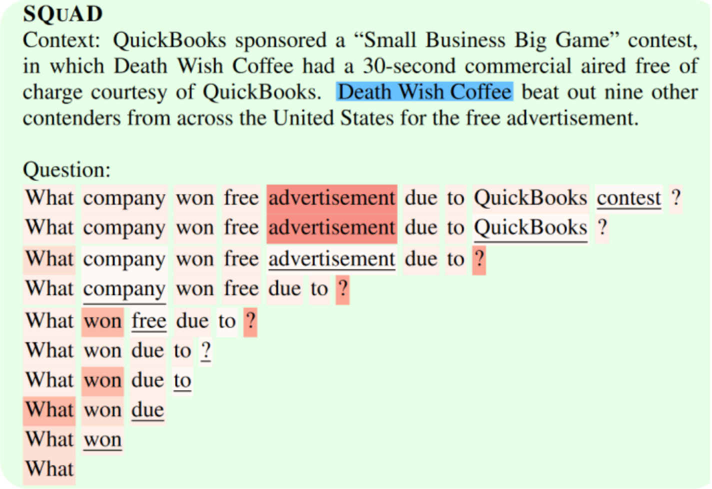
$$E(f)(x)_i = f(x) - f(x\backslash \mathrm{i})$$

How to remove ?

1. Zero out pixels in image
2. Remove word from the text
3. Replace the value with population mean in tabular data

Problems with Saliency Maps

- Only capture first order information
- Strange things can happen to heatmaps in second order.



SQUAD

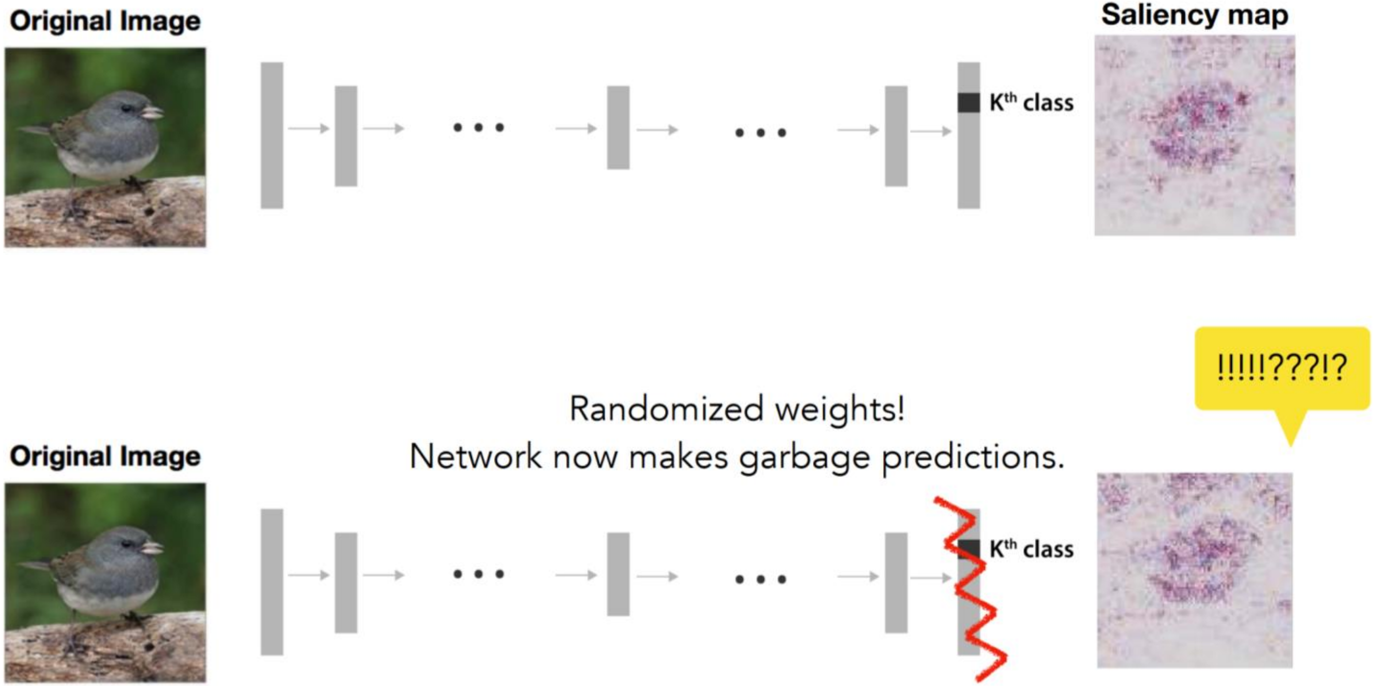
Context: QuickBooks sponsored a “Small Business Big Game” contest, in which Death Wish Coffee had a 30-second commercial aired free of charge courtesy of QuickBooks. Death Wish Coffee beat out nine other contenders from across the United States for the free advertisement.

Question:
What company won free advertisement due to QuickBooks contest ?
What company won free advertisement due to QuickBooks ?
What company won free advertisement due to ?
What company won free due to ?
What won free due to ?
What won due to ?
What won due to
What won due
What won
What

Figure 6: Heatmap generated with leave-one-out shifts drastically despite only removing the least important word (underlined) at each step. For instance, “advertisement”, is the most important word in step two but becomes the least important in step three.

[Feng et al 2018]

Sanity check:
When prediction changes, do explanations change?



