



Machine Learning

Explainable Artificial Intelligence

Karol Przystalski

April 6, 2022

Department of Information Technologies, Jagiellonian University

Agenda

1. Introduction
2. Use cases
3. Approaches and methods
4. Adversarial attacks
5. xAI methods
6. Tools

Introduction

What is xAI?

Explainable AI is about understanding how the model works and how to interpret each stage. It can be also known as Interpretable Artificial Intelligence or Responsible Artificial Intelligence. Explainable AI should cover the following terms:

- Understandability,
- Comprehensibility,
- Interpretability,
- Explainability,
- Transparency.

Understandability Also known as intelligibility. A model should be understandable by humans. It means how the models works without explaining the internal structure of it, the algorithm or architecture. We shouldn't know how the model process the data internally.

Comprehensibility It means that a method is able to represent its learned knowledge in a human understandable fashion. The way how it works should be done in a same way like a human expert would do it and should be interpretable in natural language. This term is related to the evaluation of model complexity.

xAI – Interpretability, Explainability, and Transparency

Interpretability It is defined as the ability to explain or to provide the meaning in understandable terms to a human.

Explainability It's about an interface between humans and a decision maker that is at the same time, both an accurate proxy of the decision maker and comprehensible to humans.

Transparency A model is considered to be transparent if by itself it is understandable. Since a model can feature different degree of understandability, transparent models are divided into three categories.

Levels of transparency

The transparency of a model can be divided into several levels, depending on how transparent the model is. We have three levels:

- Simulatability,
- Decomposability,
- Algorithmic transparency.

Simulatability and Decomposability

Simulatability We can easily simulate the process as a human. The model needs to be quite simple to be able to simulate it.

Decomposability It's about explainability of each part of the model. Like a complex model that consist of many layers, we should be able to explain the way how it works on each layer, including the way how the layer process the data, but also explain the input and the output of the layer.

Algorithmic transparency

Depends on the view, but in most cases it's about the users that understands the process of the model when producing the output (prediction). Linear model are transparent, because of the simplicity and can be easily followed by the user.

Why xAI?

There are many scenarios where XAI is very useful. Just to point out a few groups:

- domain experts,
- regulatory agencies,
- managers, executive board members,
- data scientists,
- users affected by model decisions.

Why xAI?

The domain experts or users of the model like doctors trust the model itself, gain scientific knowledge.

The regulatory agencies certify the model compliance with the legislation in force.

The managers assess regulatory compliance and understand the corporate AI applications.

The data scientists ensure and improve the product efficiency or develop new functionalities.

Each other user affected by the model decision want to understand the situation and verify fair decisions.

There are many goals of an XAI model. Not each goal is met by every method and each goal has a different target audience.

Trustworthiness Is important for domain experts and other users affected by the model.

Causality It is about the causality of the variables/features we provide.

Transferability It's about reusing the knowledge you get from the model and use it also in other problems/challenges.

Informativeness A great deal of information is needed to be able to relate the user's decision to the solution given by the model, and avoid falling in misconception pitfalls; that's why the models should give information about the problem being tackled.

Confidence A model should be robust and stable to assure confidence of the predictions.

Fairness The model should be fair and ethical; usually it's ensured by analyzing the method visually.

Accessibility Allowing the non-technical or non-expert users to deal with the model/method.

Interactivity Interaction with the users can be important in cases where the end user is very important and the impact of the input to the model can tweak the model.

Privacy awareness It's not hard to convince about the privacy of the data and the internal representation is important.

Ways to explain

There are many ways how we can explain how the model works. Just to mention a few methods:

- Text explanation – we are able to explain the model using text, symbols, formulas,
- Visual explanation – we visual the model behavior. It's an easy interpretable way for humans to understand the way how the model works,
- Local explanation – we take a subspace of the model and explain it in different way,
- Explanations by example – we take an example and go through the model,
- Explanations by simplification – if the model is a bit too complex, we may simplify it and explain the way how it works on the simplified model,
- Feature relevance explanation – explain the relevance of the features we have in our data set and the importance of each to the model outcome.

