Mechatronic Engineering program:
Python for machine learning and data science

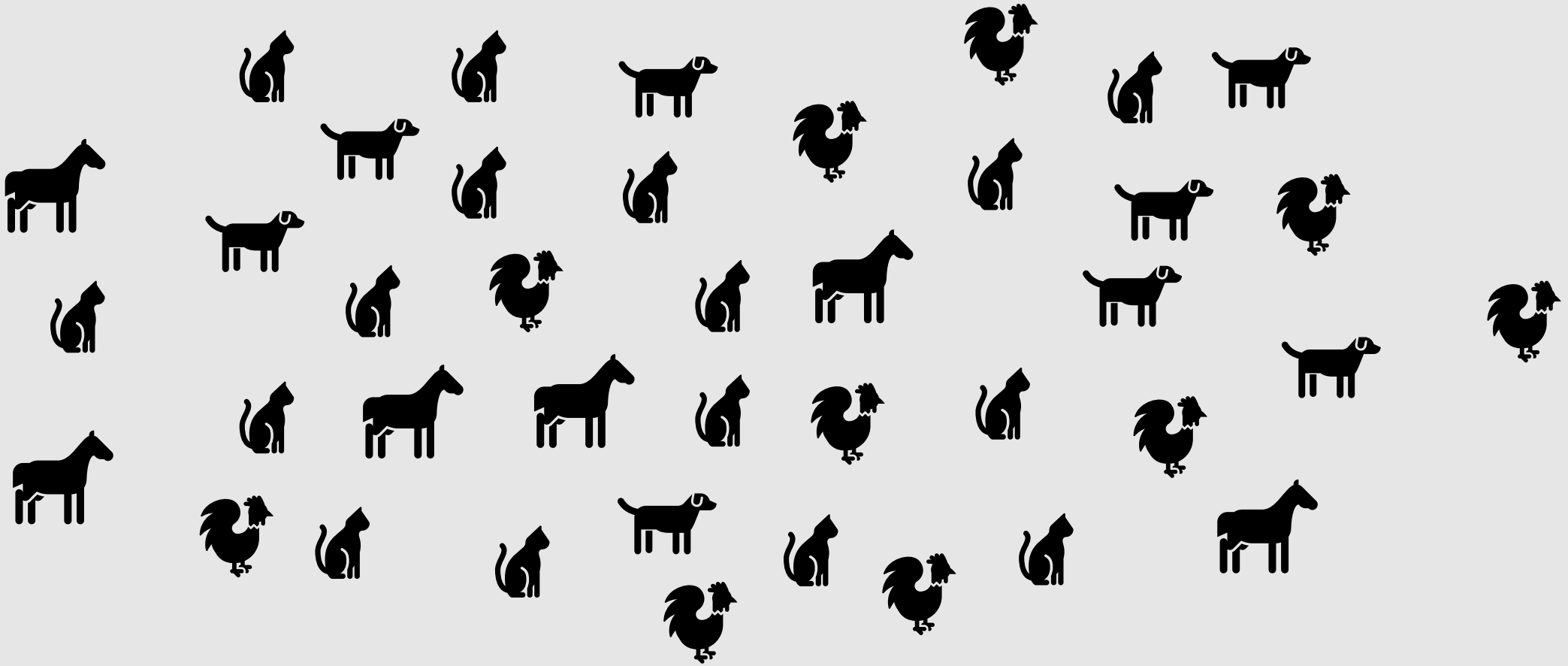
Results interpretation
Models improvement