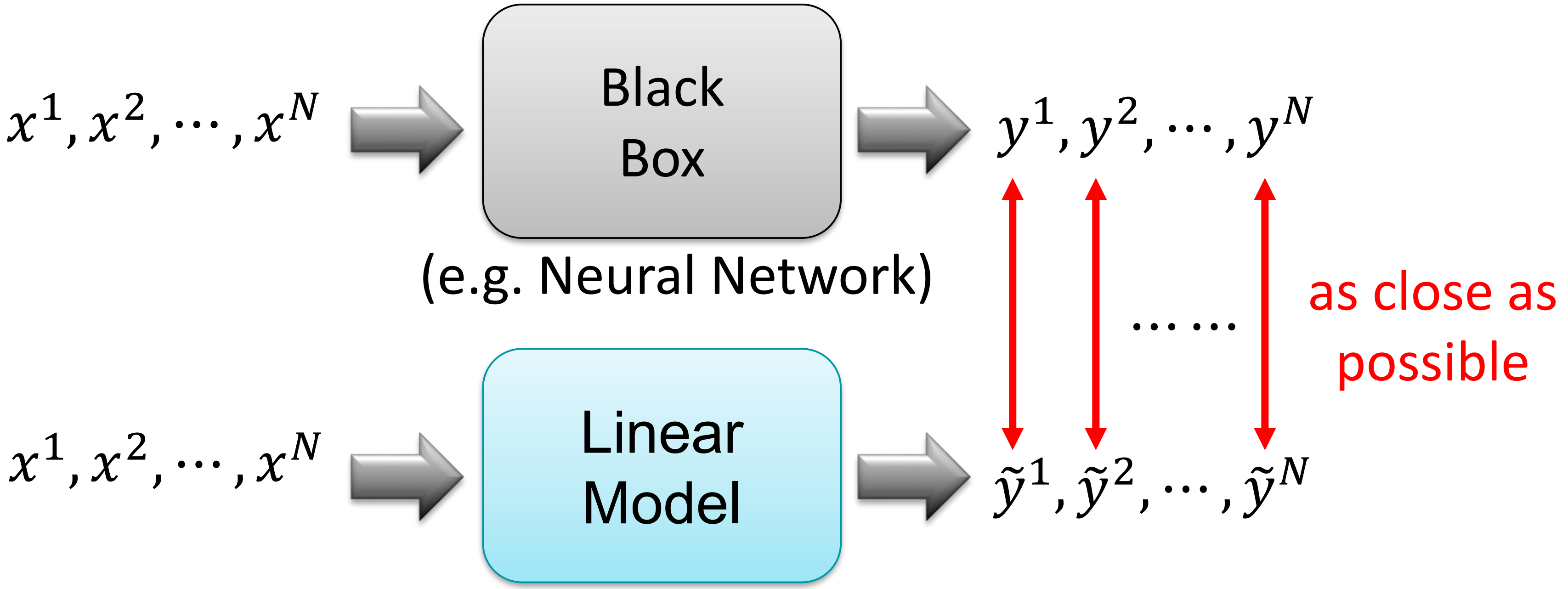
Randomized weights!
Network now makes garbage predictions.

(Slide Credit – Julius Adebayo)

Method: LIME

(Image Credit – Hung-yi Lee)

LIME – locally interpretable model agnostic



Can’t do it globally of course, but locally ? Main Idea behind LIME

Intuition behind LIME

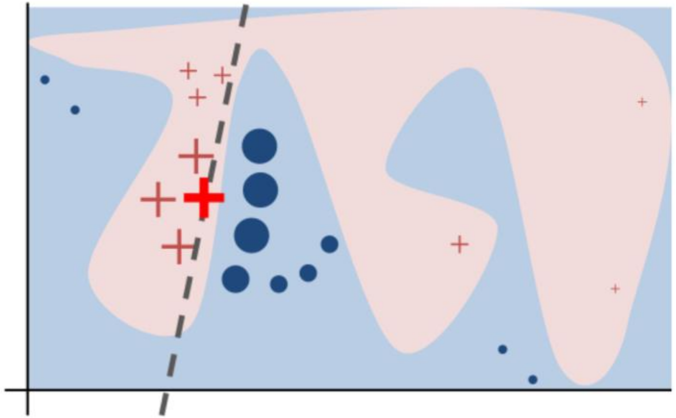
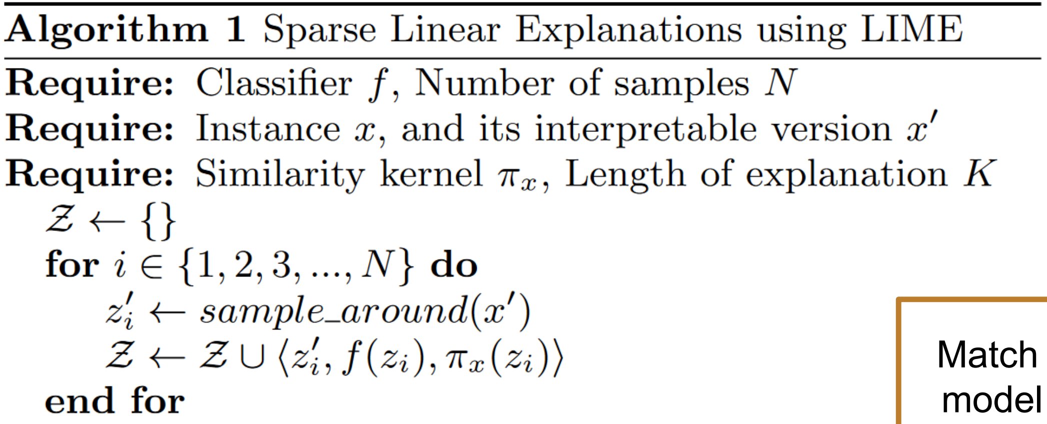


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

[Ribeiro et al 2016]

The Math behind LIME



Algorithm 1 Sparse Linear Explanations using LIME

```
Require: Classifier $f$, Number of samples $N$
Require: Instance $x$, and its interpretable version $x'$
Require: Similarity kernel $\pi_x$, Length of explanation $K$
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, 3, ..., N\}$ do
    $z'_i \leftarrow sample\_around(x')$
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
  end for
```

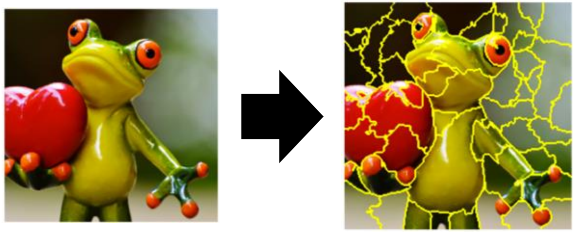
Match interpretable model to black box

Control complexity of the model

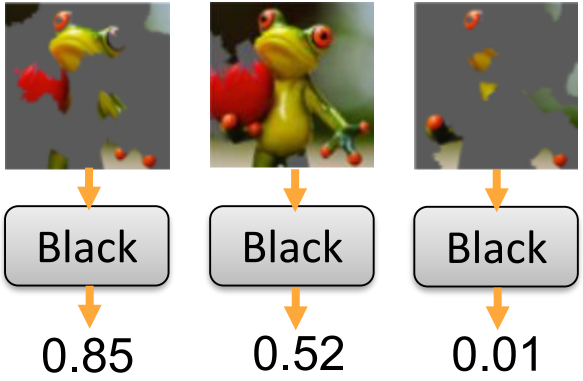
$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \; \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

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

LIME – Image



- 1. Given a data point you want to explain
- 2. Sample at the nearby - Each image is represented as a set of superpixels (segments).



Black → 0.85

Black → 0.52

Black → 0.01

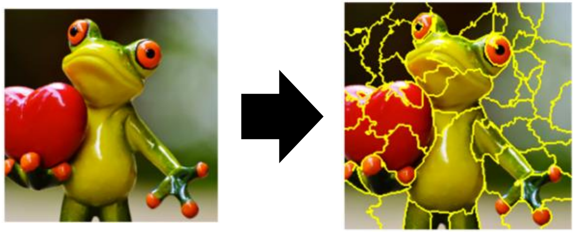
Randomly delete some segments.

