Explainable AI: The unreachable peak

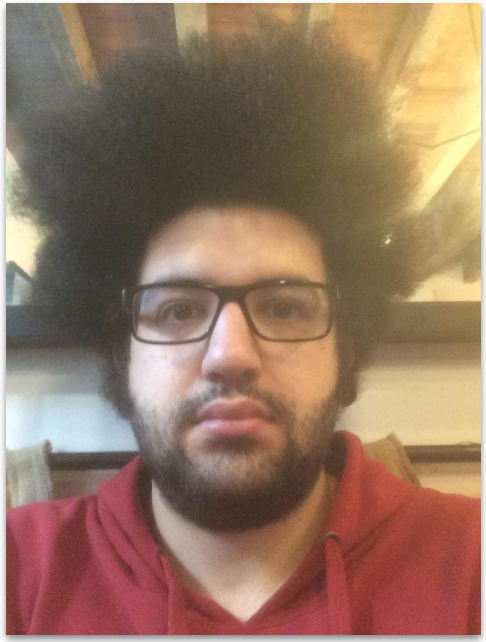
Payam Esfandiari

About me

- PhD in Computer Science
- Ex Googler
- Former Head of Data Team @ blu Bank
- **Current Head of Data Team @ Snapp! Express**
- Avid Gamer

All and all,

Have Some experience in the field



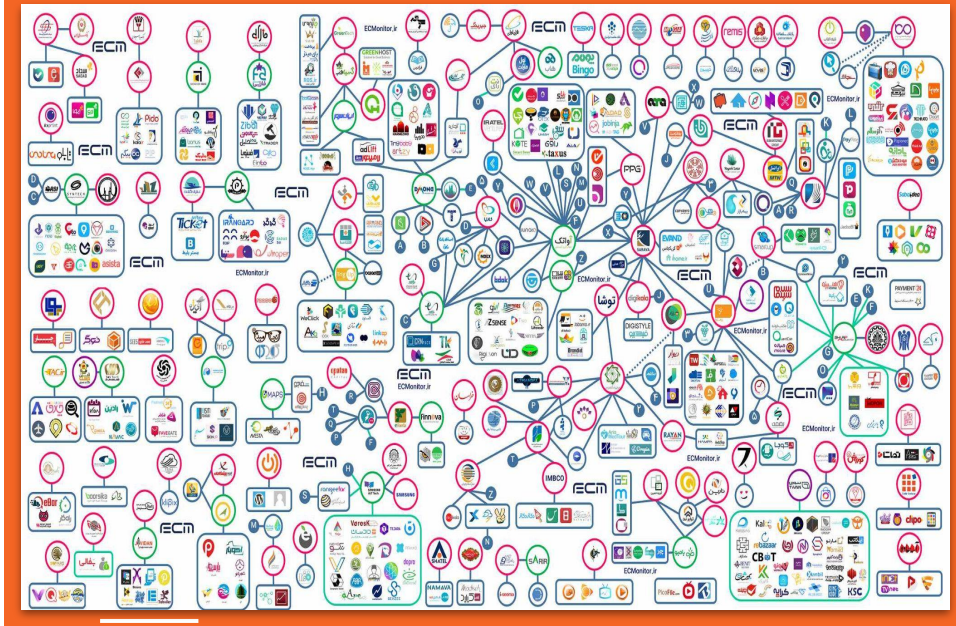
Agenda

- Part I : Introduction and Motivation
  - Definition and Motivation
  - Challenges of XAI
- Part II : Explainable ML
  - From Machine Learning to Knowledge Representation
- Final Notes