Types of insights required from XAI

Even though, there are several definitions of xAI, typically we consider three different types of insights, such a system should be able to give. They may be formed as the following questions:

1. What features are thought to be the most important for the model?
2. How did each feature in the data affect a particular prediction?
3. How does each feature affect the model's predictions in a big picture sense?

What can be achieved by such insights?

The possibility to explain the decision of Machine Learning systems allows to improve the data science projects in the following ways:

- debugging,
- improving the feature engineering process,
- directing future data collection,
- informing human decision-making,
- building trust,
- bias removal,
- eliminating overfitting.

Our Machine Learning models in any Data Science project won't be perfect from the very beginning. There are a lot of different issues which may happen. Some of them are related to the algorithms themselves (i.e. underfitting, overfitting), but some errors occur in the very beginning, when we prepare the data for the training.

- Data leakage,
- Leaky predictors,
- Leaky validation strategies.

Improving the feature engineering process

Feature engineering is an important step when it comes to data preparation. It is an ongoing process of creating some new variables from the already existing features. Finding out, which variables our model consider to be important, may allow us to look for some similar features in the future.

Sometimes, we cannot even have an intuition about the provided data and have to work on the anonymized feature names, due to some privacy reasons. The beliefs of our model may show us what we should pay most attention to.

Directing future data collections

If our Data Science project is done incrementally, there are some possibilities to include some more data sources. If we have already created a model which prefers some features over the rest, we may use that knowledge to find some similar data sources and direct the data collection process.

Sometimes the idea behind using some ML models is not only to make any kind of predictions, but to put some valuable insights to the human understanding of the owned data. If, for any reason, our process cannot be automated with algorithms, automated analysis may deliver some directions of how to interpret it, and what to pay most attention to.

That's especially important for the areas which may simply affect people's health and wealth. Applying AI may sound great, but increasing the trust to the usage of these methods may be one of the most important steps to convince the society for the further application of Data Science. As long as ML is thought to be a black-box mechanism, which cannot be understood at all, the applicability to various sectors is limited.

An existence of bias in the dataset may be hard to be found at the very beginning. This is where white-box Machine Learning methods may be used. If we have a possibility to check the impact of all the features on the final outcome of the model, then we can simply see there is a kind of bias that we wouldn't like to have. That's a straight way to a self-fulfilling prophecy, and should be spotted as early as possible.

Eliminating overfitting

Overfit model tends to work great for the known examples, but behaves surprisingly badly for any new data. As long as we are not able to spot such behavior, we may think we prepared a great solution.

Use cases



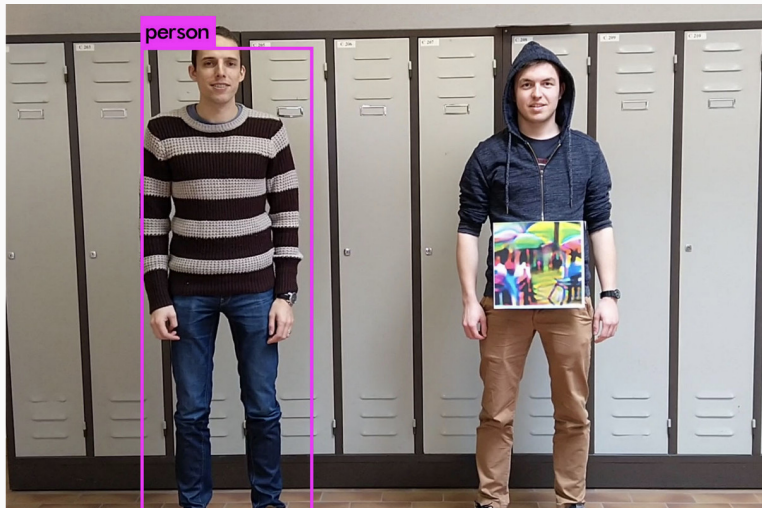
Source:

https://www.theregister.co.uk/2017/06/20/tesla_death_crash_accident_report_ntsb/



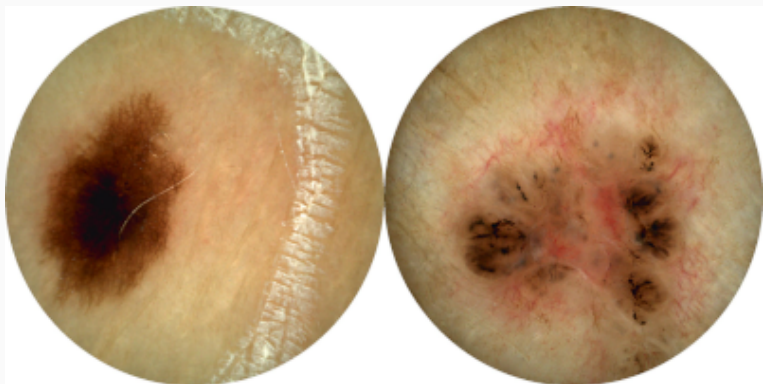
Source:

<https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist>



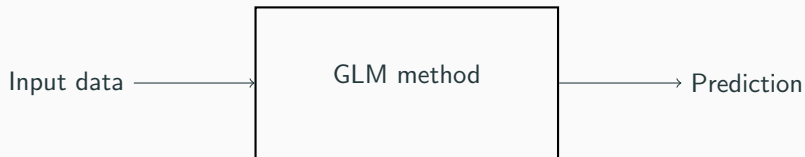
Adversarial attacks



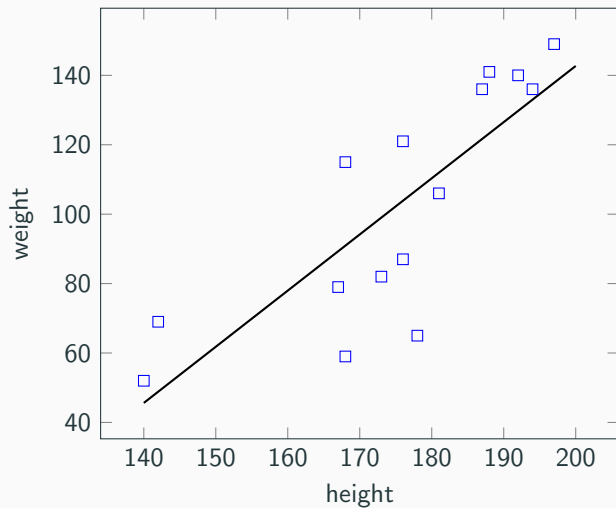


Approaches and methods

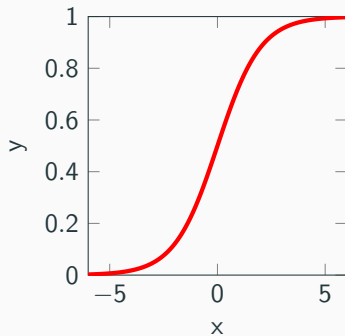
White-box methods



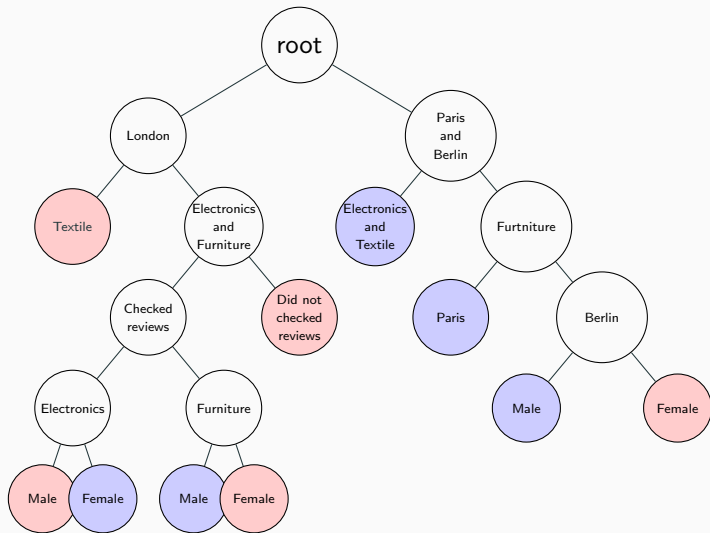
GLM – linear regression



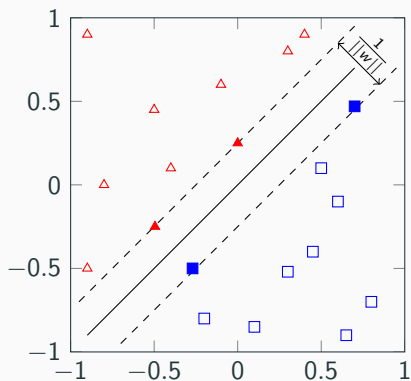