Ziemowit Dworakowski

AGH University of Krakow

(some) measures of classifiers' efficiency

Consider a classifier for recognizing cats from other animals



(some) measures of classifiers' efficiency

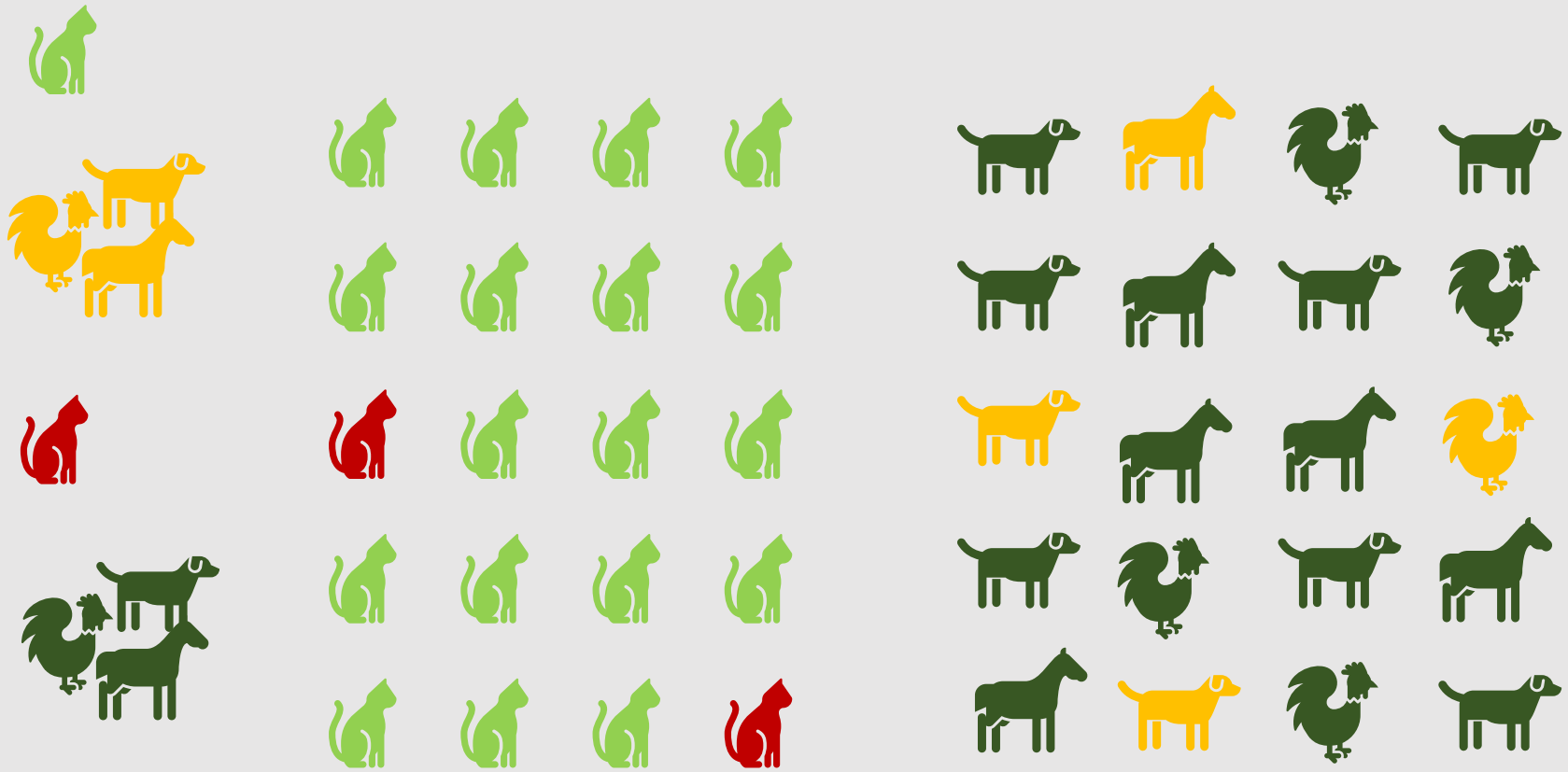
Lets order animals and test our classifier on them.
We are assuming, that classifier „targets" one class. Say: ***cats***
So it answers a question „Is this a ***cat?****". The answer can be* ***Positive*** *or* ***Negative***

We have 18 ***True Positives:***

4 ***False Positives:***

2 ***False Negatives:***

16 ***True Negatives:***

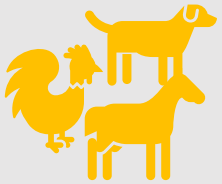


(some) measures of classifiers' efficiency

These will form different quality metrics

True Positives: 

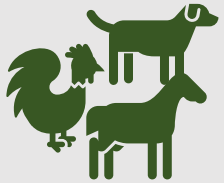
Samples correctly assigned a target class label

False Positives: 

Samples incorrectly assigned a target class label (not having a property to which the classifier is tuned)

False Negatives: 

Samples incorrectly assigned as not having target class label (having a property to which the classifier is tuned but missed in classification)