Introduction and Motivation

Introduction

From Business Perspective



Business to Customer

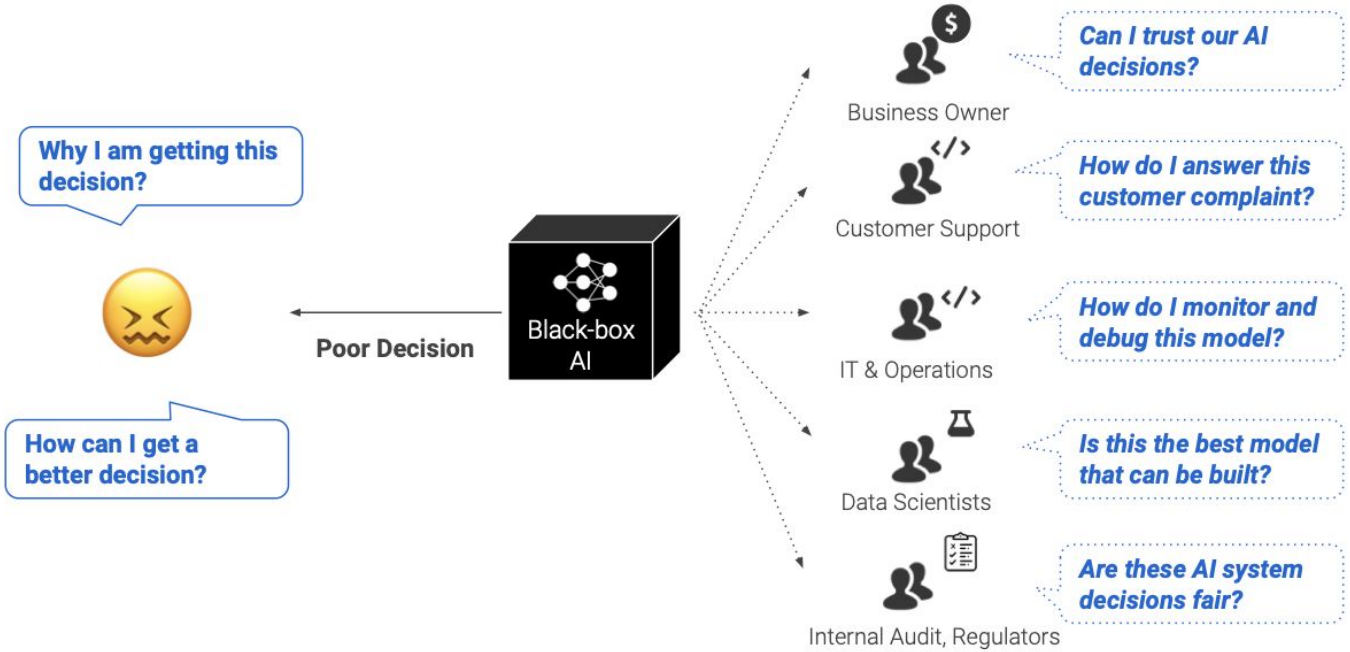
Business to Customer

Business to Customer

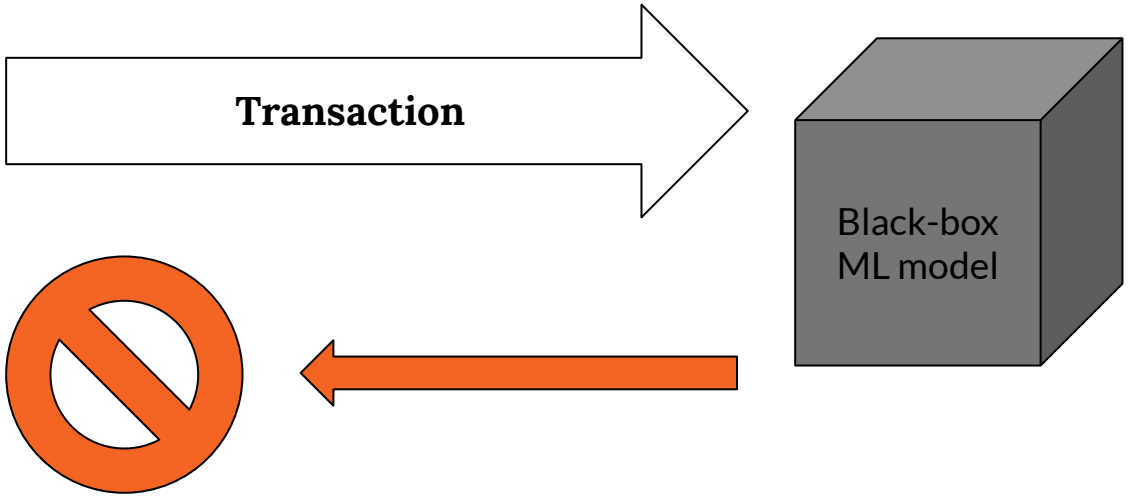
Business to Customer

- Finance
  - Credit Scoring, Loan Approval
  - Insurance
- Healthcare
  - COVID detection
  - Responsibility? Confidentiality ?