Compute the probability of “frog” by black box

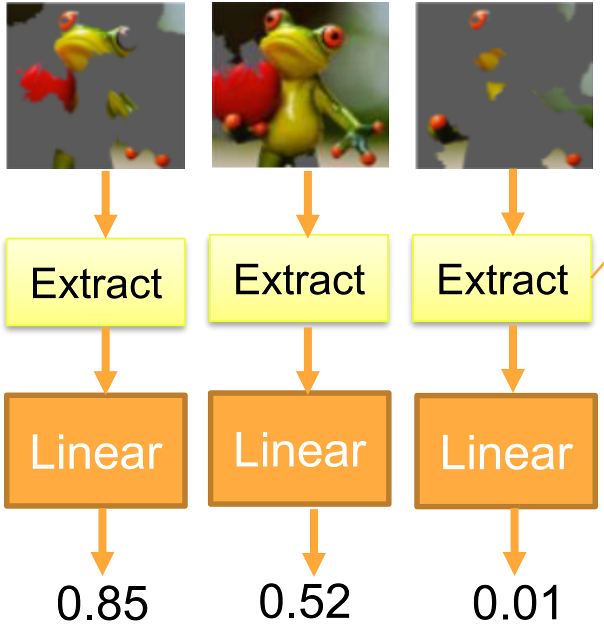
Ref: https://medium.com/@kstseng/lime-local-interpretable-model-agnostic-explanation%E6%8A%80%E8%A1%93%E4%BB%8B%E7%B4%B9-a67b6c34c3f8

(Slide Credit – Hung-yi Lee)

LIME – Image



- 3. Fit with linear (or interpretable) model



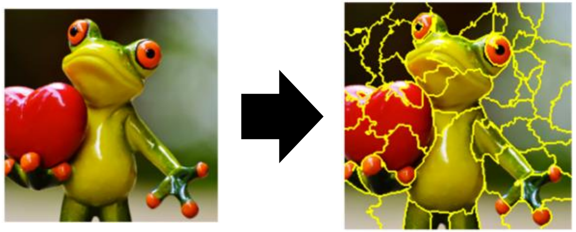
$x_1 \quad \cdots\cdots \quad x_m \quad \cdots\cdots \quad x_M$

$$x_m = \begin{cases} 0 & \text{Segment m is deleted.} \\ 1 & \text{Segment m exists.} \end{cases}$$

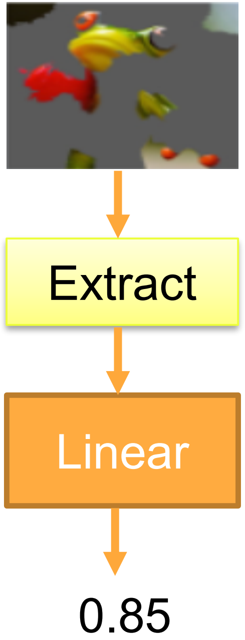
M is the number of segments.

(Slide Credit – Hung-yi Lee)

LIME – Image



- 4. Interpret the model you learned



$$y = w_1 x_1 + \cdots + w_m x_m + \cdots + w_M x_M$$

$$x_m = \begin{cases} 0 & \text{Segment m is deleted.} \\ 1 & \text{Segment m exists.} \end{cases}$$

M is the number of segments.

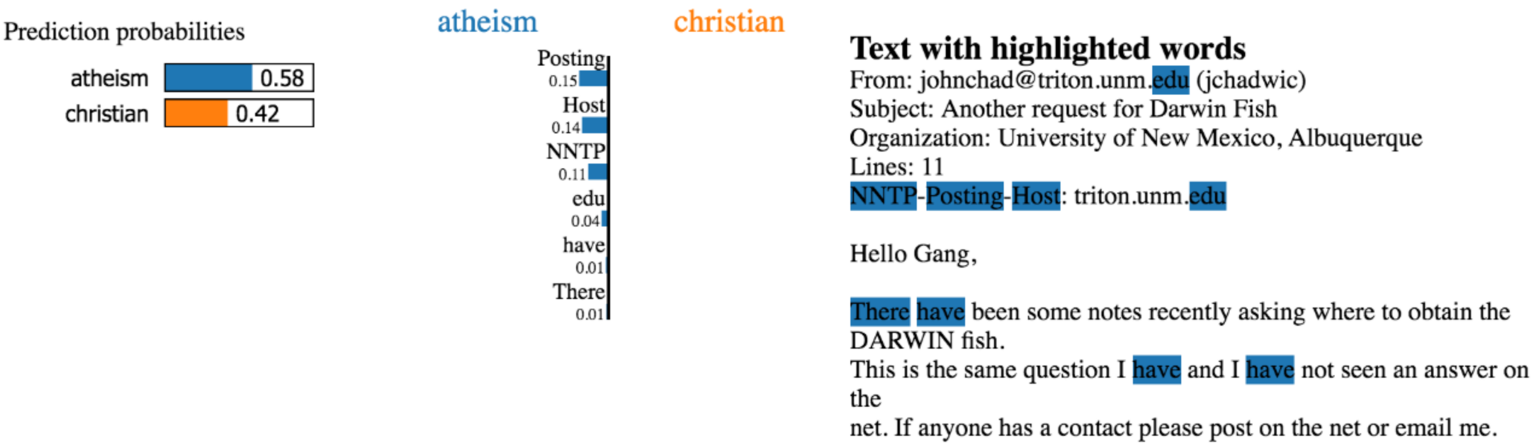
If $w_m \approx 0$ → segment m is not related to “frog”

If w_m is positive → segment m indicates the image is “frog”

If w_m is negative → segment m indicates the image is not “frog”

(Slide Credit – Hung-yi Lee)

Example from NLP



Text with highlighted words

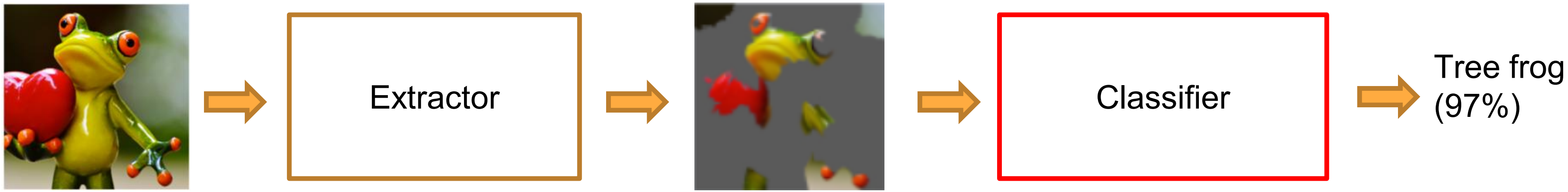
From: johnchad@triton.unm.edu (jchadwic)
Subject: Another request for Darwin Fish
Organization: University of New Mexico, Albuquerque
Lines: 11
NNTP-Posting-Host: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the
DARWIN fish.
This is the same question I have and I have not seen an answer on
the
net. If anyone has a contact please post on the net or email me.