GLM – logistic regression



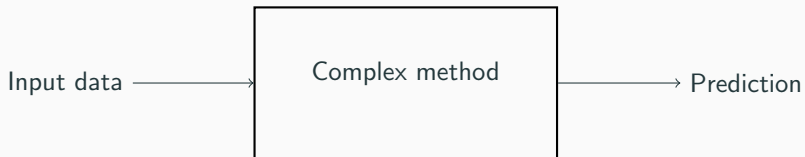
GLM – decision tree



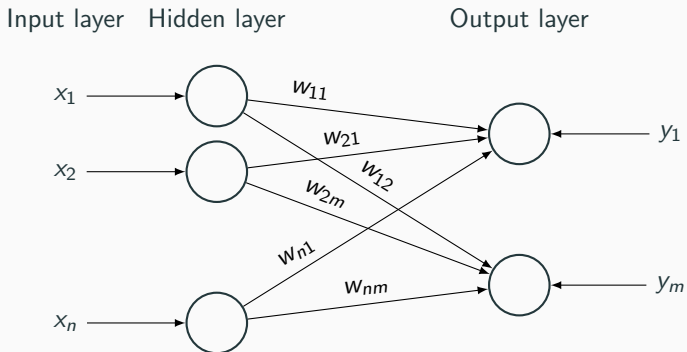
GLM – Support Vector Machine



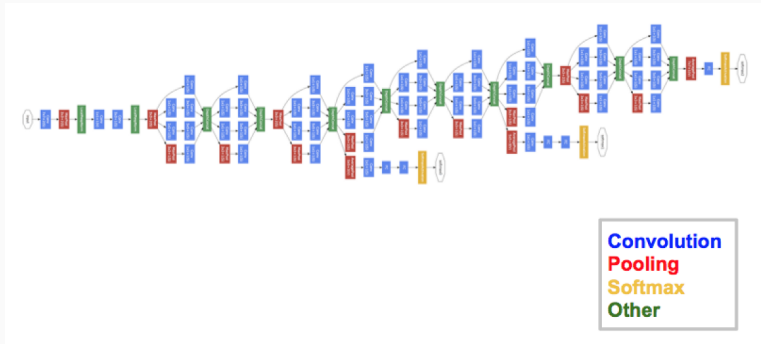
White-box methods



Neural networks







Source: <https://medium.com/analytics-vidhya/cnns-architectures-lexnet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>

Adversarial attacks

Examples



(a)

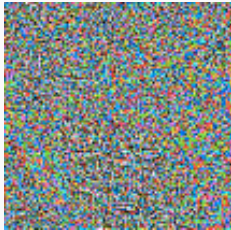


(b)

Examples



(a)

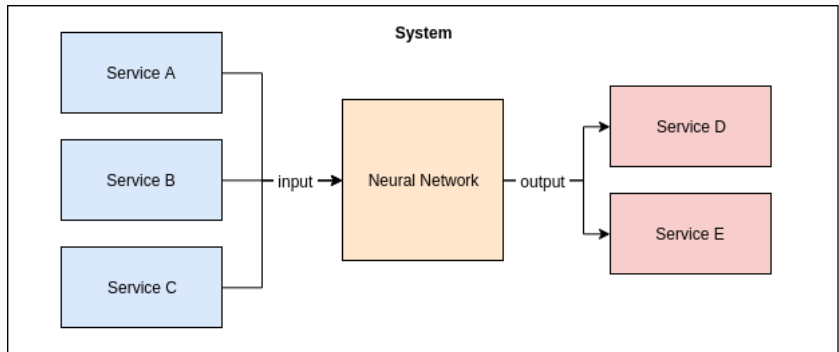


(b)