AI is a black-box



Example - Fraud

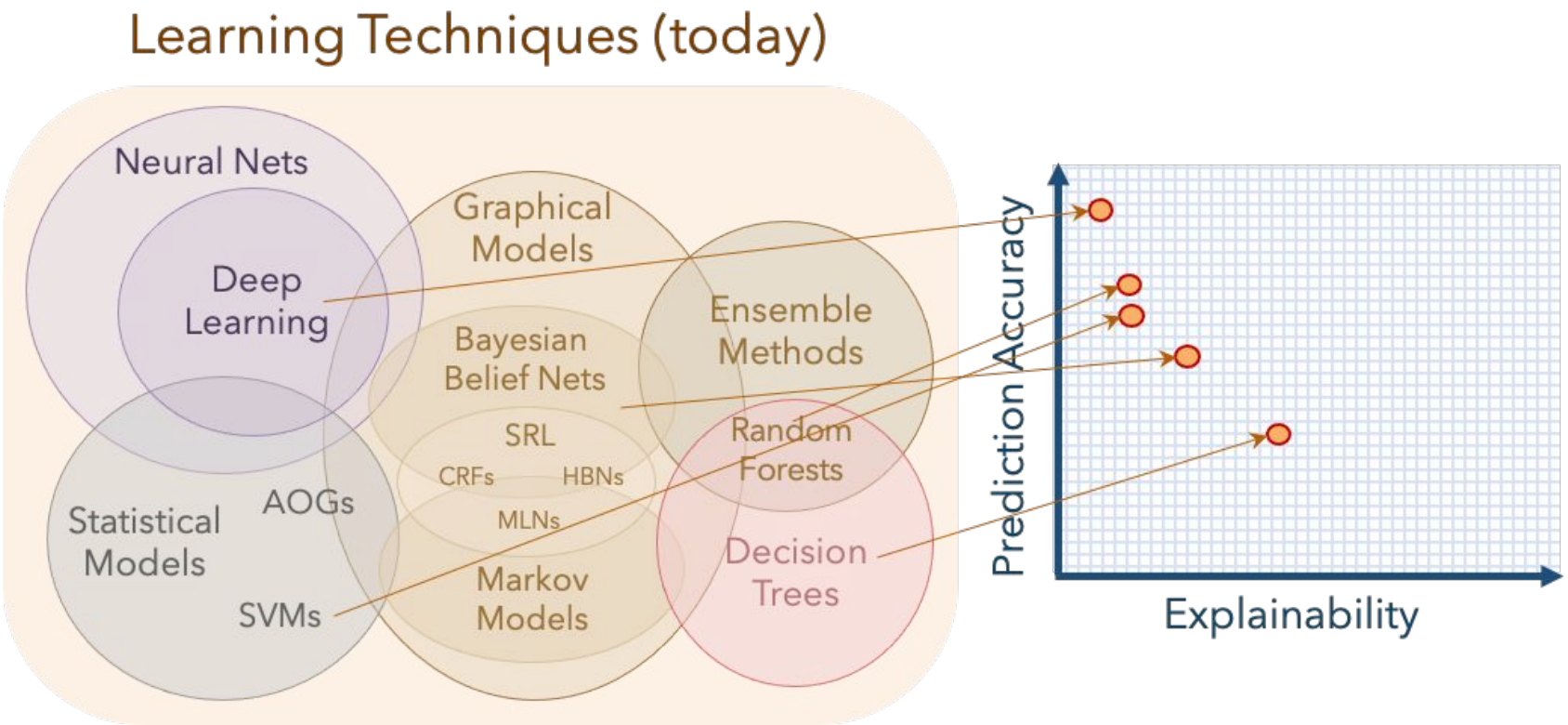


Introduction

From Data Perspective

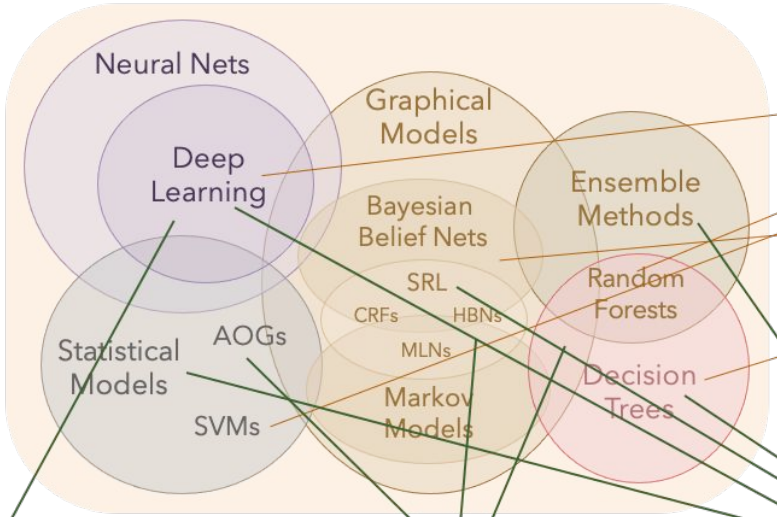
ML

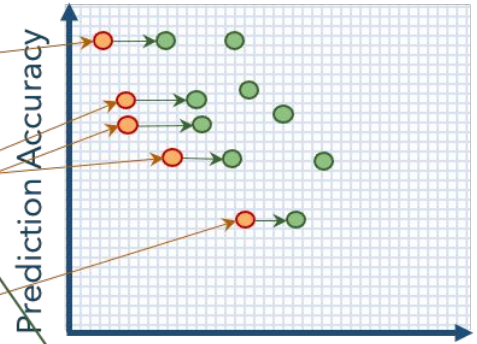
What Models Say?

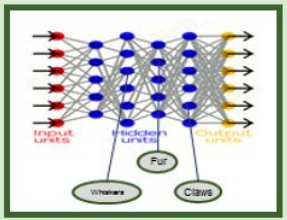


Learning Techniques (today)

We need to create a suite of machine learning techniques that
produce more explainable models, while maintaining a high level of learning performance

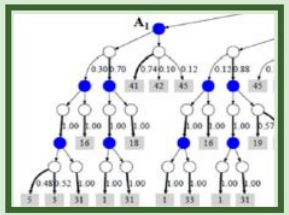






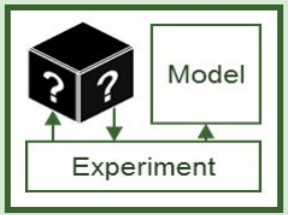
Deep Explanation

Modified deep learning techniques to learn explainable features



Interpretable Models

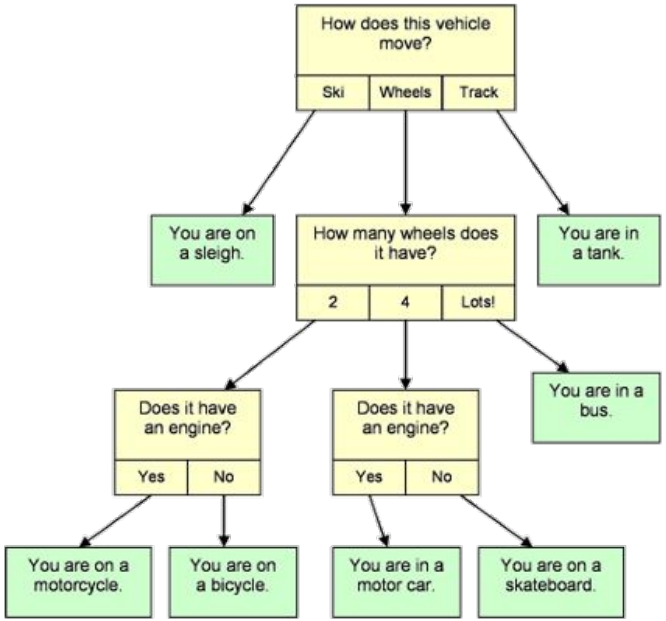
Techniques to learn more structured, interpretable, causal models

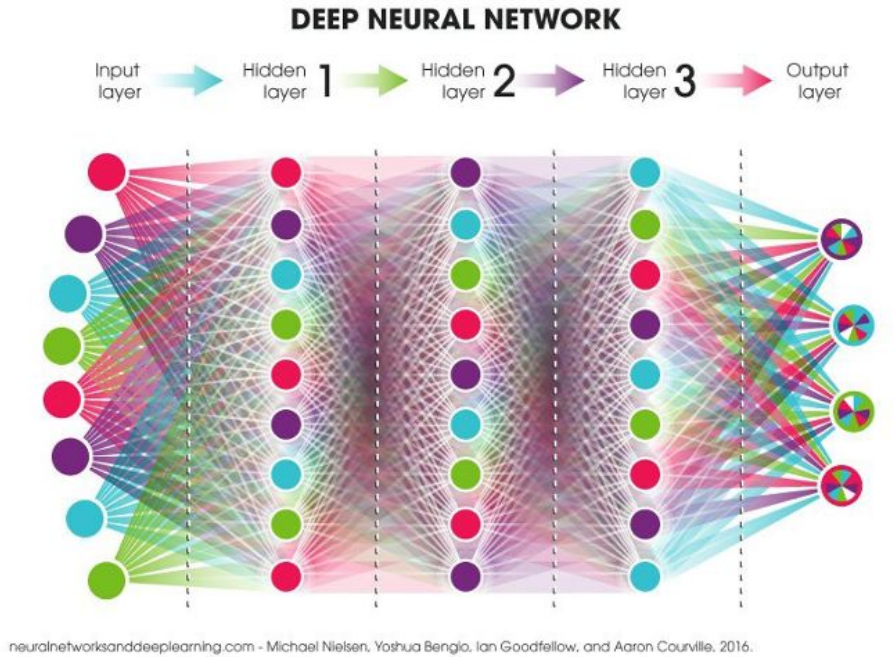


Model Induction

Techniques to infer an explainable model from any model as a black box

What Models Say?