Rationalization Models

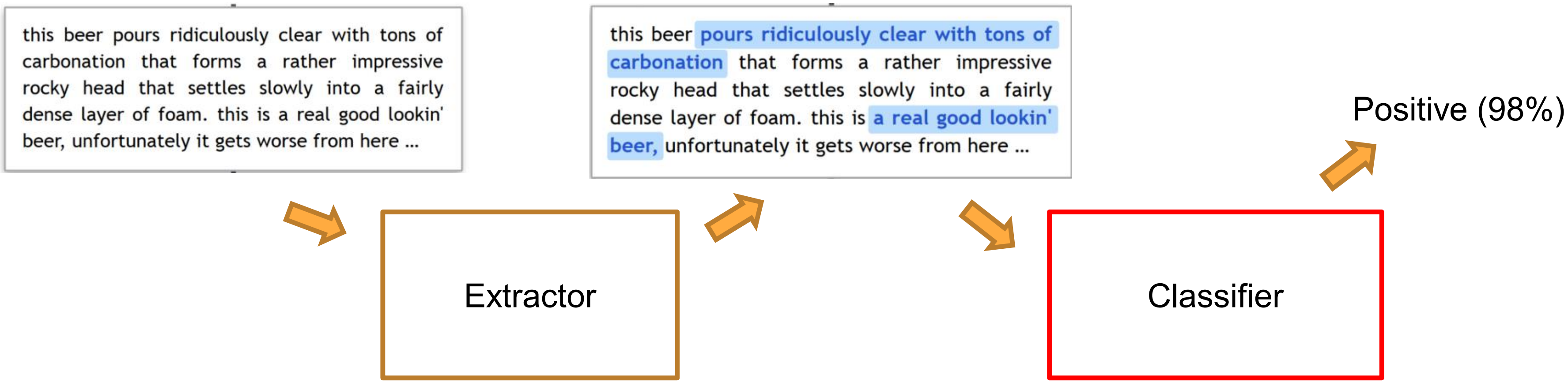
General Idea



Extractor

Classifier

Tree frog (97%)



this beer pours ridiculously clear with tons of carbonation that forms a rather impressive rocky head that settles slowly into a fairly dense layer of foam. this is a real good lookin' beer, unfortunately it gets worse from here ...

Extractor

this beer **pours ridiculously clear with tons of carbonation** that forms a rather impressive rocky head that settles slowly into a fairly dense layer of foam. this is **a real good lookin' beer,** unfortunately it gets worse from here ...

Classifier

Positive (98%)

Method: SHAP

SHAP? What is it?

- SHAP stands for **Sh**apley **A**dditive Ex**p**lanations. It’s a model-agnostic, efficient algorithm, to compute features contribution to a model output.
- With non linear black box models SHAP provides **accurate and consistent** features importance values.
- It allows meaningful, **local explanations** of individual predictions.
- SHAP borrows concepts from cooperative **game theory**: The Shapley Values

It was developed by Scott Lundberg and Su-In Lee from University of Washington (WA)*

https://arxiv.org/pdf/1705.07874.pdf

Shapley Values

- Shapley values are a concept in **cooperative game theory**. They where introduced in 1953 by the Nobel Prize winner Lloyd Shapley, one of the fathers of Game Theory*.

- The overall intuition behind the concept is that sometimes a player value in a team could be **greater** than their value if they were on their own.

- In a Machine Learning setting a Shapley value is “the **contribution** of a feature value to difference between the actual prediction and the mean prediction”…

- …which is equivalent to answer this question: “Given that without any features we would just predict an **average** value, once we bring the first feature in how much our prediction changes compared to the average?”

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

Let’s start with the Math

1) Given a set **N** of players *I,* each of which can be attributed a value $N = \{1, 2, 3\},$

2) We calculate a set of permutations R of N.

3) We then calculate the marginal contribution given by that feature in the following way:

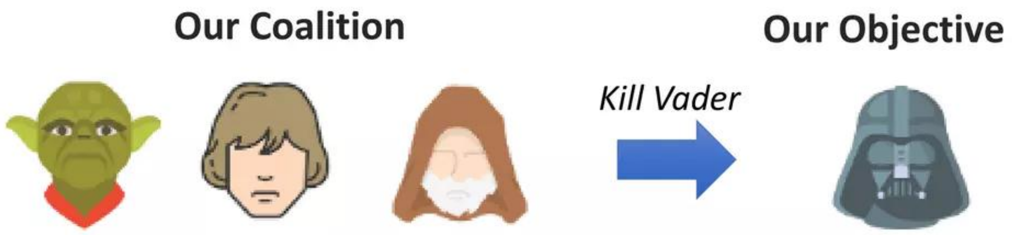
$$\varphi_i(v) = \frac{1}{|N|!} \sum_R \left[\underset{\downarrow}{\boxed{v(P_i^R \cup \{i\})}} - \underset{\downarrow}{\boxed{v(P_i^R)}} \right] ,$$

Set of features preceding *i* in order R, including *i*

Set of features preceding and excluding *i*

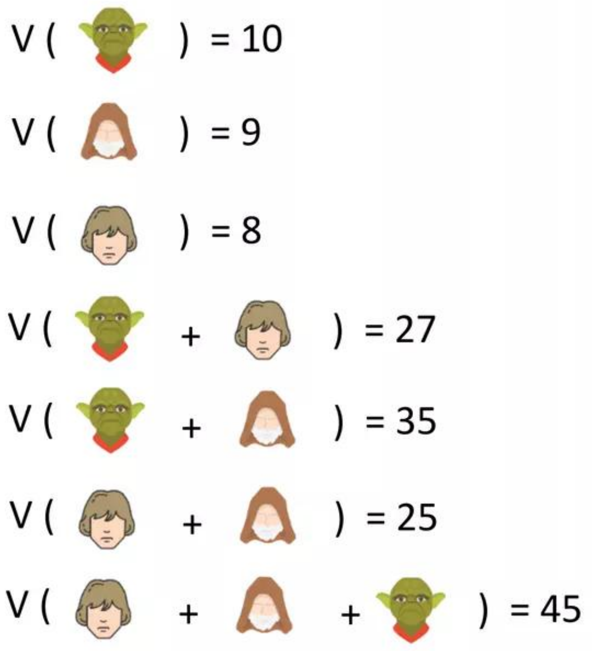
4) Where R is an ordering, given by permuting the values in set N, and P_i^R is the set of a players preceding *i* in the order R.