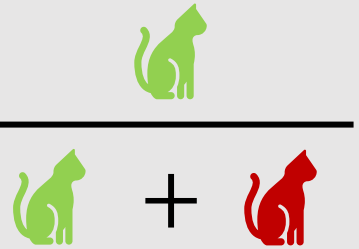
True Negatives: 

Samples correctly identified as not having a target class label

(some) measures of classifiers' efficiency

True Positive Rate (TPR): (Sensitivity, Recall)

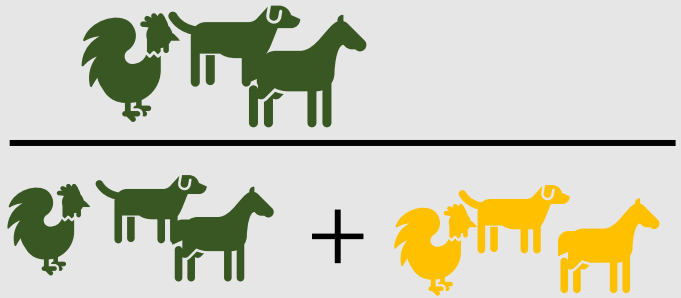
$$\frac{TP}{TP + FN}$$



How many cats are we correctly detecting on average out of all cats?
*How **Sensitive** will we be to cats in a cat-crowded room?*
*How well can we **Recall** cat properties to recognize them?*

True Negative rate (TNR): (Specificity)

$$\frac{TN}{TN + FP}$$



How many other animals can we correctly recognize as not-cats?
*Can we ignore everything that is not a cat knowing cat-**specific** features?*

False Negative Rate:

$$\frac{FN}{FN + TP}$$

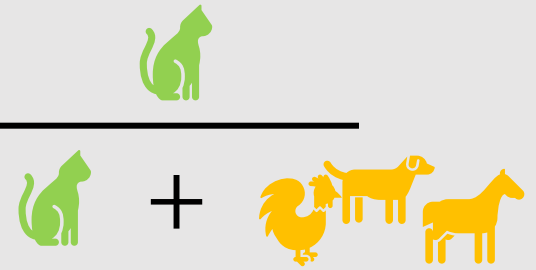
False positive rate (FPR):

$$\frac{FP}{FP + TN}$$

(some) measures of classifiers' efficiency

Precision

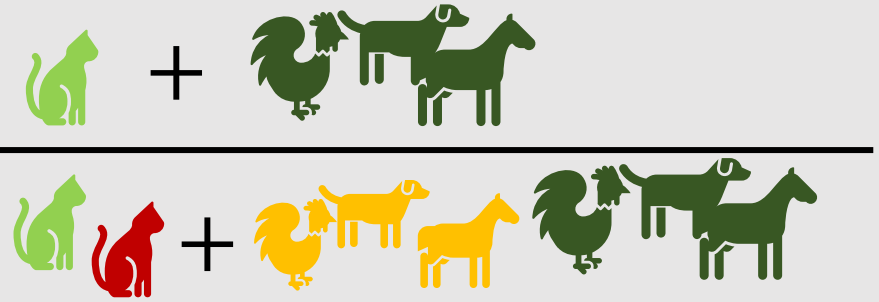
$$\frac{TP}{TP + FP}$$



If our classifier detected something as a cat, how likely it is to be correct?
*If it shows us a positive result, is this **Precisely** a cat and not something that is just cat-like?*

Accuracy

$$\frac{TP + TN}{TP + FP + TN + FN}$$



What is a percentage of correct classifications out of all data?

We are introducing just the most popular ones. There are much more metrics out there, look here for example: https://en.wikipedia.org/wiki/Confusion_matrix

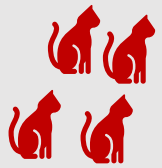
Why are not using just accuracy?