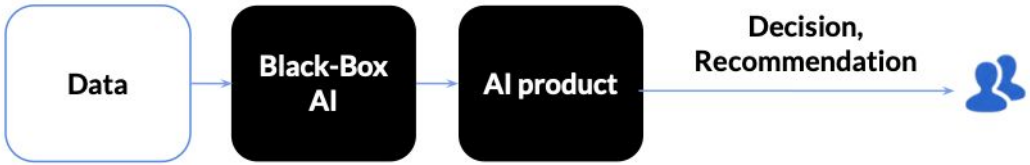


Expert system:
Good for explanations,
not so good for accuracy

Neural nets:
Good for accuracy,
not so good for explanations

Challenges of XAI

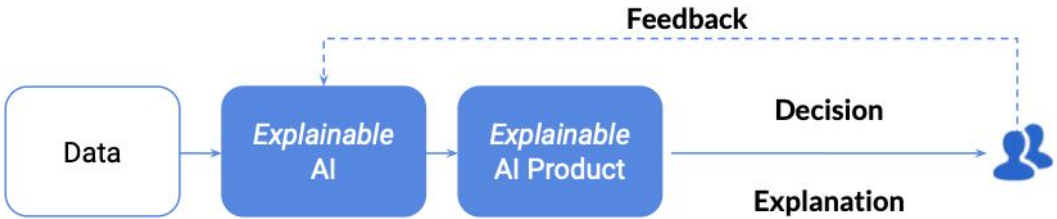
Black Box AI



Confusion with Today's AI Black Box

- Why did you do that?
- Why did you not do that?
- When do you succeed or fail?
- How do I correct an error?

Explainable AI



Clear & Transparent Predictions

- I understand why
- I understand why not
- I know why you succeed or fail
- I understand, so I trust you

Challenges of XAI

1. Confidentiality:
How to ensure that the AI system hasn't learned a biased perspective of the world (or maybe an unbiased perspective of a biased world) based on shortcomings of the training information, model, or goal function? Imagine a scenario where its human creators harbor a conscious or unconscious bias?

2. Complexity:
Sometimes algorithms are well understood but are highly complex.

3. Unreasonableness:
How to confirm that the decisions made are fair if it made based on an AI system?

4. Injustice:
We may understand the ways an algorithm is working; but, we need clarification for how the system is consistent with a legal or moral code

Explainable AI

With Great Power, Comes Great Responsibility
Spiderman's Uncle

Rule of Thumb

Within AI, we use simple models to solve challenging tasks and complex models to solve simple tasks.

Rule of Thumb

Within AI, we use **simple models** to solve **challenging tasks** and **complex models** to solve **simple tasks**.

Simple models are the classic machine learning methods, such as linear classifiers, decision trees, k-nearest neighbors, etc. As a general rule, models you could implement yourself in an afternoon without much googling or math.

Difficult problems are all those tasks we humans need training to solve and some time to think about before we can come up with a good answer. For instance, evaluating the value of a house, reviewing a loan proposal, deciding on a course of action for a patient, etc.

Complex models are all the heavily-numeric methods, such as SVMs and Neural Networks, or, more broadly, nearly all kernel and gradient-based methods.

Simple problems are all the intuitive tasks we solve each day. For example, when you see someone you know, you don't stop to think — you instantaneously recognize (1) it is a person, (2) who this person is, and (3) its facial expression.

When Choosing Model

When choosing a model keep these things in mind :

1. **Interpretability:** how much the model informs about the problem it solves
2. **Explainability:** how able it is to explain the why behind its outputs
3. **Flexibility:** how capable it is of describing complex subjects
4. **Complexity:** how costly it is to be run and trained

Examples :

For instance, when we inspect a **decision tree**, we learn how it solves the problem (**interpretability**), and we can trace which decisions brought an input to a particular output (**explainability**). The deeper a tree is, the more powerful (**flexibility**) and expensive (**complexity**) it will be.

Two Types of Model

Glass-box Models

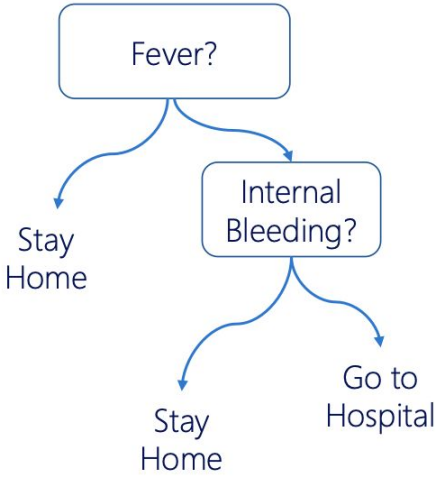
Black-box Models

Two Types of Model

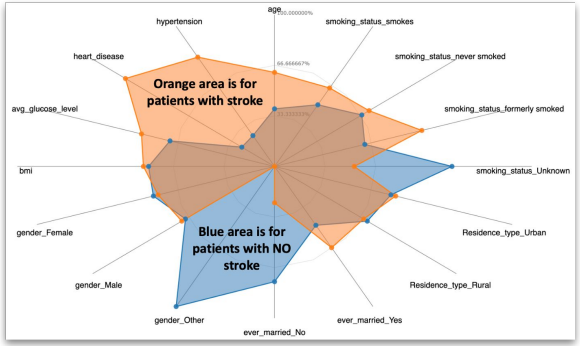
Glass-box Models

Easy to Understand
Easy to Explain
Designed to be Explainable

e.g :
Decision Tree
Rule Based Models
Linear Models



Two Types of Model



Glass-box Models
Explanation Methods

Feature Importance

Data Visualisation

Linear Models

SHapley Additive exPlanations (SHAP)

Heatmap	female	21 to 70	no sibling/spouse	no parents/children	Southampton
1st class	0.0062	0.1174	0.1075	0.1248	0.0964
female		0.0803	0.0803	0.0927	0.1150
21 to 70			0.4561	0.5328	0.4957
no sibling/ spouse				0.6811	0.5340
no parents/ children					0.6131

Two Types of Model

Black-box Models
Explanation Methods

Attribution Model:

Attributes a model's prediction on an input to features of the input

- Ablations
- Gradient based methods
- Score Backpropagation based methods
- SHAP

Local Interpretable Model-agnostic Explanations

Sensitivity Analysis

Partial Dependence

Gradient based methods

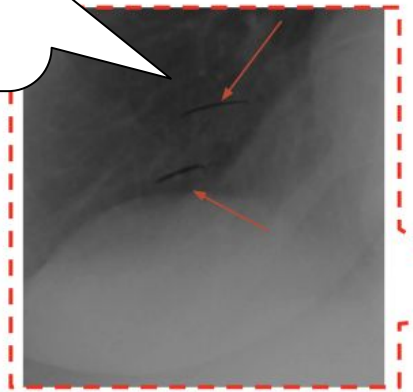
Original image

Integrated Gradients
(for label “clog”)