Some friends may help explaining this…



Algorithm

i. Calculate all possible coalitions permutations.
ii. For each permutation take the set of players preceding our target Jedi.
iii. Include the target Jedi in this subset
iv. Then subtract the contribution of the subset excluding the target Jedi



Order R	Yoda Contribution*	Obi Contribution*	Luke Contribution*
Y, O, L	V(Y) = 10	V(O, Y) – V(O) = 35 – 9 = 26	V(L, O, Y) – V(O, Y) = 45 – 35 = 10
Y, L, O	V(Y) = 10	V(O, L, Y) – V(L, Y) = 45 – 27 = 18	V(L, Y) – V(Y) = 27 – 10 = 17
O, Y, L	V(Y, O) – V(O) = 35-9 = 26	V(O) = 9	V(L, O, Y) – V(O, Y) = 45 – 35 = 10
O, L, Y	V(Y, L, O) – V(L, O) = 45 – 25 = 20	V(O) = 9	V(L, O) – V(O) = 25 – 9 = 16
L, Y, O	V(L,Y) – V(L) = 27 – 8 =19	V(O, L, Y) – V(L, Y) = 45 – 27 = 18	V(L) = 8
L, O, Y	V(Y, L, O) – V(L, O) = 45 – 25 = 20	V(O, L) – V(L) = 25 – 8 = 17	V(L) = 8

* Marginal Contributions

Now we can calculate the payout for each Jedi

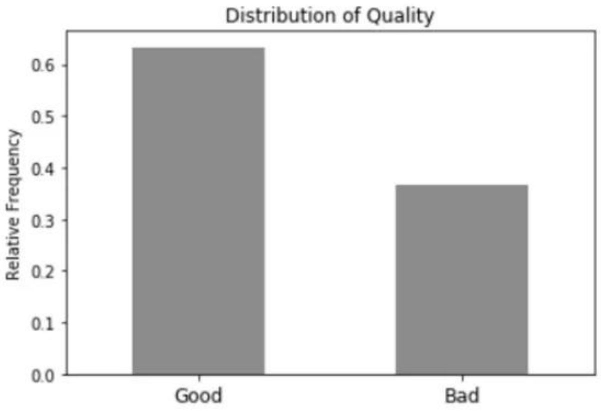
	Initial Value	Payout (SHAP Value)
	10	10 + 10 + 26 + 20 + 19 + 20 = **17,5**
	9	26 + 18 + 9 + 9 + 18 + 17 = **16,2**
	8	10 + 17 + 10 + 16 + 8 + 8 = **11.5**

So what? …After calculating each player marginal contributions* we realize that although Luke is 20% weaker the **contributed** 34% less than Yoda. Obi in terms of contribution is much closer to Yoda!

*"The Shapley value can be misinterpreted. The Shapley value of a feature value is not the difference of the predicted value after removing the feature from the model training. The interpretation of the Shapley value is: Given the current set of feature values, the contribution of a feature value to the difference between the actual prediction and the mean prediction is the estimated Shapley value" (https://christophm.github.io/interpretable-ml-book/shapley.html#general-idea)

Dataset

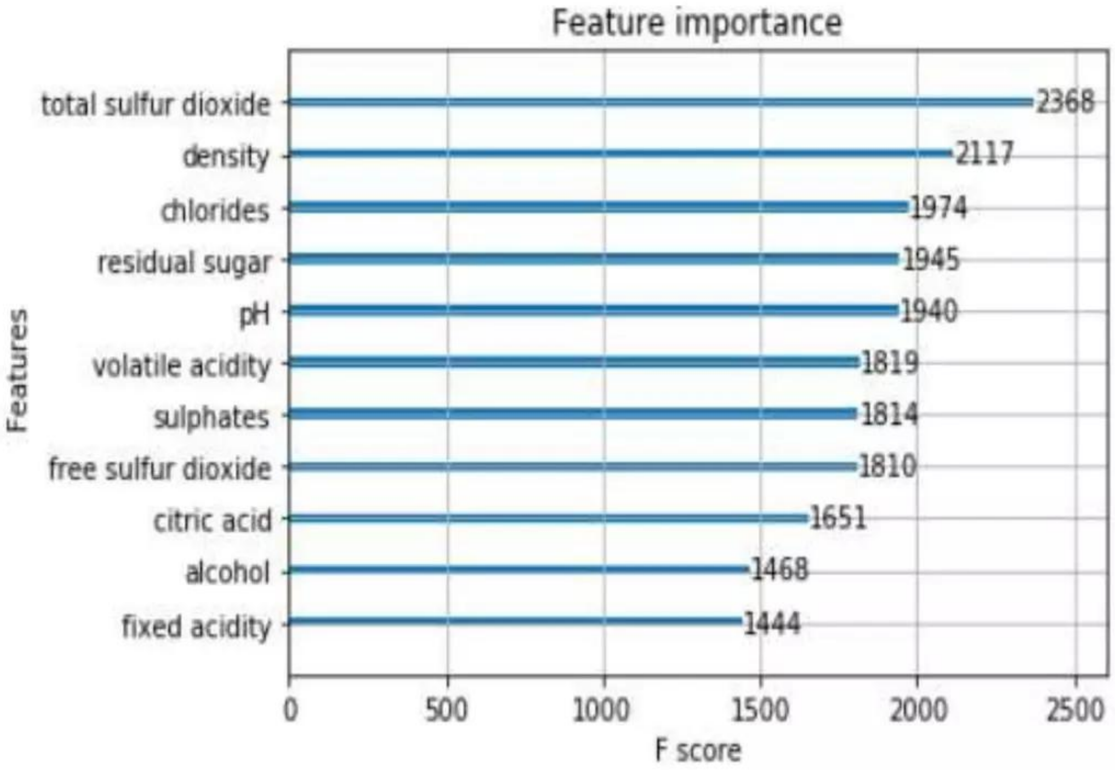
- I have used a Wine Quality* dataset for this talk.
- 12 features for 6.5k observations of Portuguese Vinho Verde from several different producers.
- For each row we have a **quality score** from 1 to 10.
- We have converted the problem to a **binary classification** exercise where 1 is a score is quality from 6 to 10, whilst 0 is quality from 0 to 5 (included)



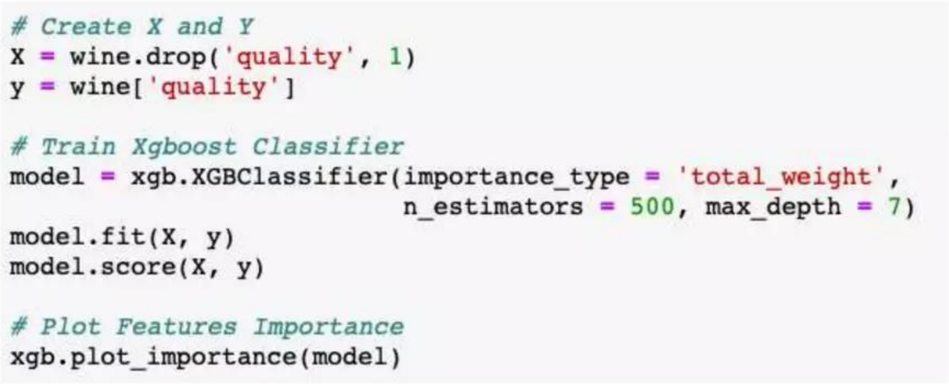
Vinho verde is a unique product from the Minho (northwest) region of **Portugal**. Medium in alcohol, is it particularly appreciated due to its freshness (specially in the summer). More details can be found at: http://www.vinhoverde.pt/en/

*P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In *Decision Support Systems*, Elsevier, 47(4):547-553, 2009.

