UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

Dipartimento Interateneo di Fisica Michelangelo Merlin

Third ML-INFN Hackathon: Advanced Level

Explainable Artificial Intelligence (XAI)

Alessandro Fania

23/11/2022

Bari Applied Physics (INFN Group-5)

Overview

- What is Explainability?
- Why is Explainability important?
- Is feature importance a good solution?
- Explainable Artificial Intelligence
- Grad-CAM
- SHAP
- Application on VGG16 Model
- Conclusion

What is Explainability?

«Explainability is the degree to which a human can understand the cause of a decision»

Tim Miller-Explanation in artificial intelligence: Insights from the social sciences

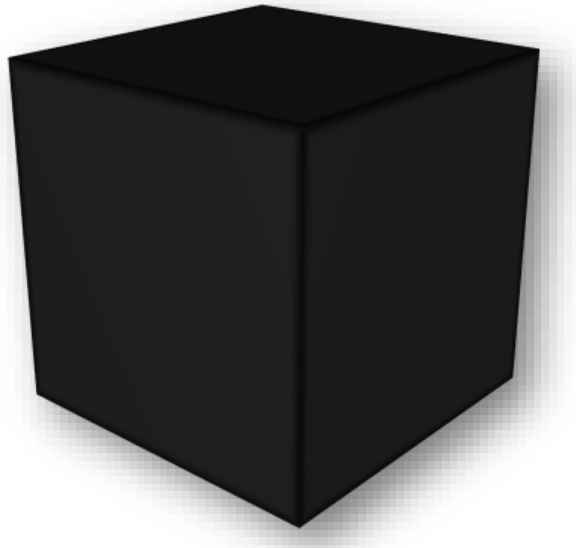
What is Explainability?

Machine Learning and Deep Learning models are difficult to understand

They are often treated as **Black Boxes**

Black box

$$X = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix}$$



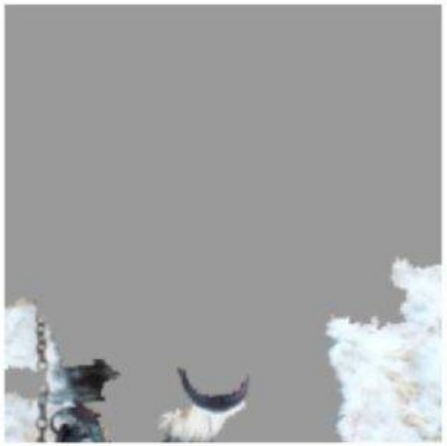
Y

Input data

Predicted output

Why Explainability is so important?

- Helps analysts to understand system outputs simply and quickly.
- Explainability can provide recommendations and spot anomalies for analysts to investigate
- Sometimes AI can give an output that's correct but for the wrong reasons.
- Likewise, Explainability makes possible to understand why a mistake was made and even train the system to stop it from happening again.
- This driver for Explainability provides some overlap with the General Data Protection Regulation (GDPR): the customer has the right to obtain explanations.
- The European Commission recently published the first draft of its Artificial Intelligence Regulation which stipulating requirements around AI and Explainability.



-a husky (on the left) is confused with a wolf, because the pixels (on the right) characterizing wolves are those of the snowy background.

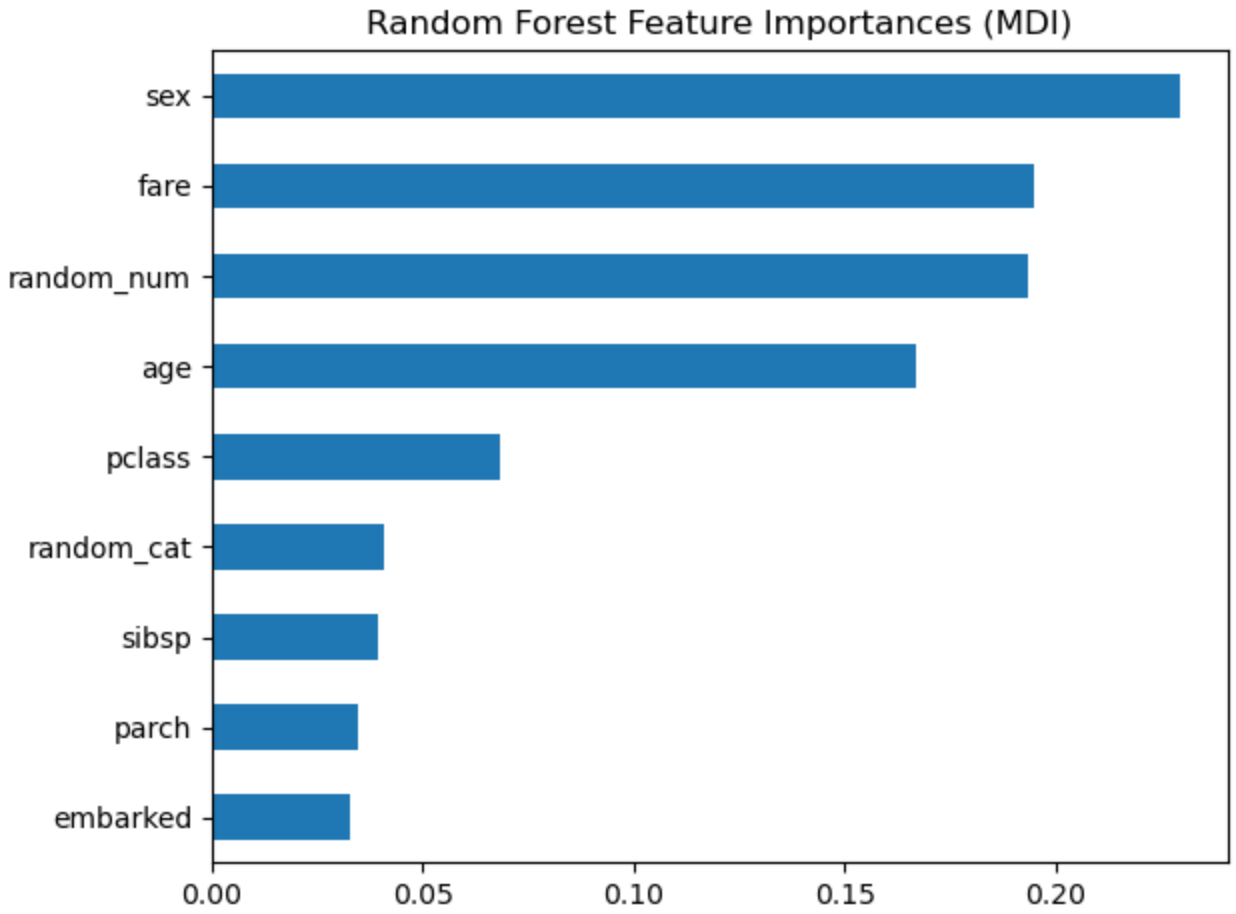
https://towardsdatascience.com/from-data-to-decisions-bef3649f0b17

Is feature importance a good solution?

- Disgregate the final predicton to a single feature attribution and untangle interaction between features are very difficult tasks.
- A possible way to solve this problem would be to use the **feature importance**.

  An example is the *Mean Decrease Accuracy* implemented in the random forest, in which the importance of a feature is evaluated by the change performance after the variable is deleted.

Is then feature importance a good solution?



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

Is feature importance a good solution?

Every good feature importance should have the following properties:

- **Consistency**: a change of the model so that it relies more on a feature (ex. Decision trees), should not decrease the importance of that feature.
- **Accuracy**: given a fixed metric to measure importance of a model, the attribution of each feature should add up to that metric.
- **Insightfulness**: a feature ranking does not provide a good explanation of how a feature increase or decrease the model output.

Unfortunately, it is **not always** possible to have this situation

In addition, feature importance gives **global** information, not on the **single prediction**

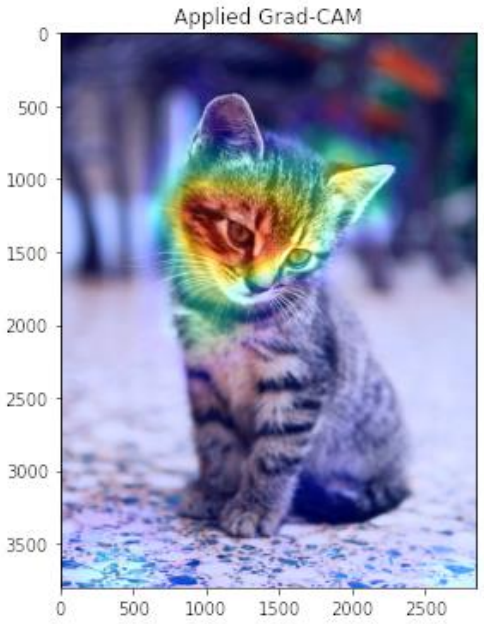
For all these reasons, feature importance is **not** the best strategy to promote Explainability.