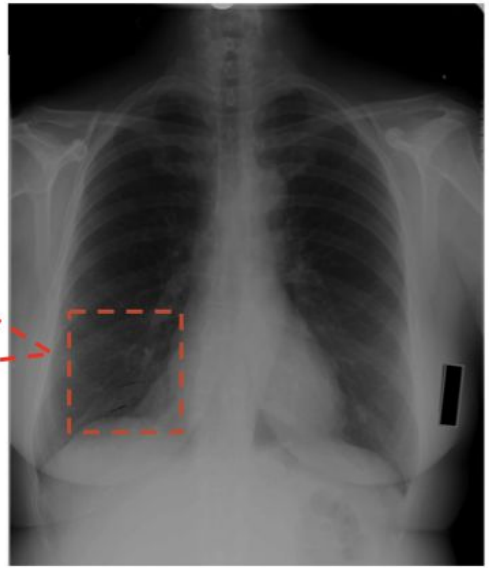
“Clog”

Gradient based methods

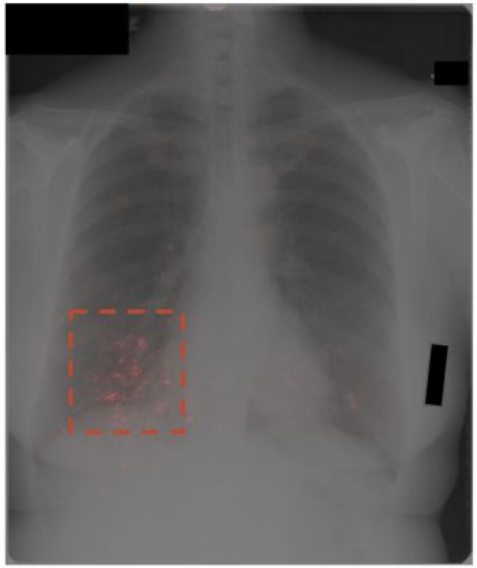
Radiologist put black lines with Market on the image

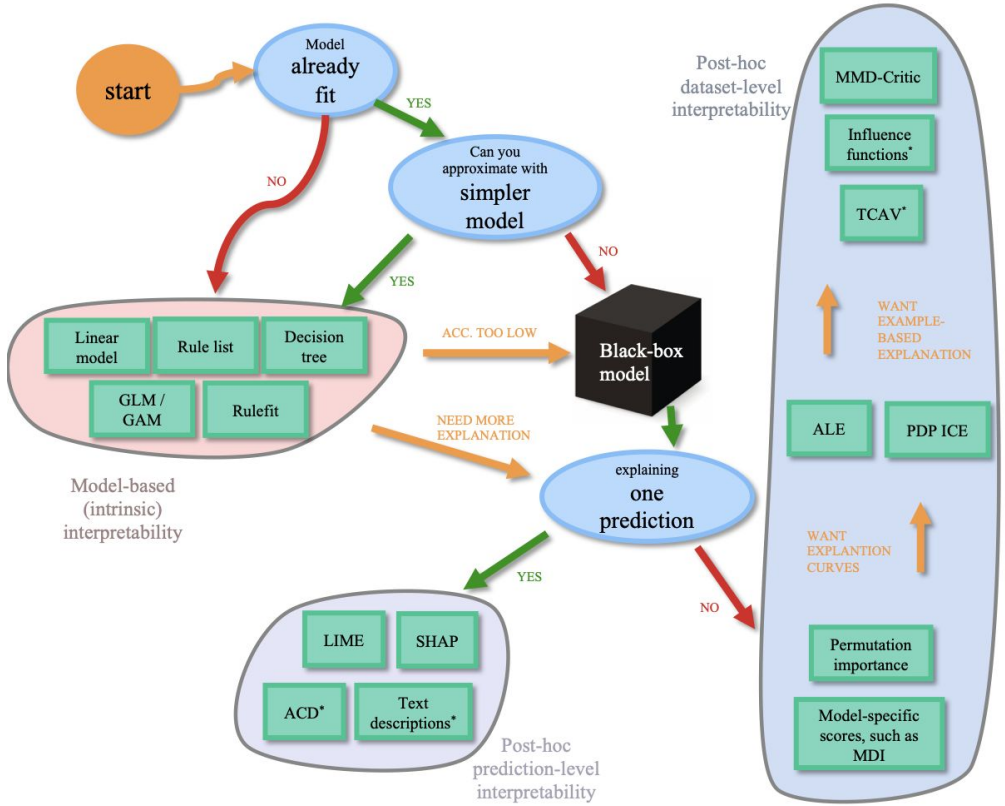


Original image



Integrated gradients (for top label)





* Denotes that a method only works on certain models (e.g. only neural networks)

interpretability cheat-sheet

View on github

Based on this interpretability review and the sklearn cheat-sheet.
More in this book + these slides.

Summaries and links to code

RuleFit – automatically add features extracted from a small tree to a linear model

LIME – linearly approximate a model at a point

SHAP – find relative contributions of features to a prediction

ACD – hierarchical feature importances for a DNN prediction

Text – DNN generates text to explain a DNN's prediction (sometimes not faithful)

Permutation importance – permute a feature and see how it affects the model

ALE – perturb feature value of nearby points and see how outputs change

PDP ICE – vary feature value of all points and see how outputs change

TCAV – see if representations of certain points learned by DNNs are linearly separable

Influence functions – find points which highly influence a learned model

MMD-CRITIC – find a few points which summarize classes

Case Study : Fraud Detection at blu

First Problem:

How can we make sure that a new Transaction is initiated by the Card/Account owner ?

Secondary Problem:

Did the customer change his/her behaviour or did someone impersonate our customer ?

Case Study : Fraud Detection at blu

Feature Engineering :

For each customer :

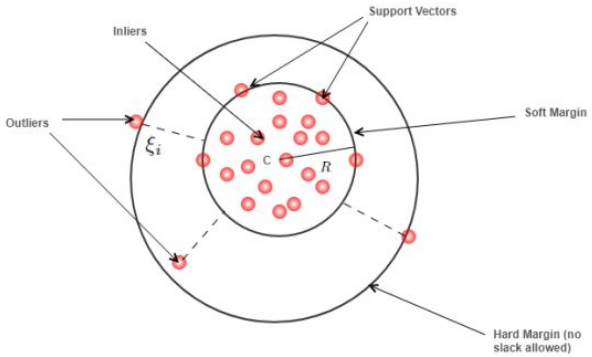
- **How much money they spend daily**
- **Days after last Transaction**
- **How many transactions per day, hour and etc.**
- **Hour of the day / Day of week**
- **Amount**

and so many more ...