Model Feature Importance



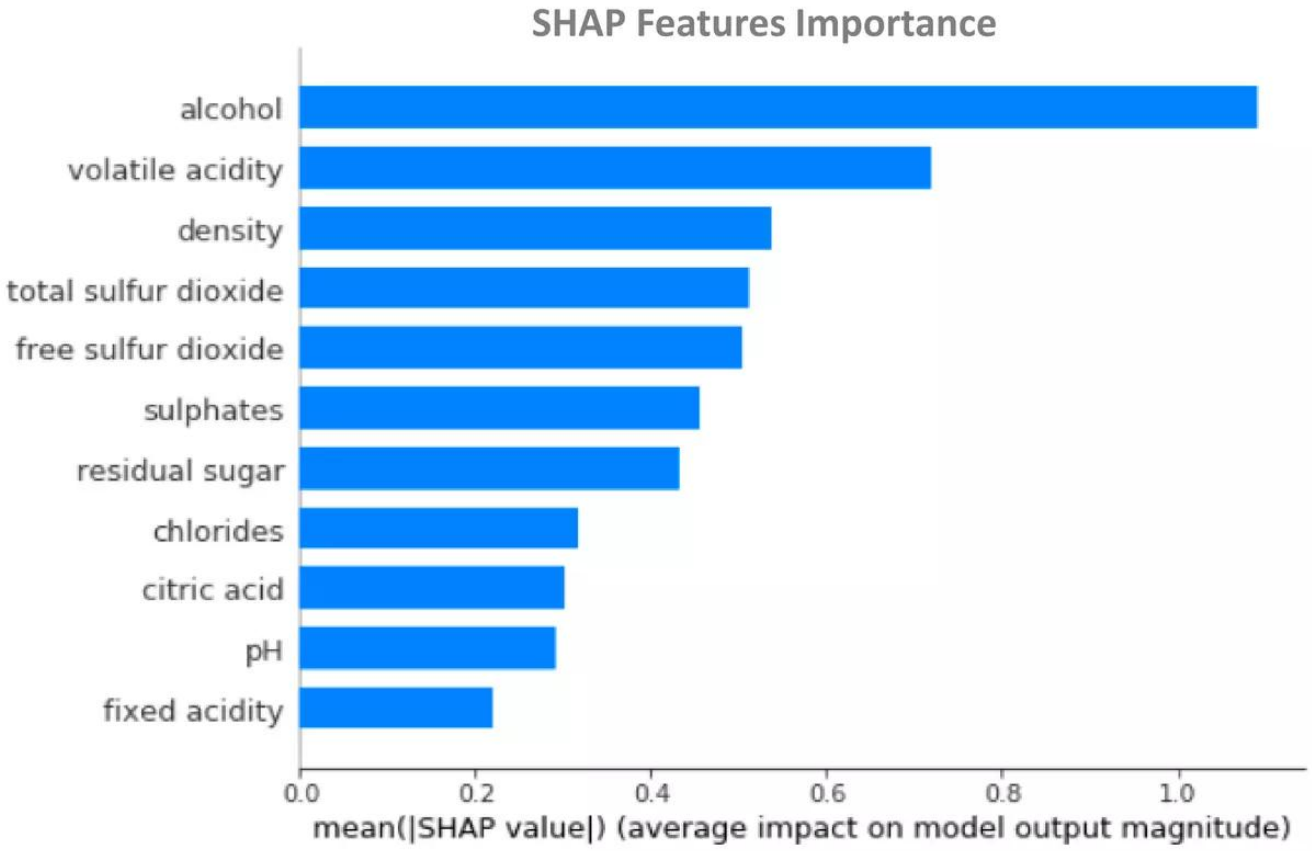
Example Code



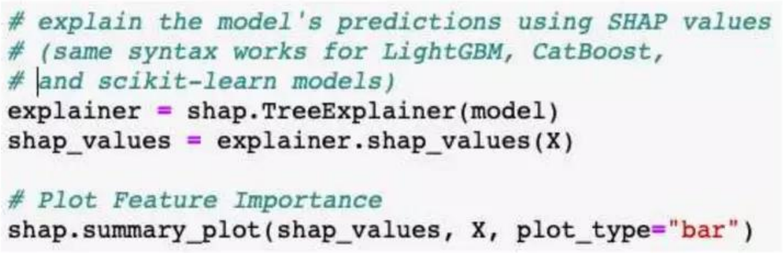
```
# Create X and Y
X = wine.drop('quality', 1)
y = wine['quality']

# Train Xgboost Classifier
model = xgb.XGBClassifier(importance_type = 'total_weight',
                          n_estimators = 500, max_depth = 7)
model.fit(X, y)
model.score(X, y)

# Plot Features Importance
xgb.plot_importance(model)
```

- We used Xgboost to train a classifier for this dataset.
- We get feature importance at a global level, but **insightfulness** is quite low.
- **We see that 'Total Sulfur Dioxide' is the most important feature, but how can we tell whether it tends to trigger a 0 or a 1?**

SHAP Features Importance



Example Code



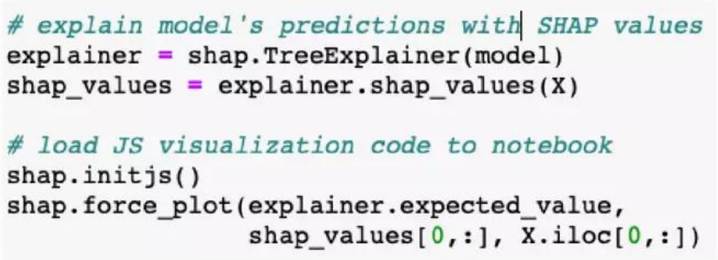
```
# explain the model's predictions using SHAP values
# (same syntax works for LightGBM, CatBoost,
# and scikit-learn models)
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X)

# Plot Feature Importance
shap.summary_plot(shap_values, X, plot_type="bar")
```

- SHAP features are built **averaging** the feature contribution for each row in the dataset.
- They look completely **different** that Xgboost feature importance! Actually they are the other way around, why?
- **Tree attribution methods** give more value to features far away from the root, but this is counterintuitive.