Exaplainable Artificial Intelligence

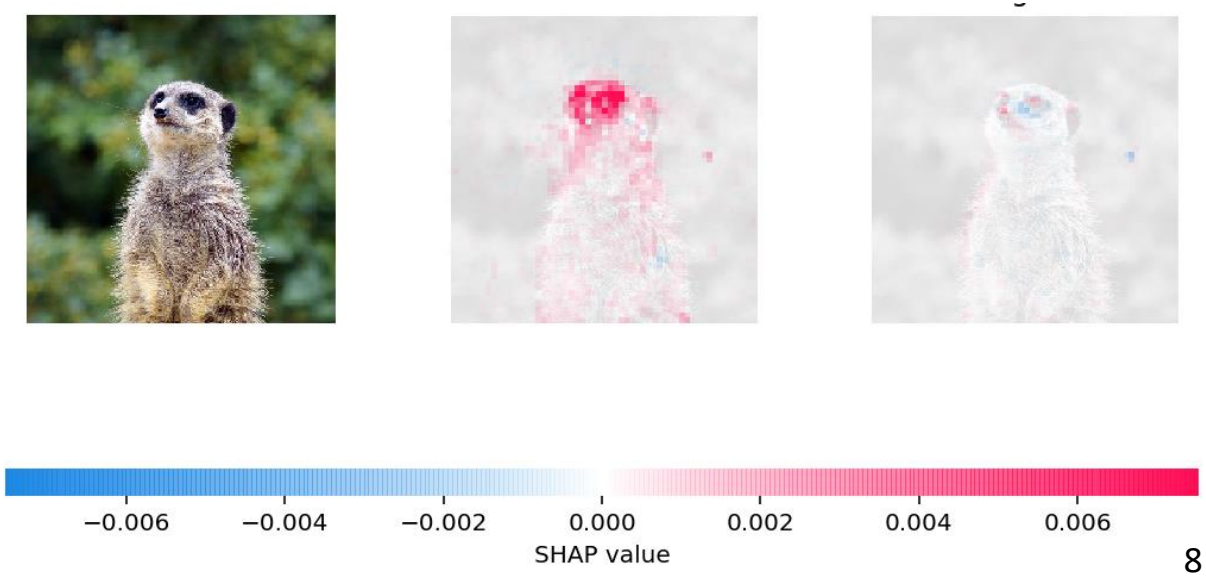
One possible solution is to use **eXplainable Artificial Intelligence** (*XAI*) models.

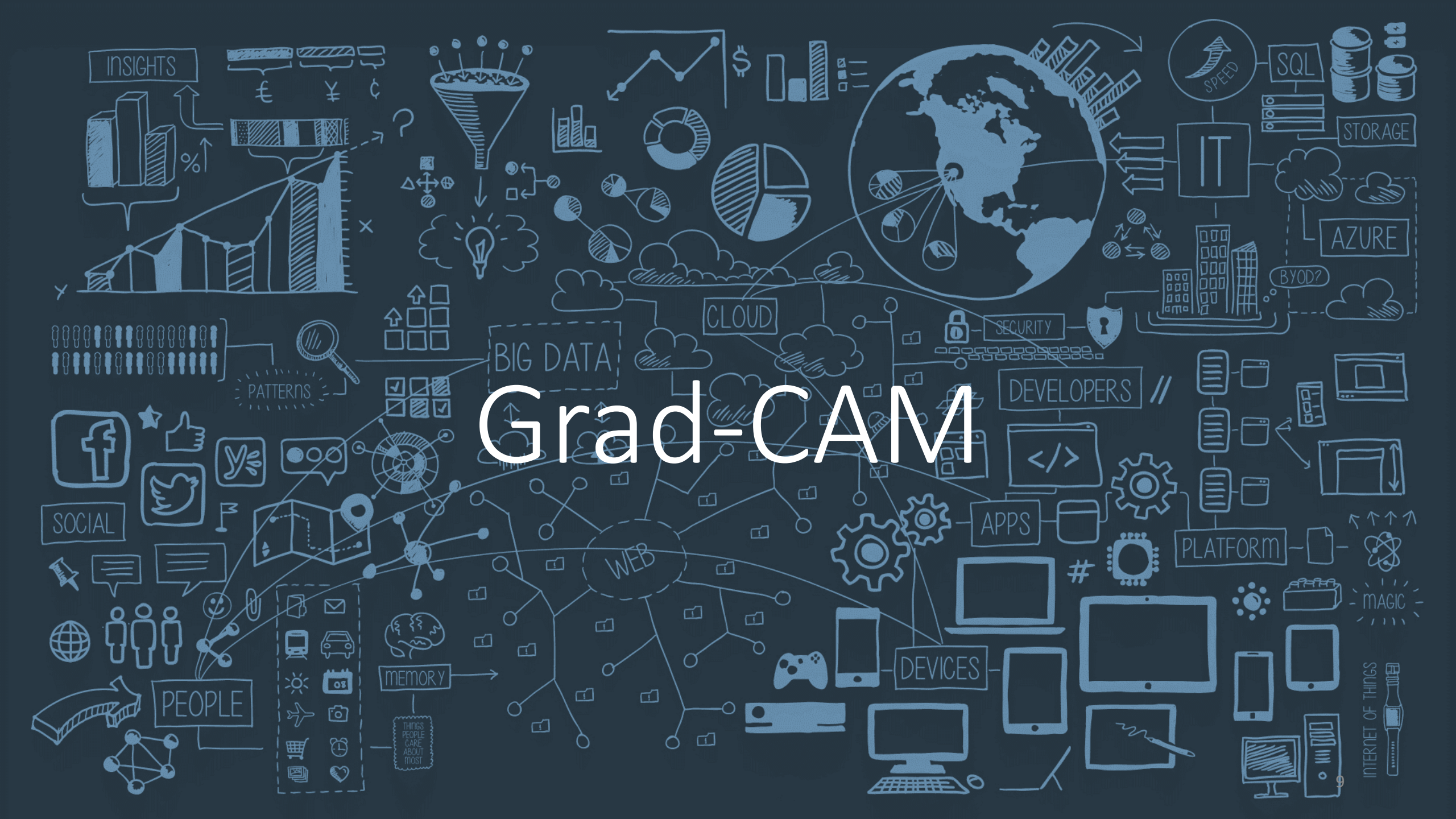
In this short presentation we will see:

Grad-CAM



SHAP




INSIGHTS
SPEED
SQL
STORAGE
IT
AZURE
BYOD?
CLOUD
SECURITY
BIG DATA
PATTERNS
DEVELOPERS
SOCIAL
APPS
PLATFORM
WEB
MAGIC
PEOPLE
MEMORY
THINGS PEOPLE CARE ABOUT MOST
DEVICES
INTERNET OF THINGS

Grad-CAM

CAM-model

What is a CAM model?