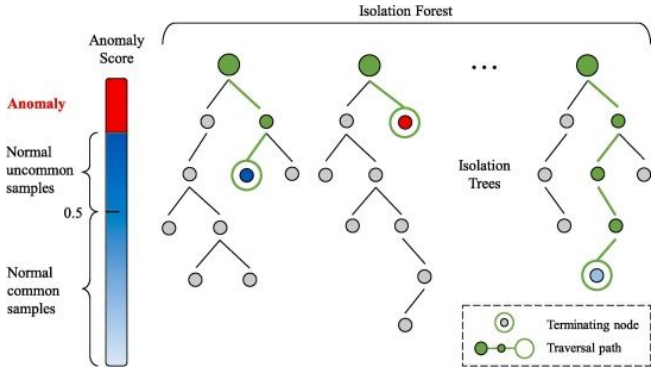
Case Study : Fraud Detection at blu

Our three contenders

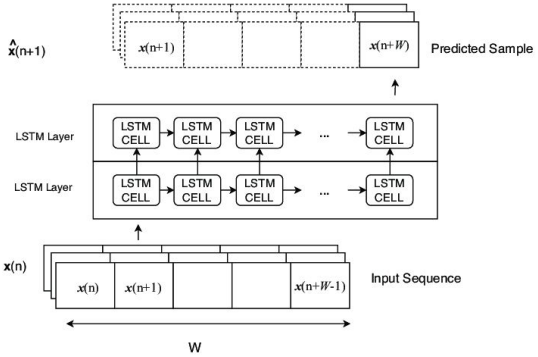
One-Class SVM



Isolation Forest

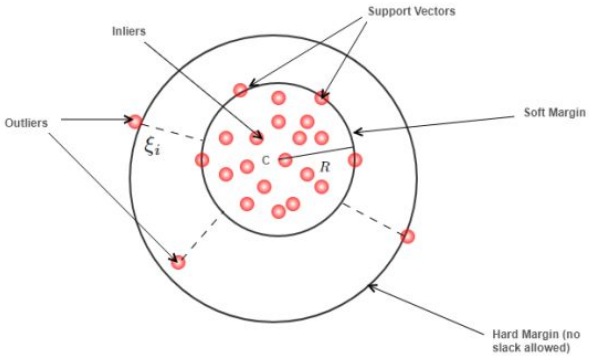


LSTM



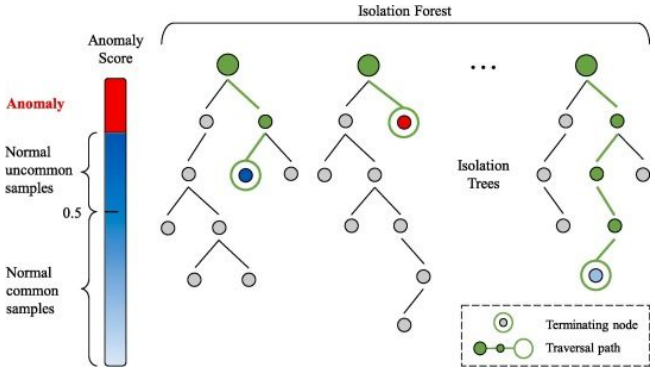
Case Study : Fraud Detection at blu

One-Class SVM



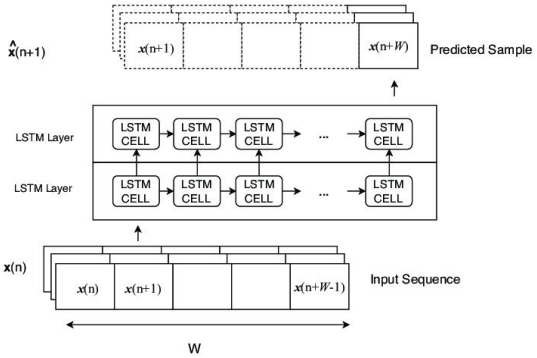
Easy to Interpret
Easy to Understand
Easy to Train
Not Complex Enough
Bad Performance

Isolation Forest



Easy to Train
Fast
Good Performance
No Explanation
Hard to Interpret

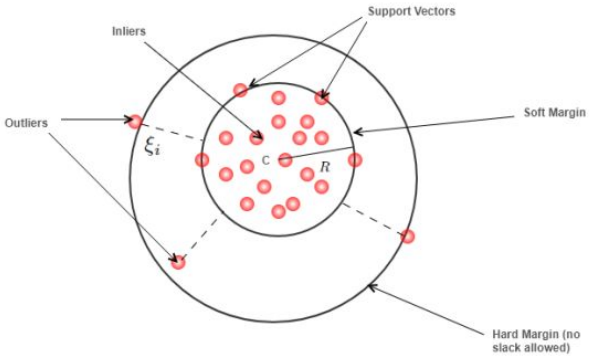
LSTM



Very Good Performance
Fast, Scalable
How do you Explain it ?
Hard to Interpret
Hard to Train

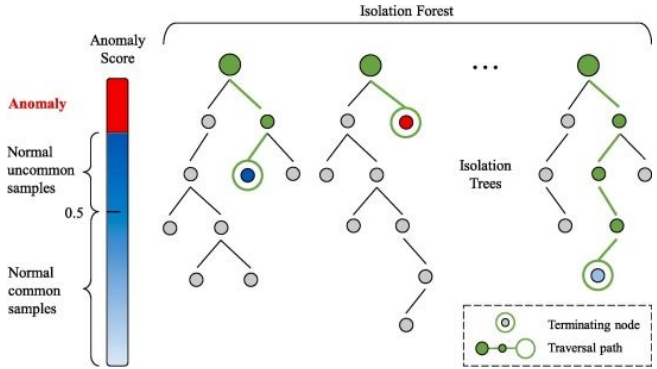
Case Study : Fraud Detection at blu

One-Class SVM



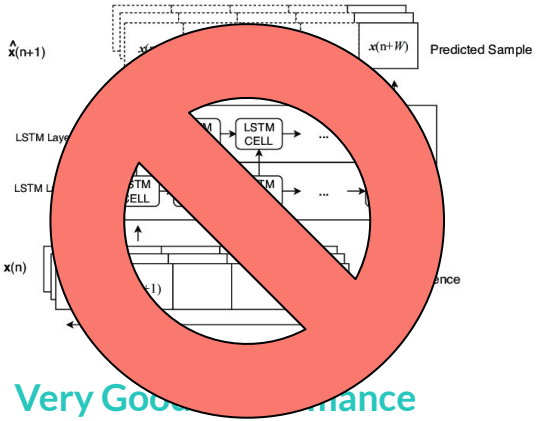
Easy to Interpret
Easy to Understand
Easy to Train
Not Complex Enough
Bad Performance