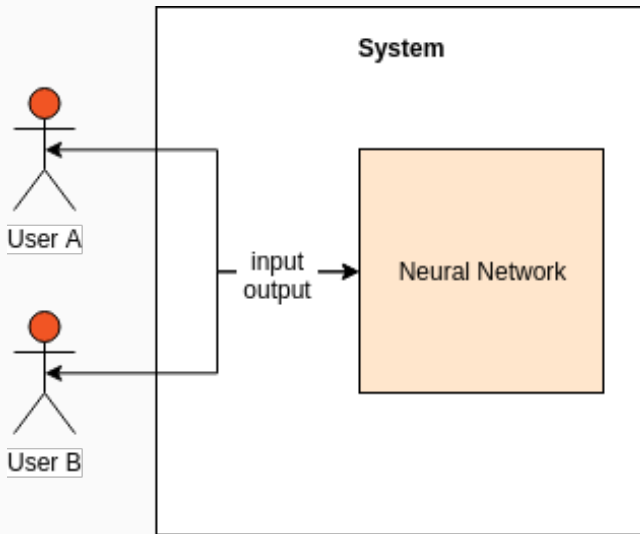


(c)

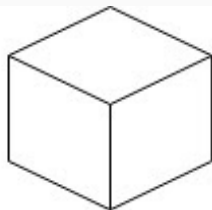
Isolated environments



Non-isolated environment



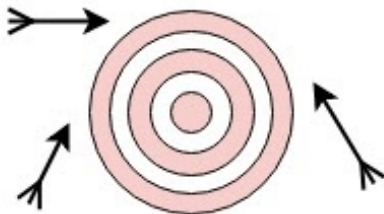
Attack types



WHITE BOX



BLACK BOX

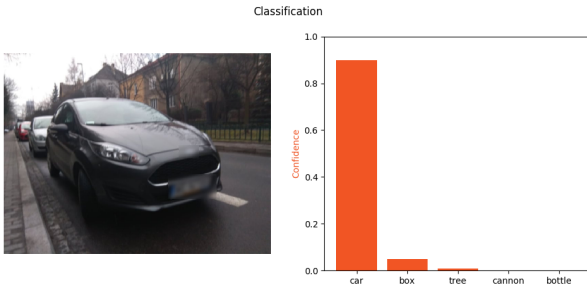


UNTARGETED



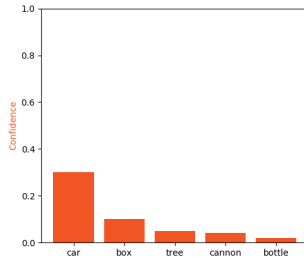
TARGETED

Targeted vs. untargeted attacks



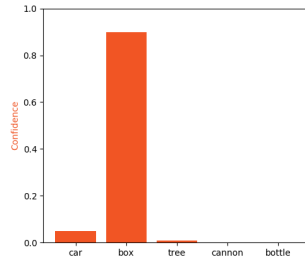
Untargeted attack – confidence reduction

Untargeted attack - confidence reduction

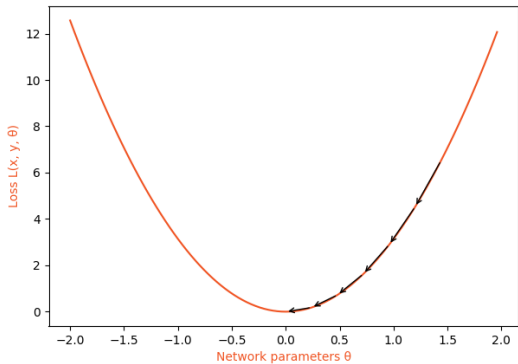


Untargeted attack – misclassification

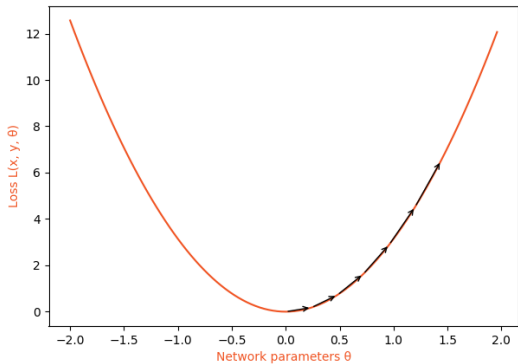
Untargeted attack - misclassification



Gradient decent



Gradient ascent



Attack examples – one pixel attack



Cup(16.48%)
Soup Bowl(16.74%)