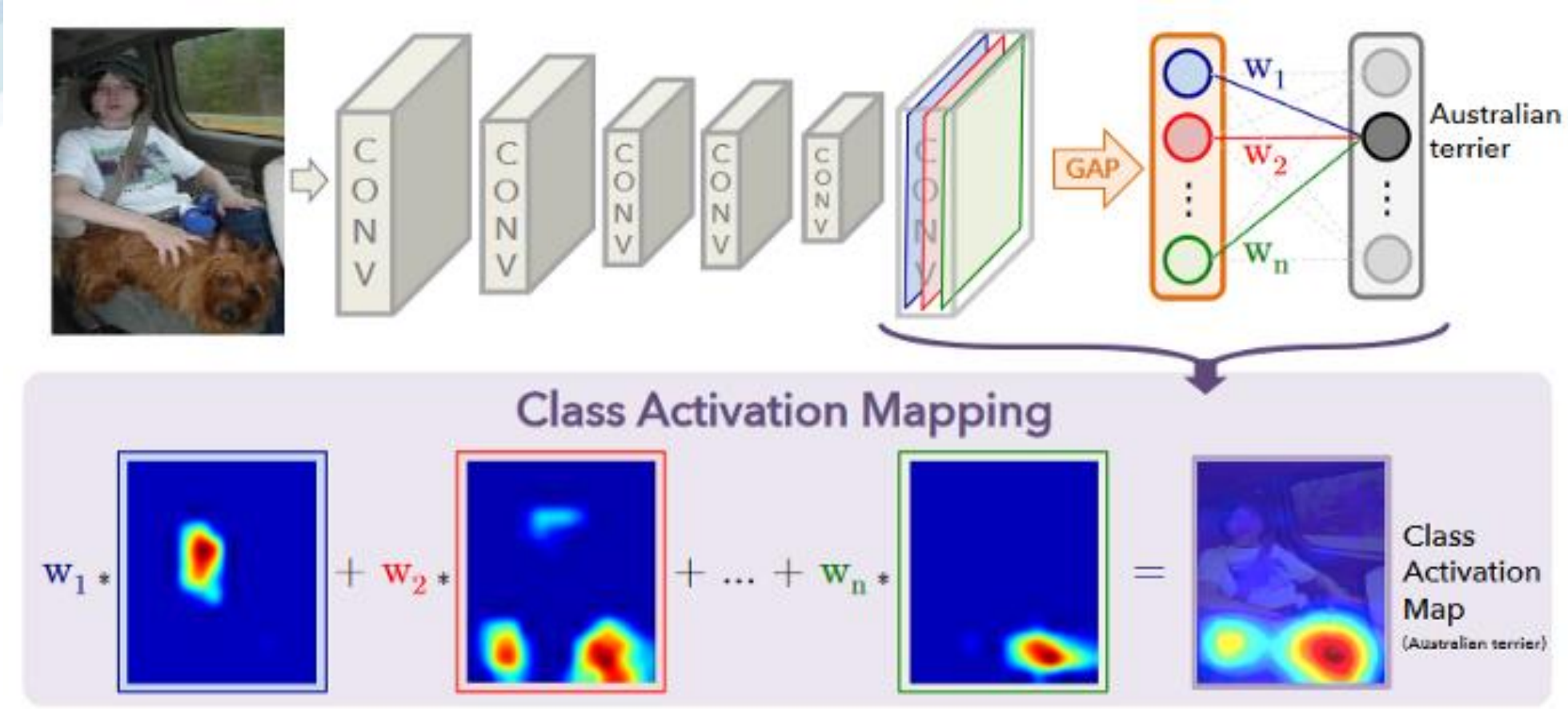
Class Activation Maps (CAM) are a technique to get the discriminative image regions used by a CNN to identify a specific class in the image.

In other words, a class activation map lets us see which regions in the image were relevant to this class.

-Area of the image that explains a prediction "Dog"

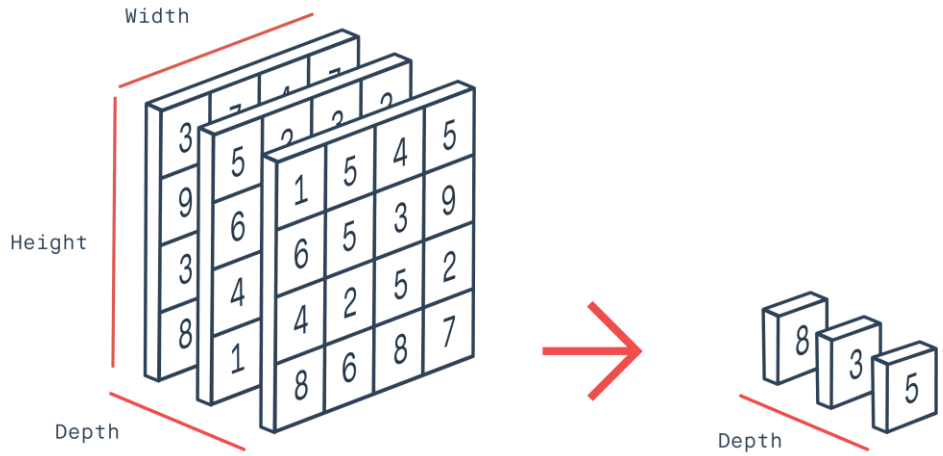
The idea is to use the feature maps of a CNN model as weight to explain a certain prediction

CAM-Architecture

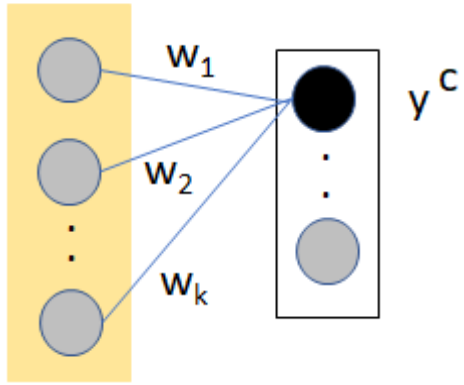


CAM-Algorithm

Post the last convolutional layer in a typical neural network through **Global Average Pooling** you get weights from feature maps



A weight is then assigned to each output per category by either augmenting dense linear layers with **softmax** or by stacking linear classifiers atop of GAP.



CAM-Algorithm

We can then express the c_{th} output of the network as:

$$y^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}$$

$A^k \in \mathbb{R}^{u \times v}$

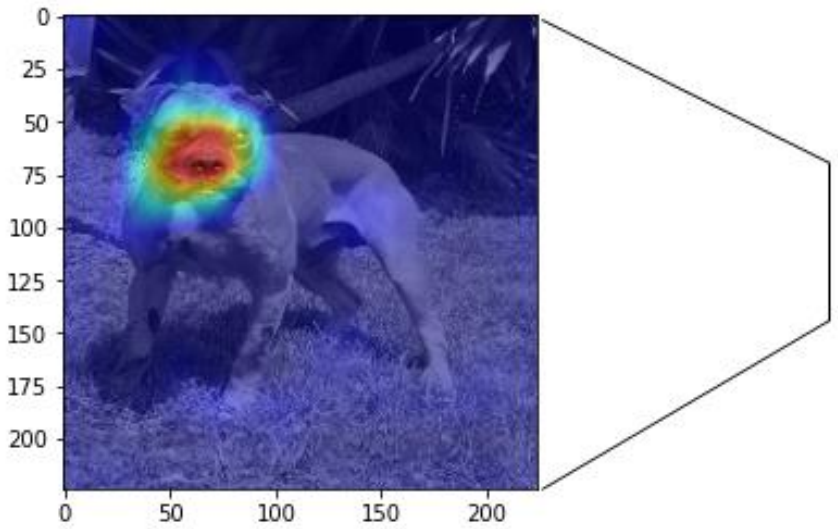
k_{th} feature map

Y_c represents the score of the c_{th} class

Z is the number of pixels in feature map

CAM-Algorithm

We finally get the *class map*:



$$L^c_{\text{CAM}} = \underbrace{\sum_k w^c_k A^k}_{\text{linear combination}} .$$

The assumption of the CAM model is that the final score can always be expressed as a **linear combination** of pooled average feature maps.

CAM-Cons

CAM only works on architectures that have Global Average Pooling (GAP) as a layer before the Dense that deals with the classification:

Limitations:

- The model needs to be modified in order to use CAM.
- The modified model needs to be retrained, which is computationally expensive.
- Since fully connected Dense layers are removed. the model performance will surely suffer. This means the prediction score doesn't give the actual picture of the model's ability.
- The use case was bound by architectural constraints, i.e., architectures performing GAP over convolutional maps immediately before output layer.

Grad-CAM

A possible solution is to use **backpropagation** to calculate the weights of the maps.

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

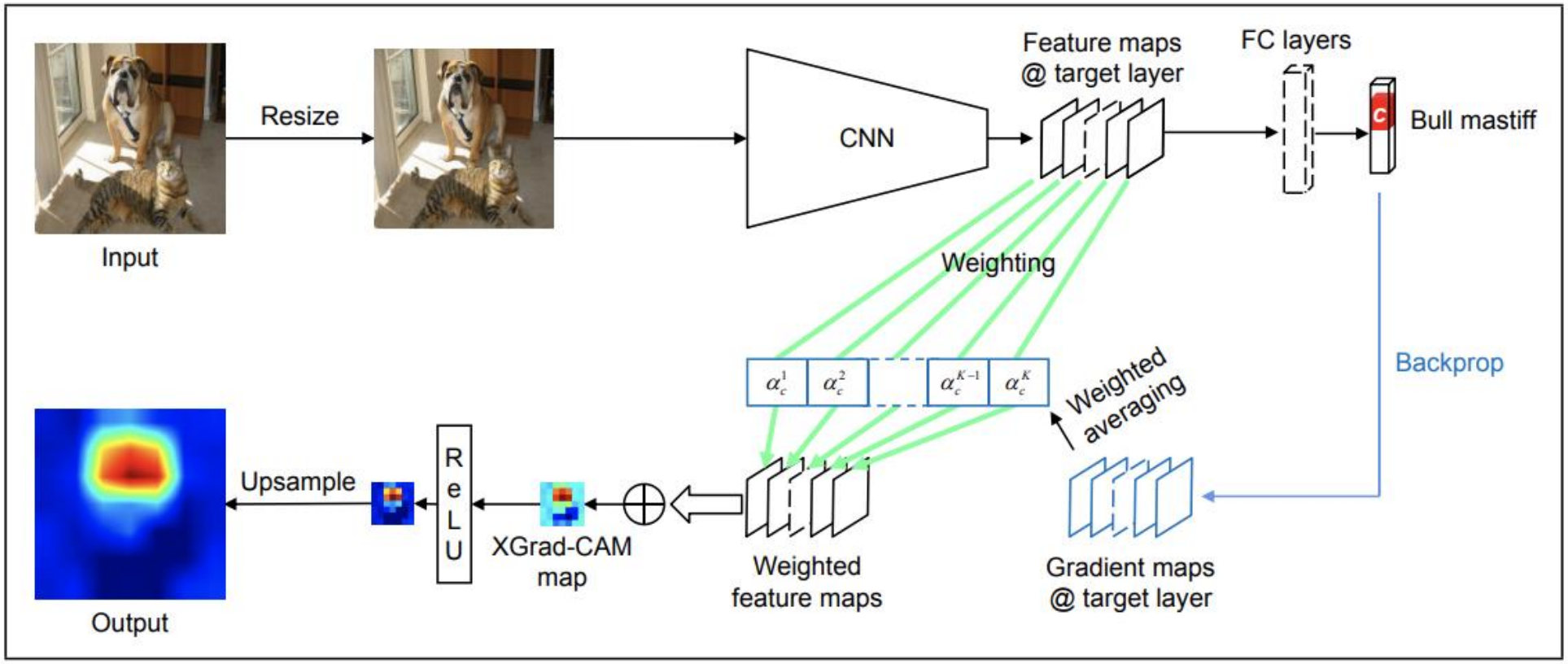
Gradient Class Activation Map (Grad-CAM)