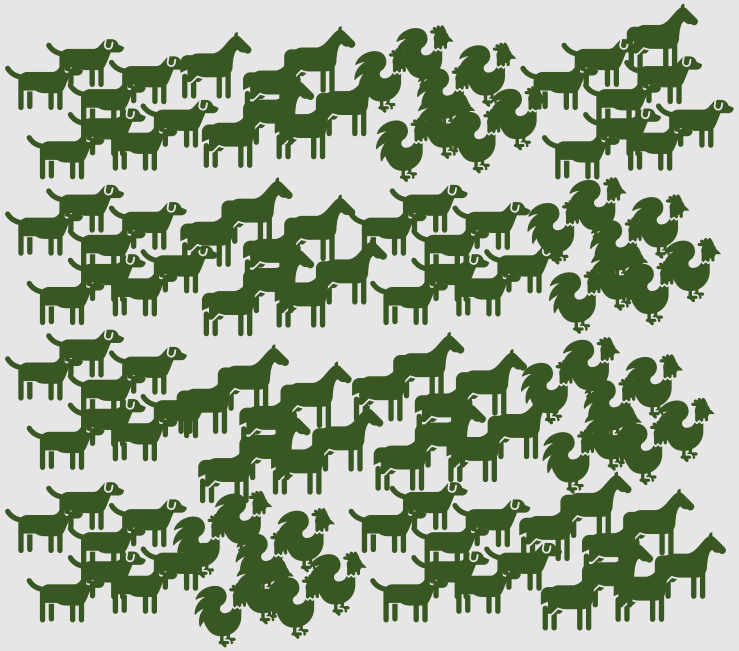


Cat classifier prescription:
„Always answer „its not a cat”

Accuracy…?

Consider dataset where one class has much more samples than the other:





Errors

Correct classifications

Here:

Accuracy *is very high*
Specificity *(**TNR**) is very high*

But…

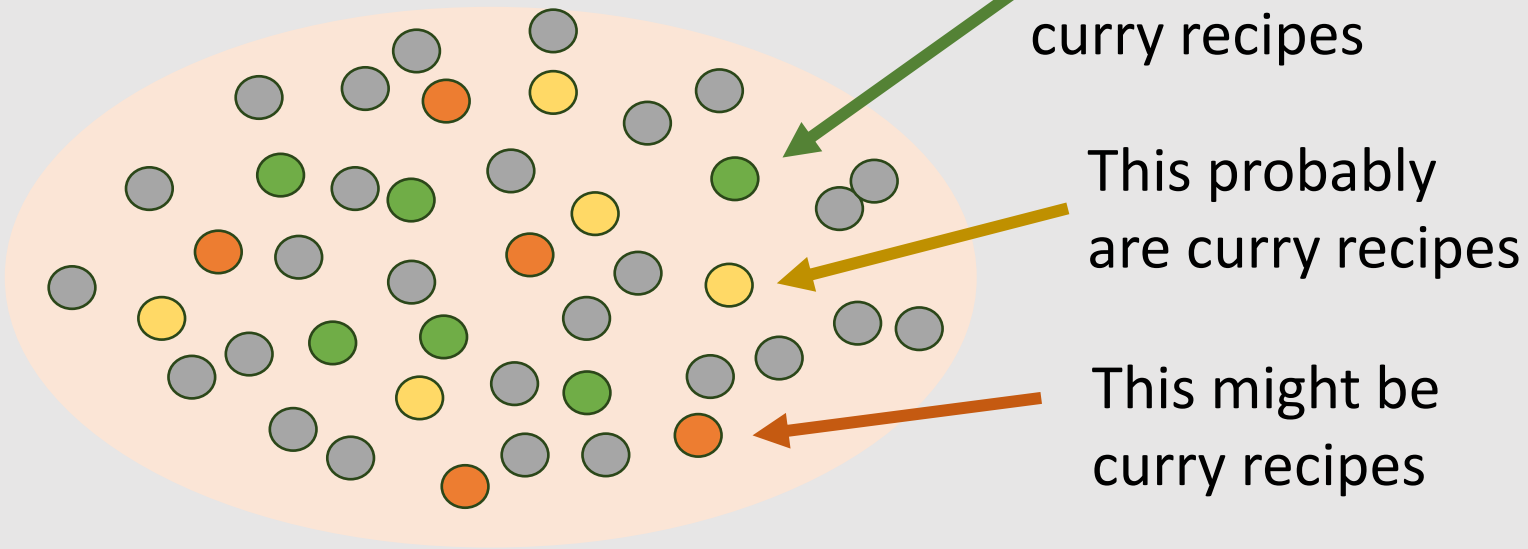
Sensitivity *is 0*

When do we especially need high TPR or high TNR?