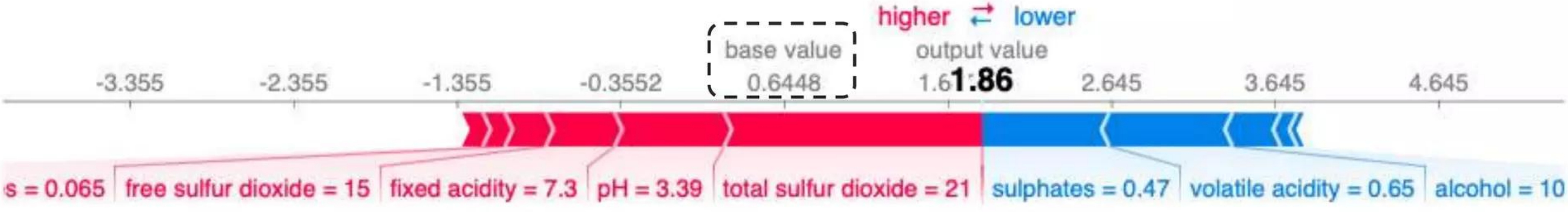
SHAP Local Explanations

- With SHAP we are able to get **local explanations** by using the Force plots.
- Those tell us how much each feature contributed to make the prediction diverge from a **base value**. This is the reference value that the feature contributions start from*.
- We can see that a low level of ‘total sulfur dioxide’ (mean is 30) pushes the output towards a positive prediction, while the level of sulphates makes it go in the opposite direction.

Example Code



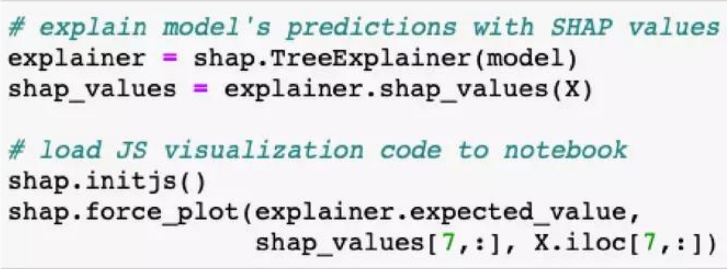
```
# explain model's predictions with SHAP values
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X)

# load JS visualization code to notebook
shap.initjs()
shap.force_plot(explainer.expected_value,
                shap_values[0,:], X.iloc[0,:])
```



Force plot doc string:https://github.com/slundberg/shap/blob/master/shap/plots/force.py

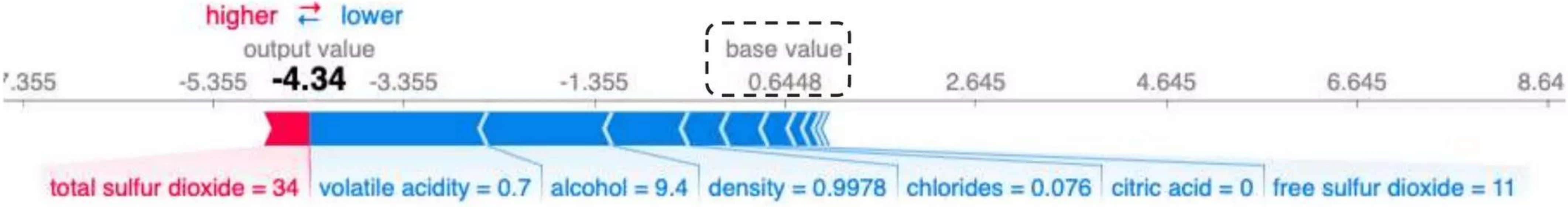
SHAP Local Explanations

Example Code



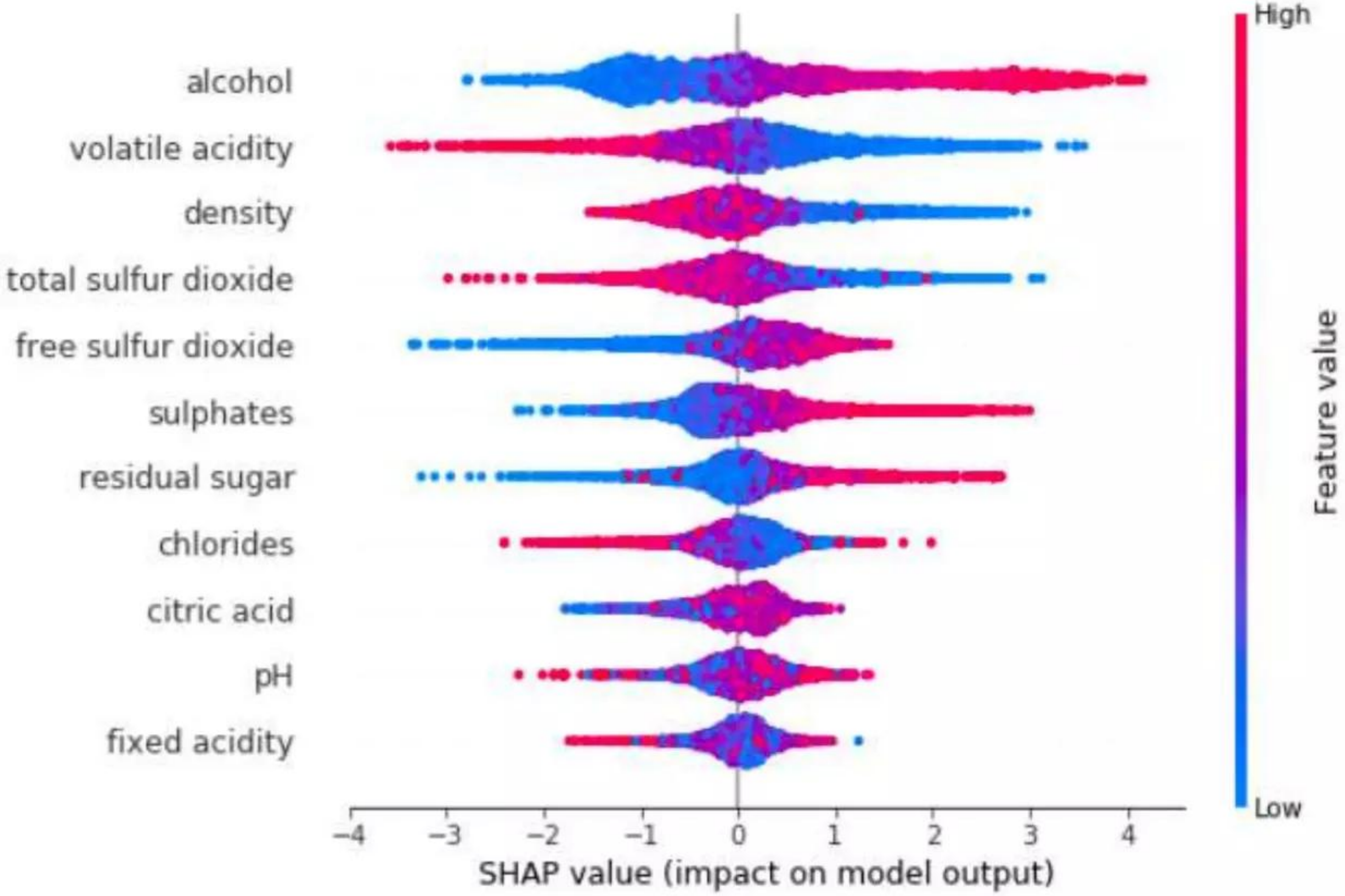
```
# explain model's predictions with SHAP values
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X)

# load JS visualization code to notebook
shap.initjs()
shap.force_plot(explainer.expected_value,
                shap_values[7,:], X.iloc[7,:])
```

- Here we have a **negative case**, with a total shap value much lower than the baseline
- Low level of alcohol, high volatile acidity, density and chlorides push the boundaries to a negative prediction.
- Only feature that pushes the score up is a decent level of total sulfur dioxide…I totally wouldn't want to drink this bottle.