Next, a ***ReLU*** function is applied to zero the negative values of the gradient.

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Grad-CAM Architecture



Grad-CAM

Pro

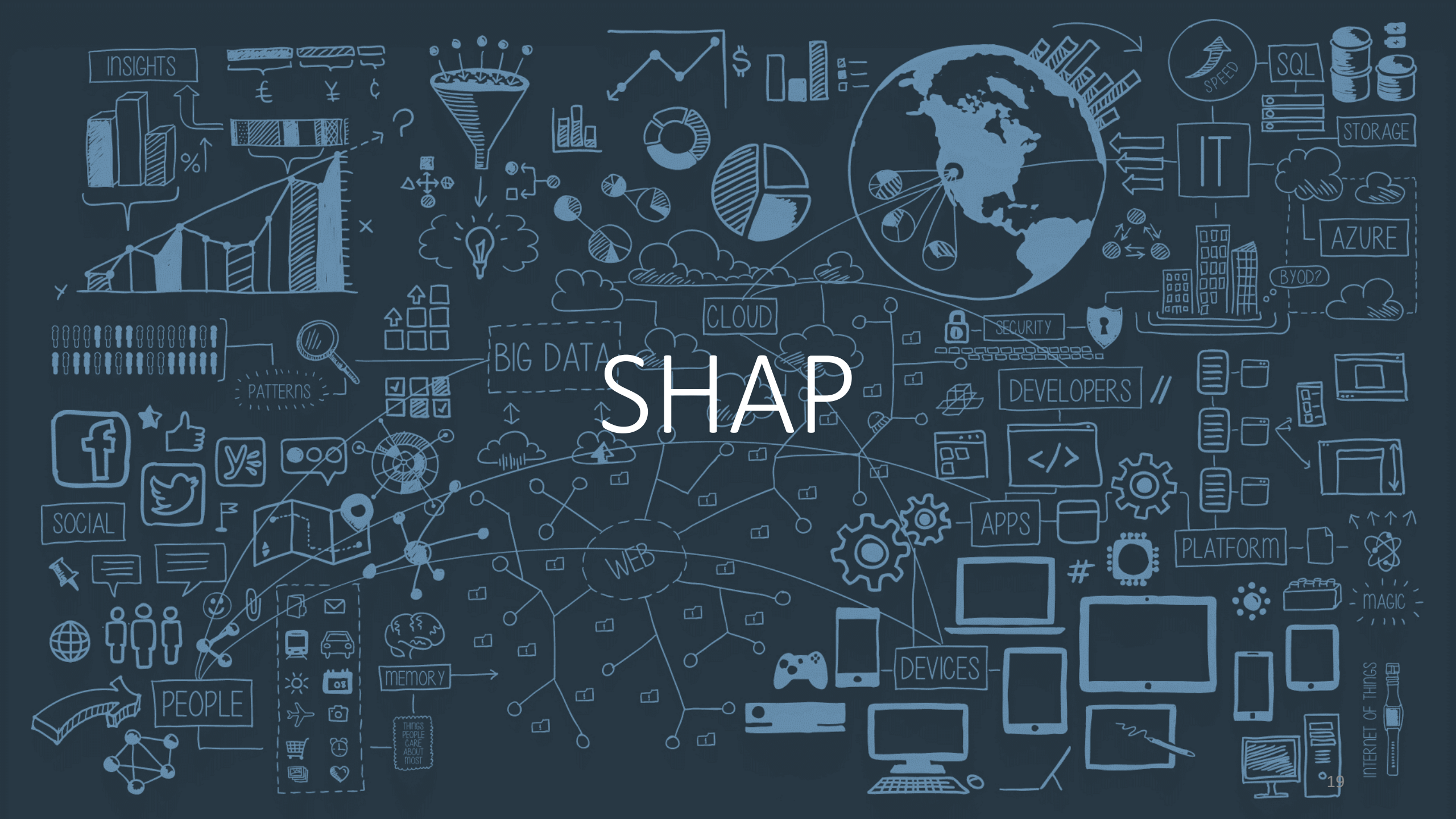
- Unlike CAM we are not bound to use only the features maps of the last convolutional layer, but we can concentrate our analysis for each conv layer of our network.
- Grad-CAM is a generalization of CAM.

Cons

- Consider only positive gradients.

It is applicable to:

•CNNs with fully-connected layers (e.g. VGG) without any modification to the network.
•CNNs used for structured outputs like image captioning.
•CNNs used in tasks with multi-modal inputs like visual Q&A or reinforcement learning, without architectural changes or re-training.

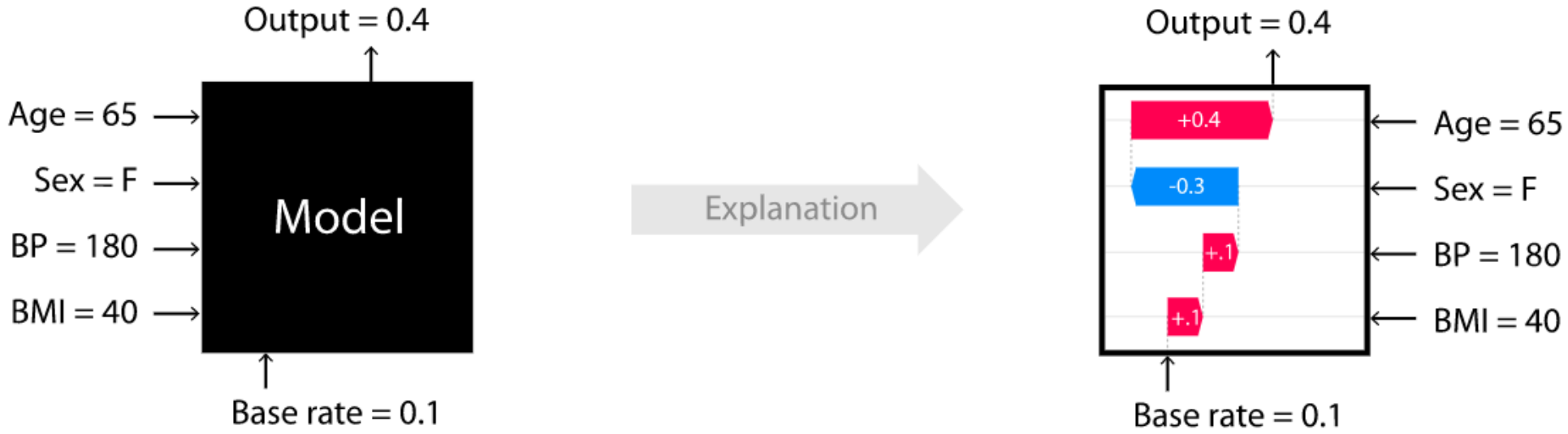

INSIGHTS
SPEED
SQL
STORAGE
IT
AZURE
BYOD?
CLOUD
SECURITY
BIG DATA
PATTERNS
DEVELOPERS
SOCIAL
APPS
PLATFORM
WEB
MAGIC
PEOPLE
MEMORY
THINGS PEOPLE CARE ABOUT MOST
DEVICES
INTERNET OF THINGS

SHAP

SHAP-Introduction

SHAP is a model agnostic explainer:

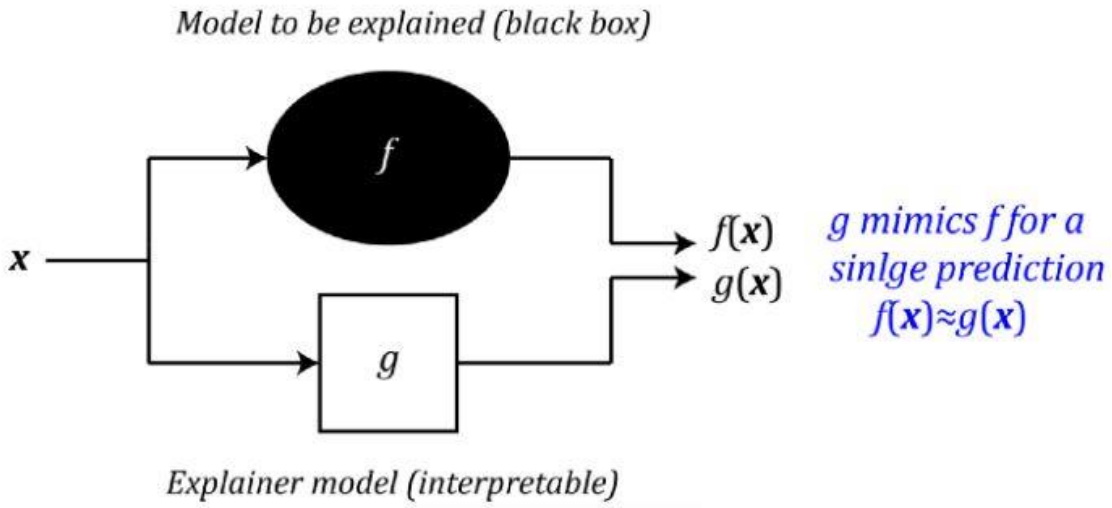
- Its purpose is to "imitate" the model used.
- It gives an understandable explanation of a **local prediction** of a model by assigning to each feature a value.



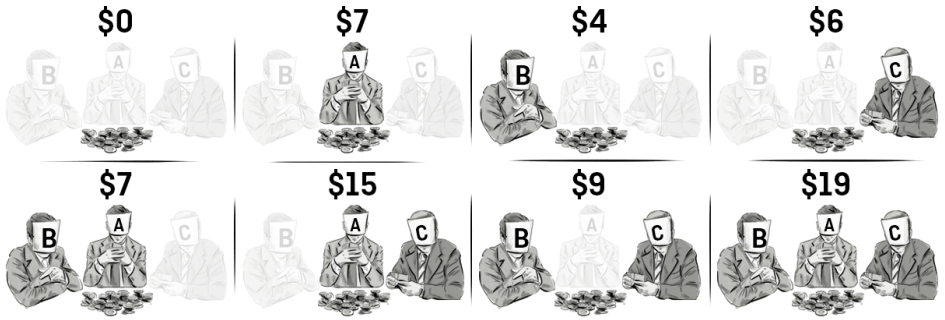
Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017).

SHAP-How it works

How to achieve this?
It uses the concept of **Shapley Value.**



The *Shapley value* is a mathematical concept in game theory that was introduced by Lloyd Shapley in 1951 [72], later awarded the Nobel Prize in Economics for this work.



-worth of a coalition on which is Shapley value based on

SHAP-How it works

- Let's consider a game in co-operative with *m* numbered players and call *F* the set of such players.

$$\mathrm{F} = \{1, 2, 3, 4, \dots, m\}$$

- We then define an *S* coalition as a subset of *F*, which also includes the empty set without players.
- An example of the possible coalitions with 3 players

$$\{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$$

SHAP-How it works

- Let us now define a function ν, called a **characteristic function**, which associates each coalition with a real number. The value *ν(S)* will be called the **worth of the coalition** S and represents the total gain obtained by the coalition if the members act together.
- Obviously in the trivial case:

$$\nu(\emptyset) = 0$$

- At this point, for the calculation of the value of {3}, for example, you can proceed by subtracting:

$$\nu(\{1,2,3\}) - \nu(\{1,2\})$$

The aim now is to assess what each player's contribution is in the total payoff.

SHAP-How it works

Considering the number of permutations and summing up on all other possible combinations we get:

$$\phi_i = \sum_{S \subseteq F - \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} (\nu(S \cup \{i\}) - \nu(S))$$

where $|F|!$ is the total number of permutations of the grand coalition, *S* indicates the coalition and $S \cup \{i\}$ indicates the coalition with the addition of *i*.

The ϕ_i value is called **Shapley Value** and represents the average contribution of the player, or variable, *i*.

Shapley Values in Machine Learning

How translate this in Machine Learning?

- **SHAP** (Shapley Additive Explanation)

Input data	x	z'
Input model	$f(x)$	$g(z')$

Simplified local inputs (generally $z' \in (0,1)$)

Explanatory models

If $x \approx z'$ then $f(x) = g(z')$

Shapley Values in Machine Learning

We assume:

SHAP values

$$g(z') = c_0 + \sum_{i=1}^{M} c_i z'_i$$

Additive Feature Attribution

- c_i represent the contribution of each feature.
- c_0 is the average output of the model.
- Easy to interpret.

Shapley Values in Machine Learning

How we calculate SHAP values?

Player ⟷ Feature

$$\nu(\emptyset) = 0 \longleftrightarrow f(NA, NA, ..., NA) = E[f(x)] = \frac{1}{k}\sum_{j=1}^{k} f(x^{(j)})$$

$$f_S(x_S) = E[f(x)|x_s]$$

$$\nu(S) = f_S(x_S) - E[f(x)]$$

Shapley Values in Machine Learning

Finally we get:

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{(|z'|-1)(|x'|-|z'|)!}{|x'|!}(E[f(z)|z_S] - E[f(z)|z_{S\{i\}}]$$

Shapley values calculated using the conditional expectations are called **SHAP** values.

Shapley Values in Machine Learning

It is proved that the function $g(z')$ has the following property:

Local Accuracy:
- If $x \approx z'$ then $f(x) = g(z')$.

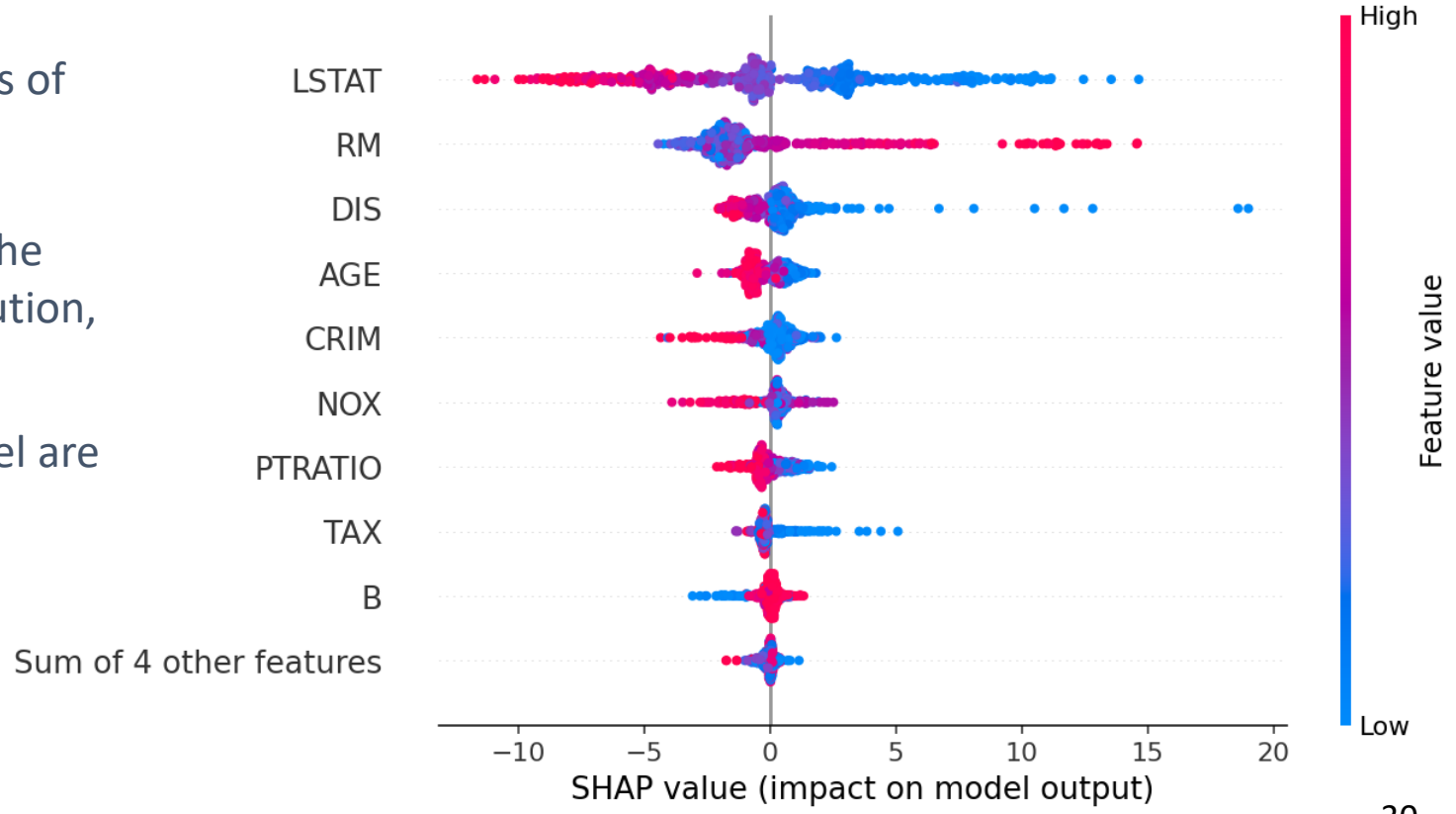
Missingness:
- If $z' = 0$, then $\phi_i = 0$.