Consider a search engine problem:



This guy wants to eat curry for dinner

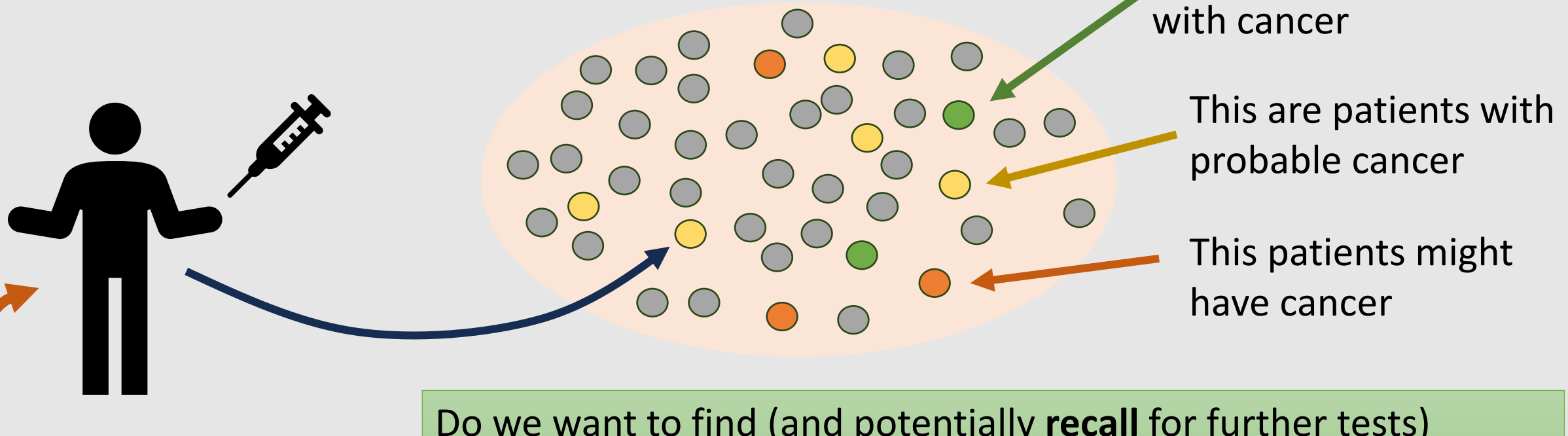


Do we want to **recall** everything that we found (risking showing e.g. curry-reaction videos?
high sensitivity (TPR, Recall)

Do we want to return **precisely** that which is a curry recipe for sure and ignore everything that is not recipe-**specific?**
high precision, high specificity (TNR)

When do we especially need high TPR or high TNR?

Now consider a cancer-detection problem:



This person is given a test to look for potential cancer symptoms

Do we want to find (and potentially **recall** for further tests) everyone that might potentially have cancer – for further confirmation: <u>high sensitivity (TPR, Recall)?</u>