Bassinet(16.59%)
Paper Towel(16.21%)











Teapot(24.99%)
Joystick(37.39%)



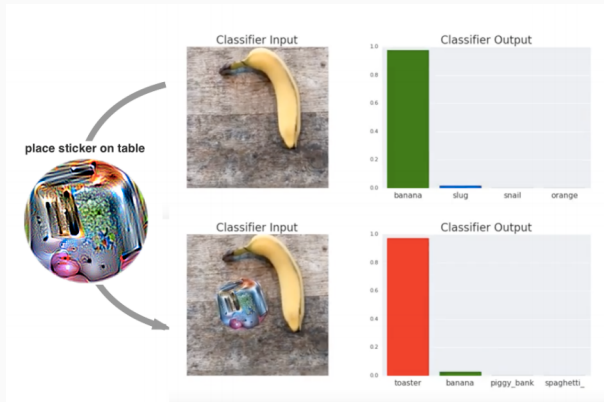
Hamster(35.79%)
Nipple(42.36%)

Black-box attacks

original image	true label	Clarifai.com results of original image	target label	targeted adversarial example	Clarifai.com results of targeted adversarial example
	viaduct	bridge, sight, arch, river, sky	window screen		window, wall, old, decoration, design
	hip, rose hip, rosehip	fruit, fall, food, little, wildlife	stupa, tope		Buddha, gold, temple, celebration, artistic
	dogsled, dog sled, dog sleigh	group together, four, sledge, sled, enjoyment	hip, rose hip, rosehip		cherry, branch, fruit, food, season
	pug, pug-dog	pug, friendship, adorable, purebred, sit	sea lion		sea seal, ocean, head, sea, cute

Source: Liu, Y., Chen, X., Liu, C., Song, D.X. (2016). Delving into Transferable Adversarial Examples and Black-box Attacks.

Patch attack



Camouflage



We cannot totally defend our networks, but there are a few known methods that can increase or at least prevent to some degree such attacks:

- drop-out,
- weight decay,
- input transformation,
- adversarial training,
- feature denoising.

- N. Carlini and D. Wagner. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. Deep Learning and Security Workshop, 2018.
- Carlini et al. Hidden Voice Commands. USENIX Security, 2016.
- Ebrahimi, J., Rao, A., Lowd, D., Dou, D. (2017). HotFlip: White-Box Adversarial Examples for Text Classification. ACL.
- Vijayaraghavan, Prashanth and Roy, Deb. (2019). Generating Black-Box Adversarial Examples for Text Classifiers Using a Deep Reinforced Model.

xAI methods

The methods can be divided into a few groups:

- based on the features influence explanation,
- output explanation,
- neural layers features extraction methods.

We have plenty of feature influence methods, just to mention a few:

- Partial Dependence Plot,
- Permutation Feature Importance,
- Accumulated Local Effects (ALE),
- Individual Conditional Expectation (ICE),
- Feature Interaction,
- Shapley Values

We have plenty of feature influence methods, just to mention a few:

- Scoped Rules,
- Counterfactual Explanations,
- Local Surrogate (LIME),
- SHAP.

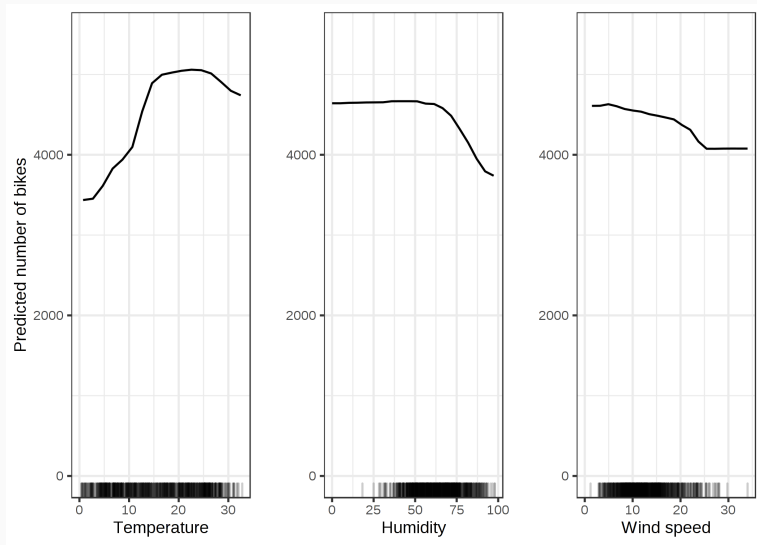
Partial Dependence Plot

The PDP shows the relationship between one or two features on the class. It is defined as:

$$\hat{f}_{x_S}(x_S) = E_{x_C} [\hat{f}(x_S, x_C)] = \int \hat{f}(x_S, x_C) d\mathbb{P}(x_C), \quad (1)$$

where x_S are the features we check the impact on, and x_C are the other features.

Partial Dependence Plot



Source: <https://christophm.github.io/interpretable-ml-book/pdp.html>

Individual Conditional Expectation (ICE)

This method plots one line per model with a feature change. The feature change the prediction. It's easily to find it out on the plot. The disadvantage of ICE plots is that we can plot only one feature at a time. Other disadvantage is that in some cases it would be hard to find the average on the plot.

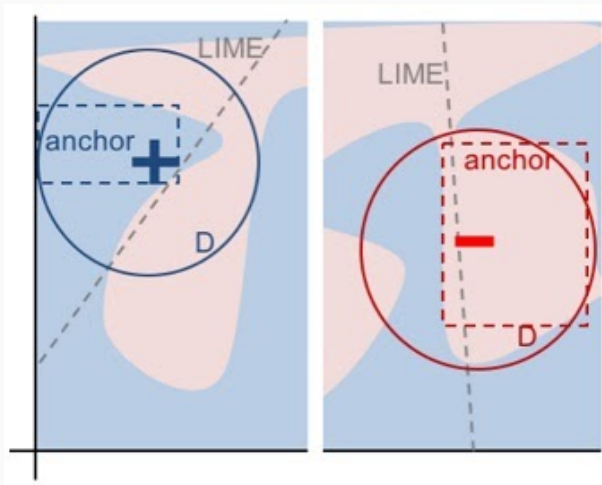
Local Surrogate (LIME)

LIME method is used to explain individual predictions of a black-box model. LIME stands for Local interpretable model-agnostic explanations. It is formulated as:

$$Ex(x) = \arg \min_{g \in G} L(f, g, \pi_x) + \Omega(g), \quad (2)$$

where g is the model, L is the loss function, G is the family of possible explanations, and Ω_G is the model complexity.

LIME



Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, Source: "Why Should I Trust You?" Explaining the Predictions of Any Classifier, 2016

Scoped Rules (Anchors)

This method use reinforcement learning method to find the rules. A rule anchors a prediction if changes in other feature values do not affect the prediction.

This method focus on the features and the influence on the output contribution of each. The prediction model can be as:

$$f(x) = w_0 + w_1x_1 + \dots + w_px_p, \quad (3)$$

where x is the object that we want to predict, x_i are the features, and w_i are the weights corresponding to the features.

Shapley Values

The contribution $\omega_j(f) = w_j x_j - E(w_j X_j) = w_j x_j - w_j E(X_j)$, where $E(w_j X_j)$ is the mean effect estimate for feature j .

The Shapley values is a solution for computing feature contributions for single predictions for any machine learning model.

Advantages:

- fairly distributed between prediction and the average prediction,
- explain a prediction as a game.

Disadvantages:

- is heavy to calculate,
- can be misinterpreted,
- access to the data is needed.

SHAP (SHapley Additive exPlanations)

SHAP stands for SHapley Additive exPlanations. The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalitional game theory. The feature values of a data instance act as players in a coalition. Shapley values tell us how to fairly distribute the prediction among the features.