Consistency:
- If feature contribution changes, the feature effect cannot change in the opposite way.

SHAP-Plot

The SHAP library allows the display of SHAP values for each feature of the model.

- In the example opposite, the distributions of the SHAP values of a model are shown.
- The values in red refer to high values of the feature, compared to the starting distribution, vice versa the blue values to low values.
- The most impactful features for the model are placed at the top.
- This plot refers to a test sample but can also be used for a local instance.

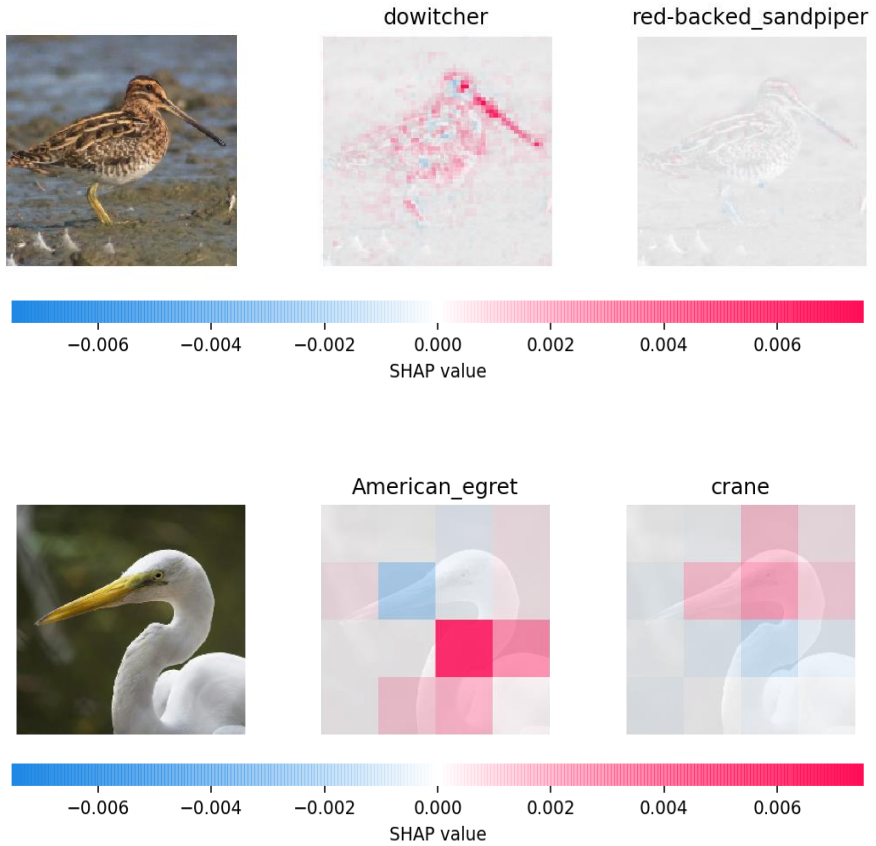


SHAP for Deep Learning: DeepExplainer vs GradientExplainer

The SHAP model is also adapted for the explanation of deep learning models.

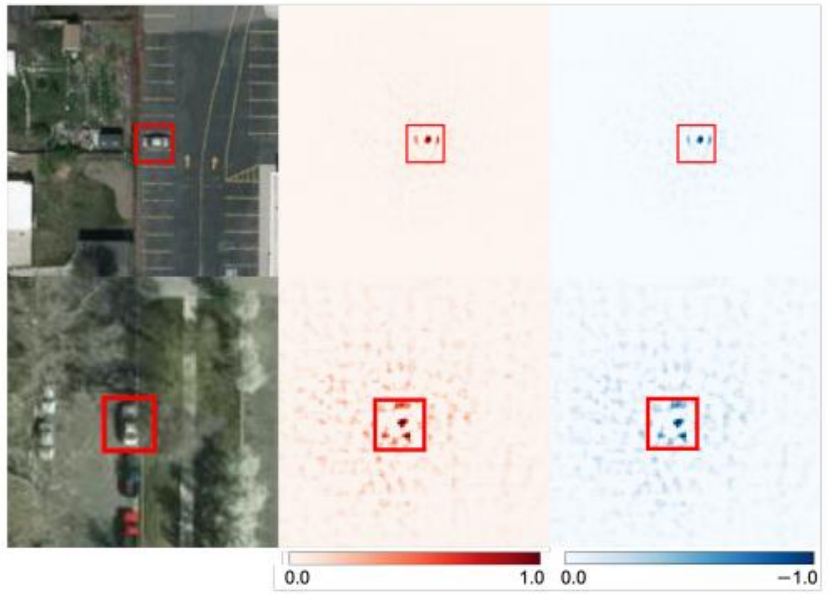
- The function in question is **DeepExplainer**:
  - It can have both *tabular* and *image* data as input.
  - In the case of images, *pixels* will be considered as features to which to assign the SHAP values

- Another very similar function is **GradientExplainer:**
  - it is possible to assign importance to the various intermediate layers of the neural network.
  - In the case of a CNN the shap value is assigned to the pixels of the feature maps.