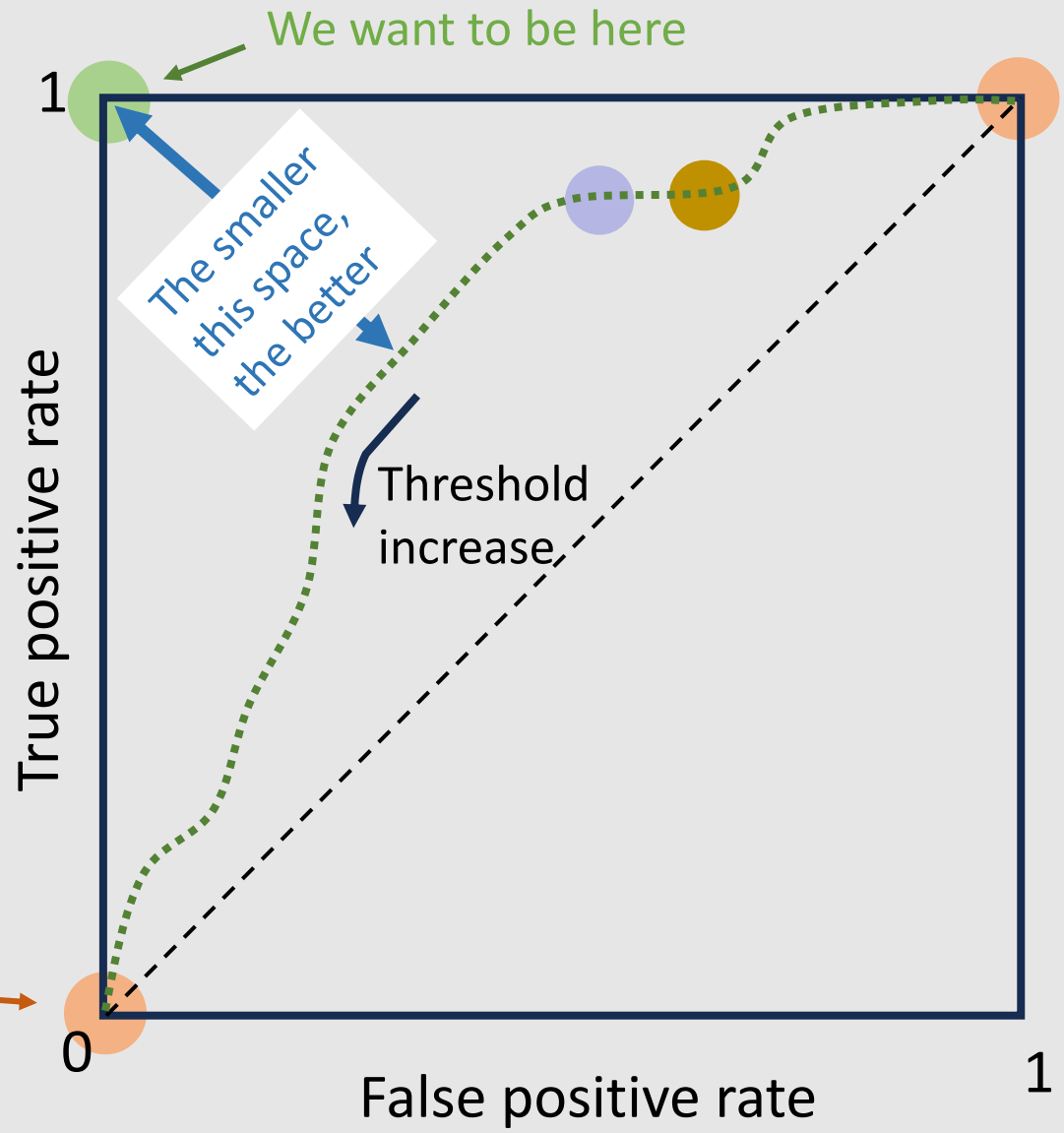
Do we want to treat **precisely** these patients who have very advanced and 100% confirmed cancer and ignore everyone else: <u>high precision, high specificity (TNR)?</u>

Receiver Operating Characteristic (ROC)



Receiver Operating Characteristic (ROC)

This is what our classifier does when we change a threshold



If a threshold is low enough, all samples will be classified positively

If a threshold is high enough, all of samples will be classified negatively

We can also use ROC to pick a reasonable threshold.

For instance point **A** is worse than **B** because at roughly the same TPR we have lover FPR

Receiver Operating Characteristic (ROC)

We can also use it to easily compare two classifiers. If low FPR is more important, orange classifier will be better.