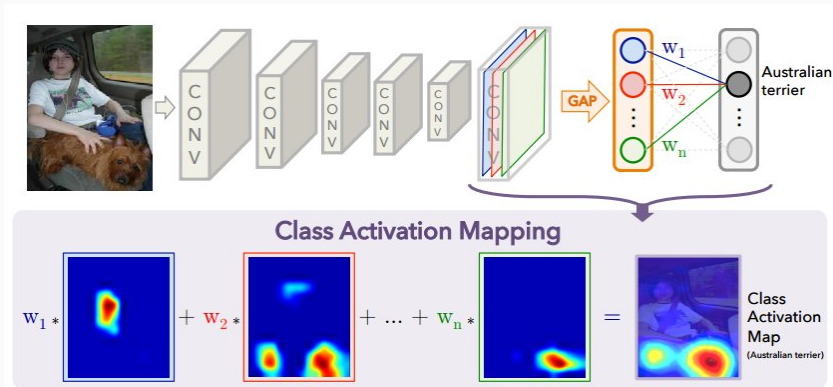
Class Activation Mapping (CAM)

In CAM method we read the features and calculate the global average pooling of the three layers for red, green, and blue:

$$C = w_1 A_R + w_2 A_G + w_3 A_B, \quad (4)$$

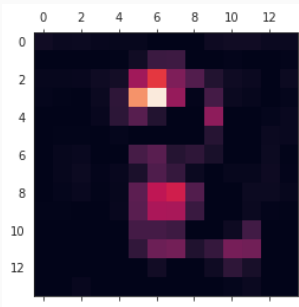
where A_k are the feature maps.

Class Activation Mapping (CAM)



Source: B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. CVPR'16 (arXiv:1512.04150, 2015).

Class Activation Mapping (CAM)



Class Activation Mapping (CAM)

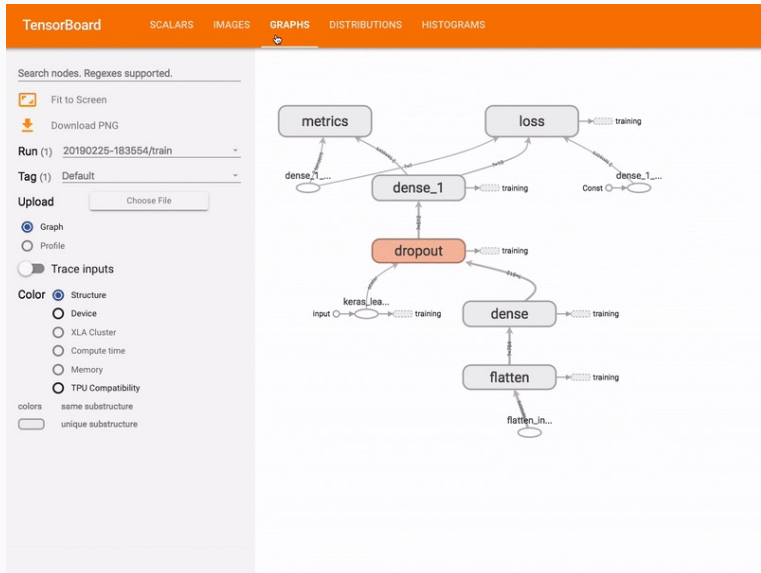


Tools

General tools – sklearn



General tools – tensorboard



You can download it from

<https://eli5.readthedocs.io/en/latest/overview.html>.

It is a tool to debug and explain the prediction of a model. With eli5 we can easily debug the models built using scikit-learn or xgboost. It also works with NLP models. Is a tool for text and feature importance explanation.

You can download it from <https://github.com/EthicalML/XAI>.

It is based on 8 Responsible AI principles. It can be used for data analysis and model evaluation. The data analysis is a similar solution to pandas profiling. Model evaluation can be used together with scikit-learn and keras.

You can download it from <https://github.com/IBM/AIX360>.

Is a tool provided by IBM with many examples and tutorials. It covers several explainability algorithms including data, local and global direct explanation.

You can download it from <https://github.com/slundberg/shap>.

Use the SHapley Additive exPlanations method. The Shap method is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extension

You can download it from <https://github.com/marcotcr/lime>.

Lime is able to explain any black box classifier, with two or more classes. It can also explain the network based on images. It's a good tool for deep networks.

Skater can be downloaded from <https://github.com/oracle/Skater>.

It can be used for various models including NLP, ensemble and image recognition models. It use lime in case of image interpretation.

Questions?