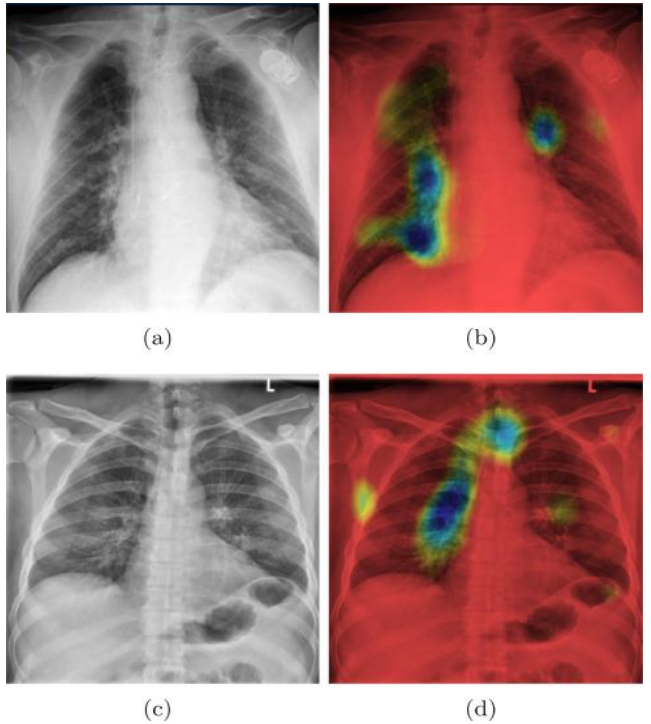


Grad-CAM and SHAP applications

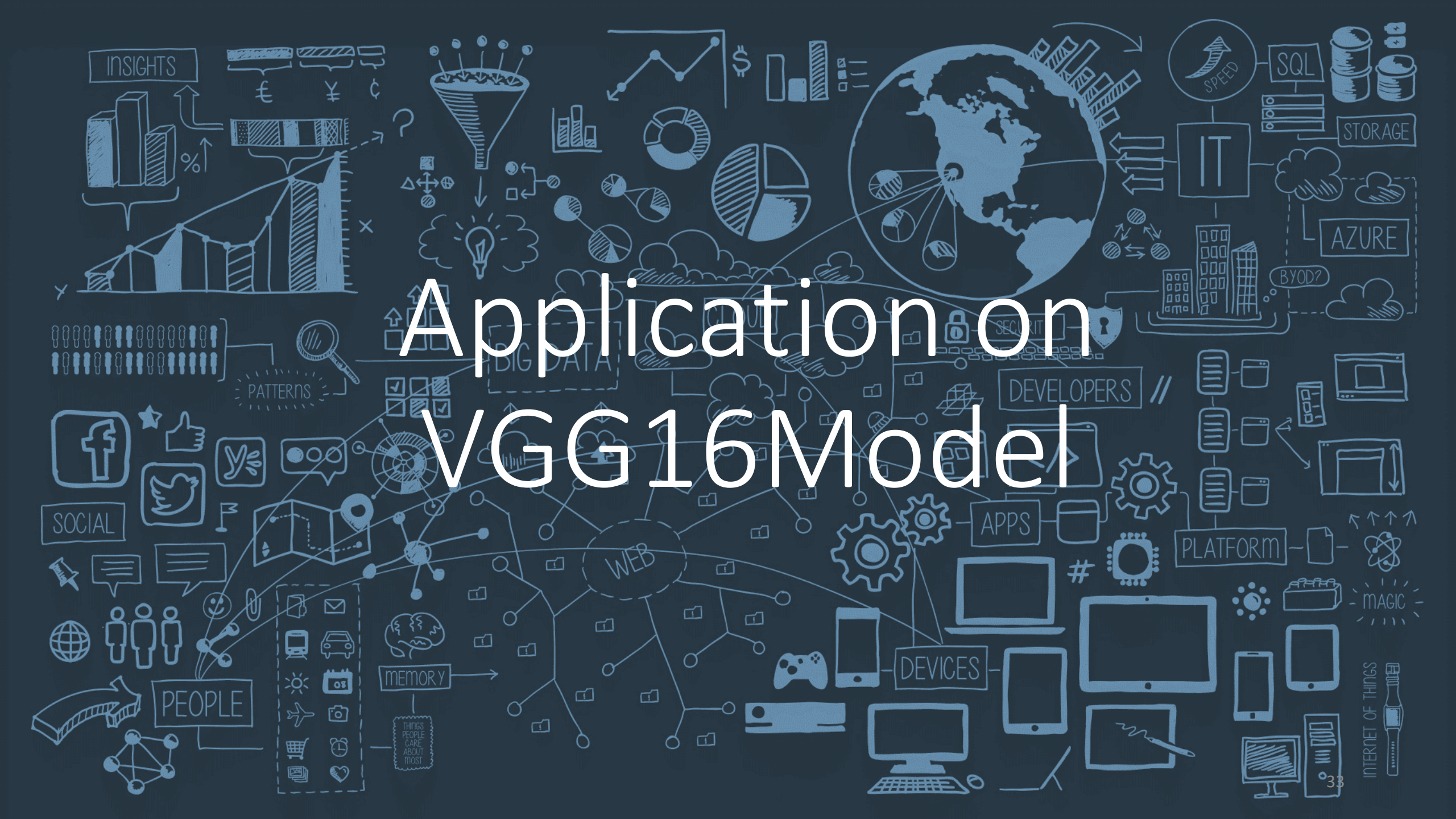
- There are numerous applications of the models seen.
- Fields range from analysis of *medical imagery*, *satellite data* and much more.



H Kawauchi et al. SHAP-Based Interpretable Object Detection Method for Satellite Imagery, 2022



H.Panwar et al. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images, 2020

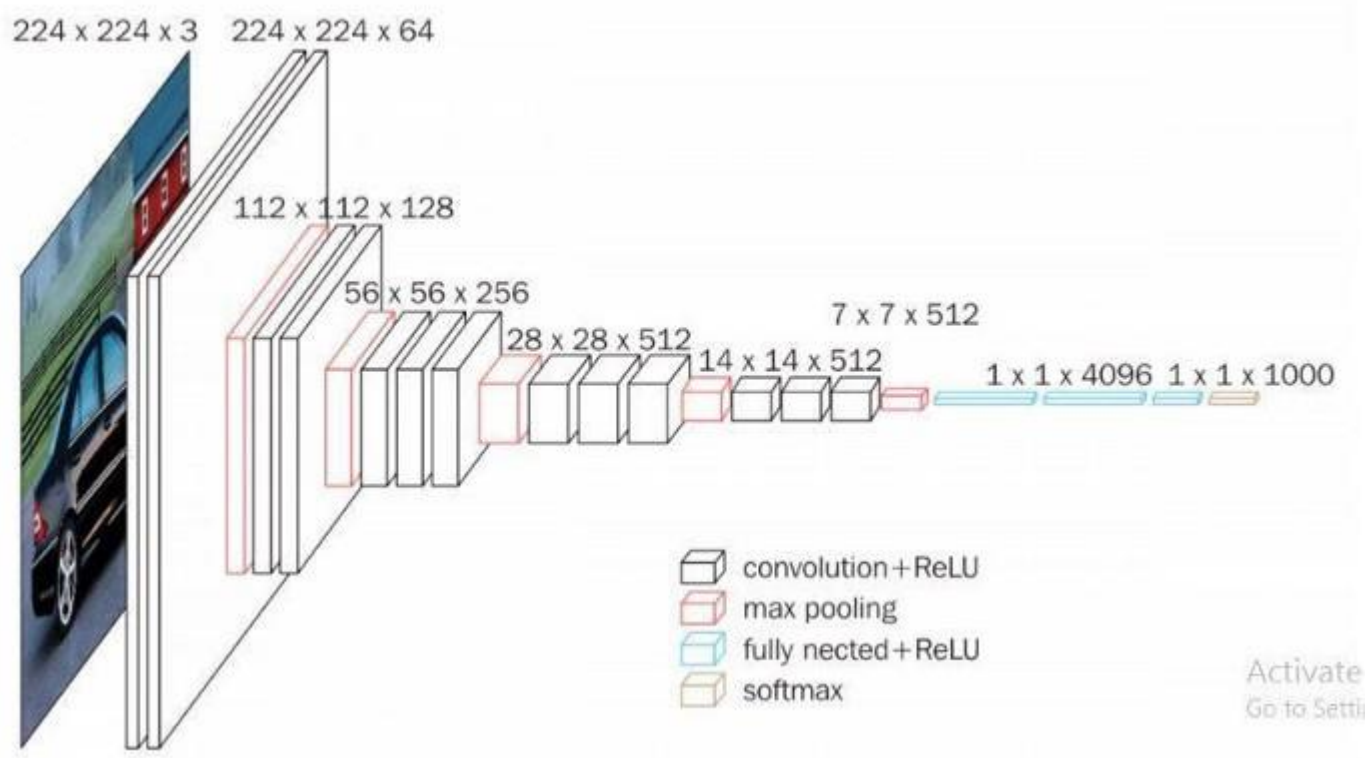


Application on VGG16Model

VGG16 model

VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date.

VGG16 is able to classify 1000 images of *1000 different categories* with **92.7% accuracy**. It is one of the popular algorithms for image classification and is easy to use with **transfer learning**.




INSIGHTS
SPEED
SQL
STORAGE
IT
AZURE
BYOD?
CLOUD
SECURITY
BIG DATA
DEMO
PATTERNS
DEVELOPERS
SOCIAL
APPS
PLATFORM
WEB
MAGIC
PEOPLE
MEMORY
DEVICES
INTERNET OF THINGS

Grad-CAM vs SHAP?

Grad-CAM and SHAP are different explainable methods:

Grad-CAM:
- Based on Feature maps
- Consider only positive gradients.
- Applicable only on CNNs.

SHAP:
- Pixel based.
- Give explanation of how the value of the feature influenced the prediction.
- Applicable on several models.

In conclusion, which is better?
- The best way is to consider the *consensus method*.
- If not possible, consider to use the most reasonable one.