Isolation Forest



Easy to Train
Fast
Good Performance
No Explanation
Hard to Interpret

LSTM

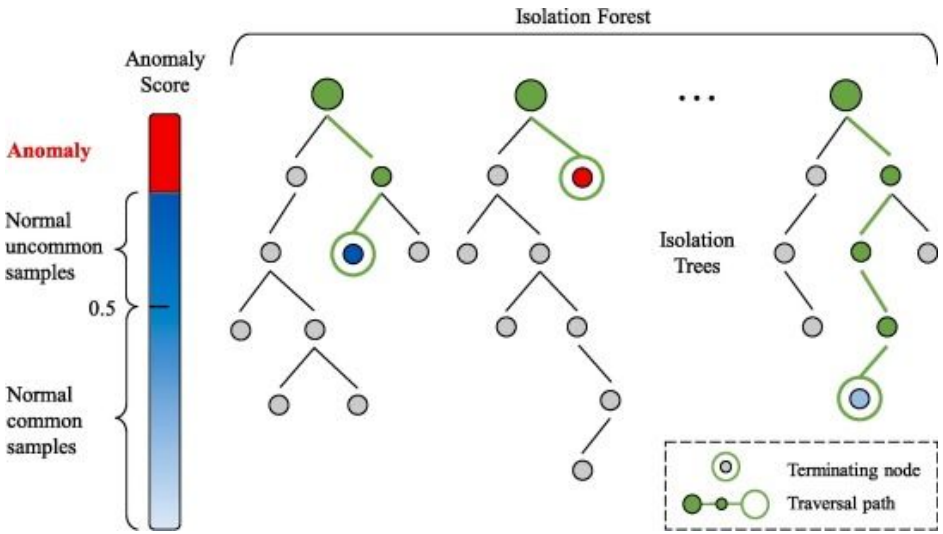


Very Good Performance
Fast, Scalable
How do you Explain it ?
Hard to Interpret
Hard to Train

Case Study : Fraud Detection at blu

Ok...
We trained it. Now what ? Did we solve the problem ?

Isolation Forest



Case Study : Fraud Detection at blu

New Problem :
Customer went to the emergency room in the middle of the night, should we reject his/her transaction ?

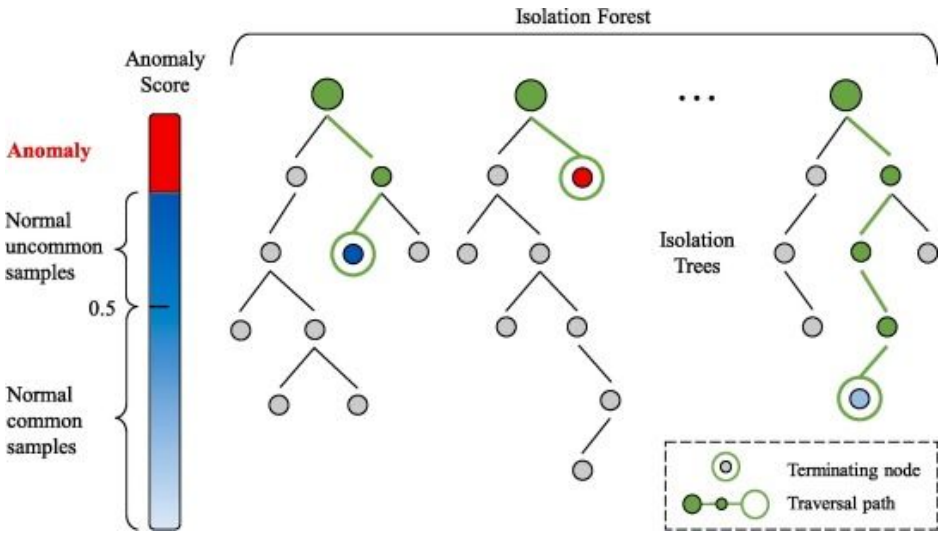
New^2 Problem :
Some of the Transactions randomly get flagged!

New^3 Problem:
Customer wants an explanation of why his transaction was rejected

New^4 Problem:
Stakeholders want our guarantee that no fraudulent transactions get pass the system

New^5 Problem:

Isolation Forest

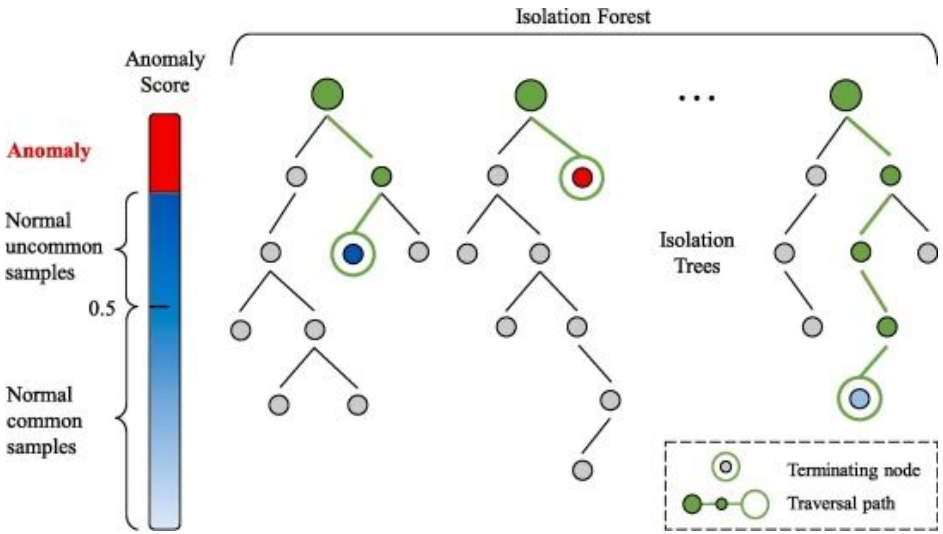


Case Study : Fraud Detection at blu

We need a way to understand Model's Behaviour ?