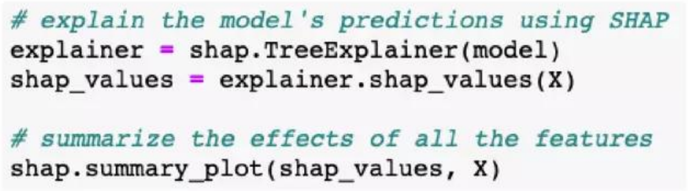


SHAP Summary Plots

SHAP Summary plot



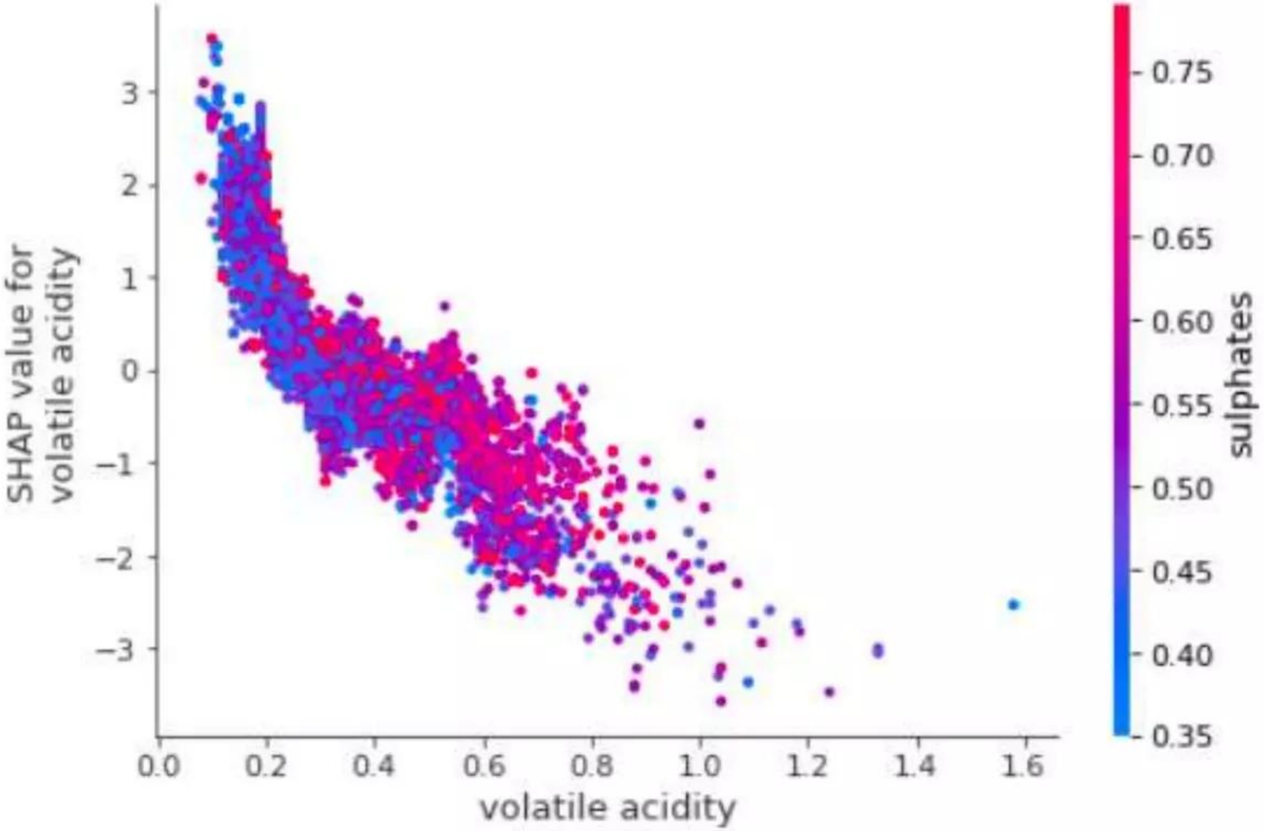
Example Code



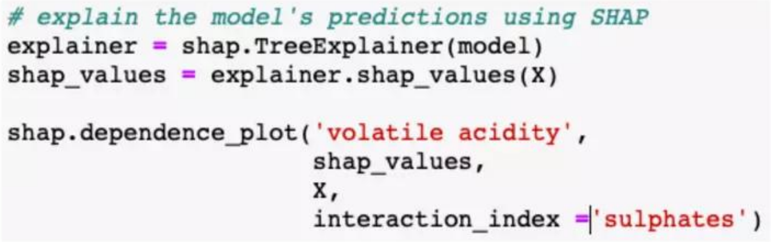
```
# explain the model's predictions using SHAP
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X)

# summarize the effects of all the features
shap.summary_plot(shap_values, X)
```

- Summary plots are powerful tools to gain **insights**. They summarize features contribution for all the rows.
- And to my experience they are easy to understand for **business people** (skilled ones) too!
- Here a high level of alcohol pushes predictions to a ‘High Quality’, whilst the opposite happens with low levels.
- Low volatile acidity means high quality, the opposite happen when acidity is high.

SHAP Partial Dependence Plots



Example Code



```
# explain the model's predictions using SHAP
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X)

shap.dependence_plot('volatile acidity',
                     shap_values,
                     X,
                     interaction_index ='sulphates')
```

- **Partial dependence plots let us visualize a feature shap values in relation to the actual values.** Can you see the non linear negative contribution increase when acidity increases?

- And more complex analysis can be made by adding up an interaction feature. Here we can see how high level of sulphates compensate for high level of acidity.

Important Points to take away

- Interpretability – no consistent definition

- When designing new system, ask your stakeholders what they want out of it .

- See if you can use inherently interpretable model .

- If not, what method can you use to interpret the black box ?

- Ask – does this method make sense ? Question Assumptions !!!

- Stress Test and Evaluate !