Conclusions

In summary:

- The development of increasingly efficient algorithms that are difficult to interpret;
- The increasingly frequent use of artificial intelligence models in various areas, especially in the medical field;
- The recent rules that regulate the use of personal data for economic purposes;

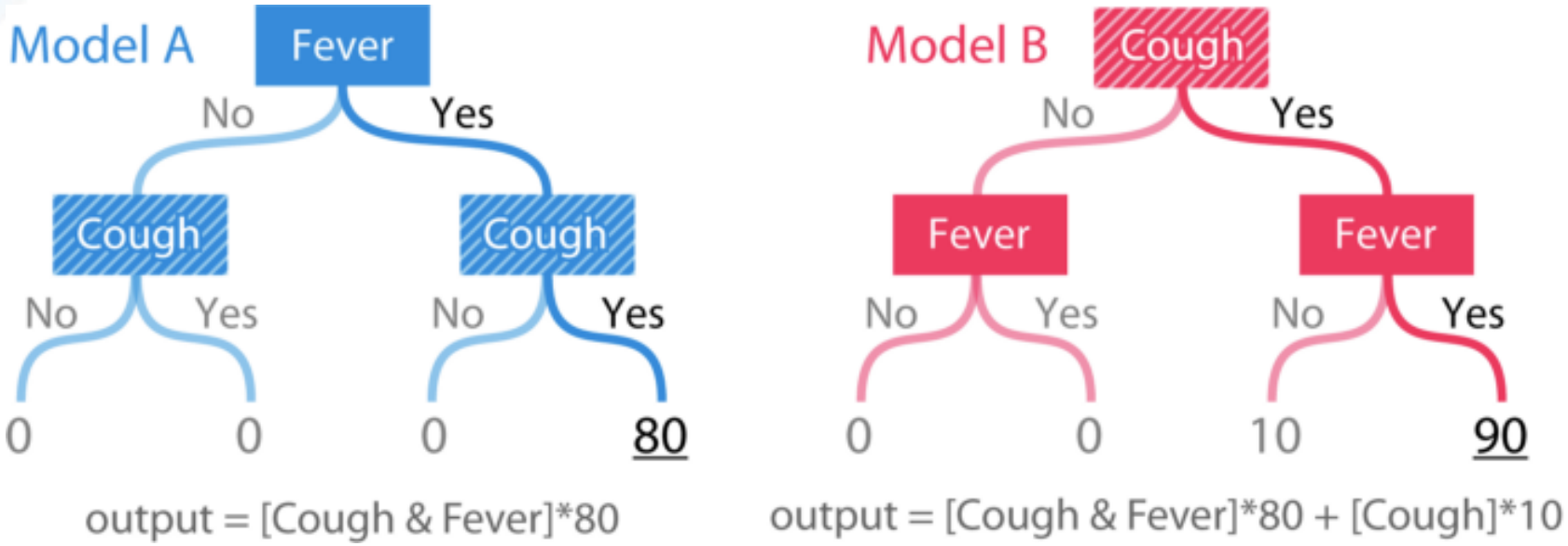
It is essential to consider the role of explainability

- In this presentation we have presented models of XAI applied to images, but of course there are many other applications for tabular data.
- The soil is very fertile for the development of interpretable models.

Thanks for your attention

Questions?

What about consistency?

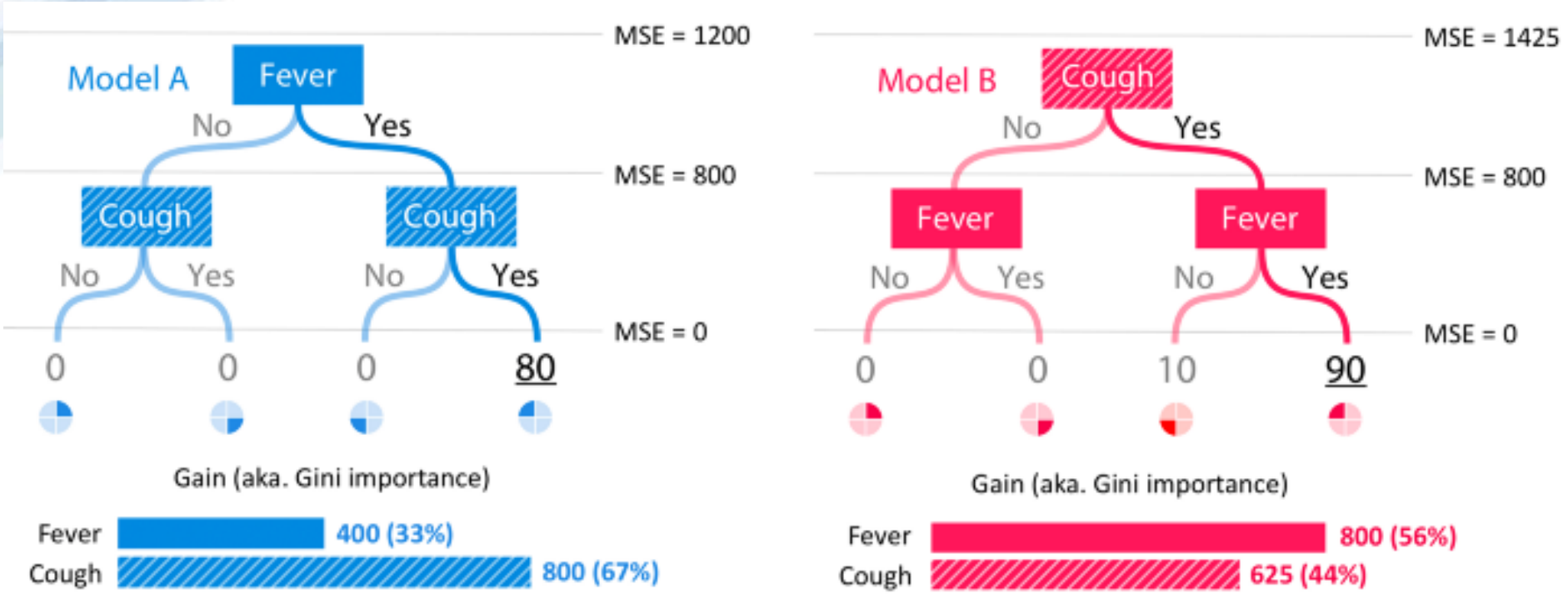


The output of the models is a risk score based on a person's symptoms. Model A is just a simple "and" function for the binary features *fever* and *cough*. Model B is the same function but with +10 whenever *cough* is yes.Cough is clearly more important in model B than model A.

What about consistency?

- A widely used method as **gini importance** can lead to such clear inconsistency results.
- Assume that 25% of our data set falls into each leaf, and that the datasets for each model have labels that exactly match the output of the models. This is the error from the constant mean prediction of 20.
- After splitting on *fever* in model A the MSE drops to 800, so the gain method attributes this drop of 400 to the *fever* feature.
- Splitting again on the *cough* feature then leads to an MSE of 0, and the gain method attributes this drop of 800 to the *cough* feature.
- In model B the same process leads to an importance of 800 assigned to the fever feature and 625 to the *cough* feature.

What about consistency?



- Features near the root of the tree to be more important than features split on near the leaves.
- When *cough* becomes more important its attributed importance **drops**.

Grad-CAM is a generalization of CAM (proof)

From the expression of Y_c:

$$Y^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}$$

Let's call: $F^k = \frac{1}{Z}\sum_i\sum_j A_{ij}^k \longrightarrow Y^c = \sum_k w_k^c \cdot F^k$

Partial derivating we get:

$$\frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A_{ij}^k}}{\frac{\partial F^k}{\partial A_{ij}^k}} \longrightarrow w_k^c = \sum_i\sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

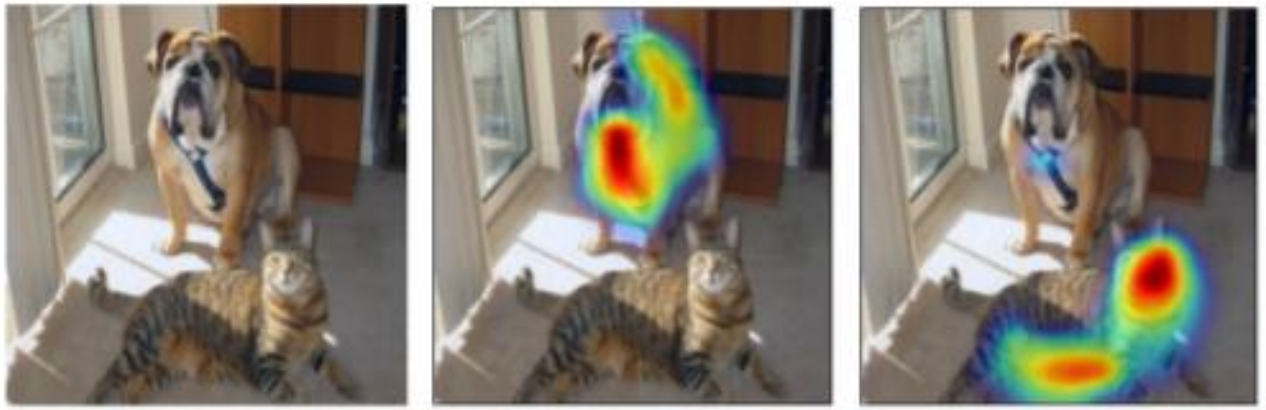
Generalization of CAM

Another use of Grad-CAM

Counterfactual Explanations:
The network can be made to change its predictions by negating the gradient of y_c with respect to feature maps A^k of a convolutional layer.

Thus, the importance weights now become:

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{-\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{Negative gradients}}$$



(a) Original Image (b) Cat Counterfactual exp (c) Dog Counterfactual exp

Grad-CAM vs Shap-CAM