SHAP to the rescue !

Isolation Forest



Case Study : Fraud Detection at blu

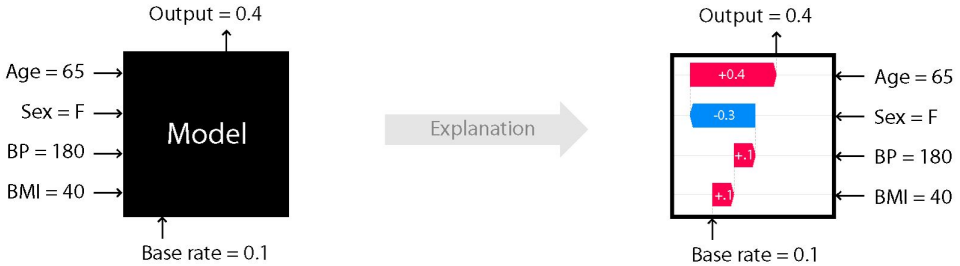
Define a coalition game for each model input X

- Players are the features in the input
- Gain is the model prediction (output), i.e., gain = F(X)

Feature attributions are the Shapley values of this game

Challenge: Shapley values require the gain to be defined for all subsets of players
What is the prediction when some players (features) are absent?

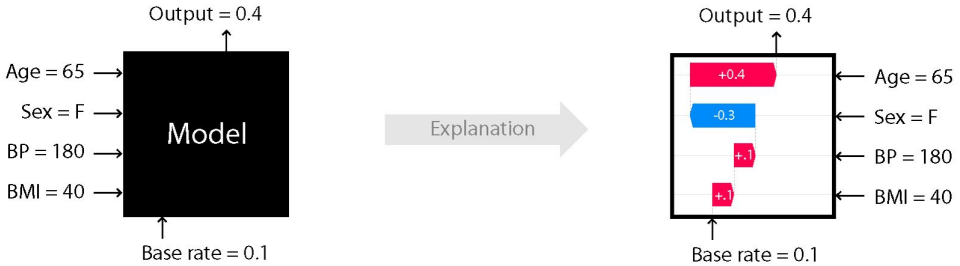
SHAP



Case Study : Fraud Detection at blu

Key Idea:
Take the expected prediction when the (absent) feature is sampled from a certain distribution.

SHAP

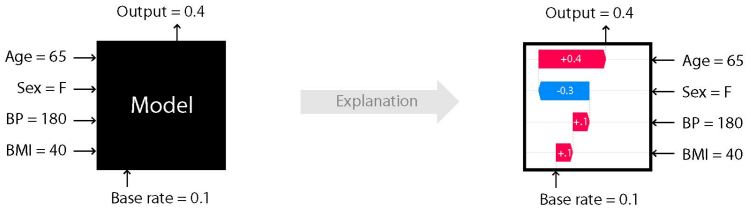


Case Study : Fraud Detection at blu

New Problem :
Customer went to the emergency room in the middle of the night, should we reject his/her transaction ?
Possible Resolution:
Add Location Based Features - Where this transaction happened
Change the Product

New^2 Problem :
Some of the Transactions randomly get flagged!
Possible Resolution:
More seasonal data added
Added Category of Transaction

SHAP



Final Notes

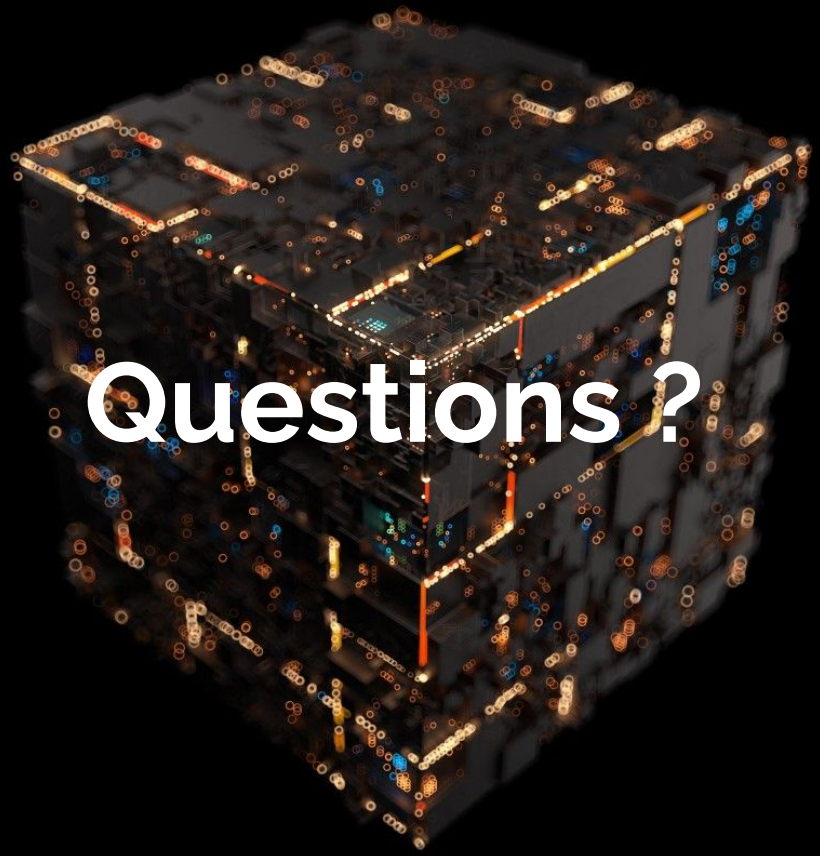
What to do next ?

1. Keep you Data Team on their toes:
If you are a decision-maker, always ask your data scientist or vendor for explanations of how the model makes decisions. As with almost everything in life, the best model and explanation option usually depends on the situation.

2. Don't use Neural Nets as your First method:
Neural Nets require a massive amount of Data and very good infrastructure to train. Most of our problems don't need this much fire-power !

3. Computer Vision tasks:
When dealing with images and videos, we usually have no other option than CNNs and Vision Transformers (ViTs).

4. Find out Whys, Not only Hows :
Make sure that you understand why your model does not perform well ! Don't just use another model.


Questions ?