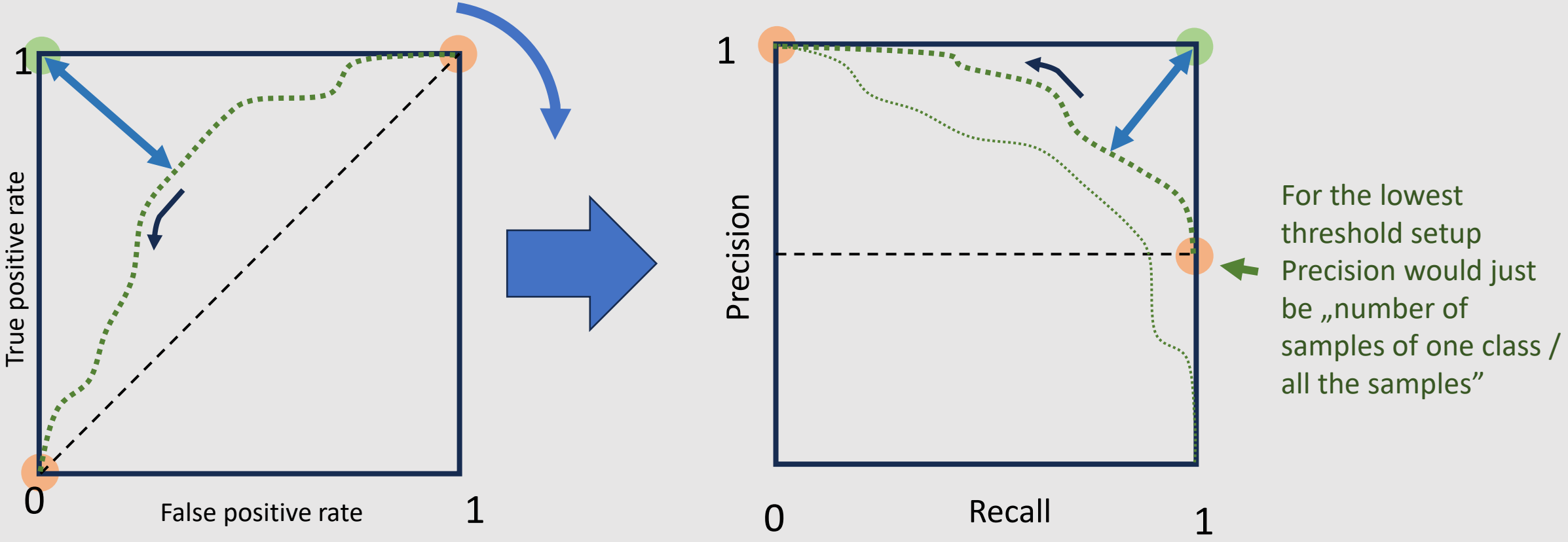
If high TPR is more important, blue one will be better

We can also build a generalized quality metric (threshold-agnostic) called Area Under Curve (AUC)

Note that while AUC is between 0 and 1 it does not straightforwardly scale to accuracy!



Precision-recall (PR) graph



Its not actually rotation! While TPR **is** Recall, Precision **is not** a negative of FPR – but conceptually PR graph serves the same purpose as ROC which is to assess a tradeoff related to focus on increase of TP or decrease of FP

The difference is that ROC focuses on efficiency class-wise (ignoring actual samples' numbers) while PR focuses on actual classifier output. For dissimilar class sizes PR shows what we will actually obtain in practice

Precision-recall (PR) graph
Receiver Operating Characteristic (ROC)

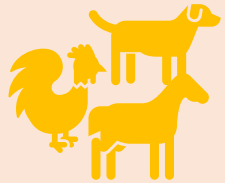
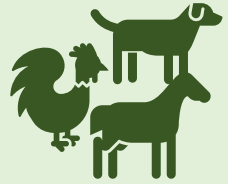


How to use them?

1) We can compare classifiers independently from threshold setup (calculation of area-under-curve)

2) We can use these graphs to configure final output with respect to our actual needs

Confusion matrix

	Actual positive	Actual negative
Predicted positive	TP 	FP 
Predicted negative	FN 	TN 

Confusion matrix

	Actual A (100)	Actual B (200)	Actual C (100)	Actual D (20)
Predicted A	100	10	30	0
Predicted B	0	150	10	0
Predicted C	0	0	60	0
Predicted D	0	40	0	20

Sum of numbers in column is equal to number of samples in class

Now: if we have B sample, how likely we are to correctly classify it?

We can calculate actual probability of B output given B:

$$P(y_B|B) = \frac{\#\, y_B|B}{\#\, y_A|B + \#\, y_B|B + \#\, y_C|B + \#\, y_D|B,}$$

Number of B outputs for B class (TP)

$$P(y_B|B) = \frac{\#\, y_B|B}{\#B}$$

Number of samples in B (TP + FN)

Confusion matrix

	Actual A (100)	Actual B (200)	Actual C (100)	Actual D (20)
Predicted A	100	10	30	0
Predicted B	0	150	10	0
Predicted C	0	0	60	0
Predicted D	0	40	0	20

Sum of numbers in column is equal to number of samples in class

Now: classifier says „A" – what does that mean?

If distribution of testing set reflects reality, we can calculate actual probability of A given A output:

$$P(A|y_A) = \frac{\#\, y_A|A}{\#\, y_A|A + \#\, y_A|B + \#\, y_A|C + \#\, y_A|D,}$$

Number of A outputs for A class (TP)

$$P(A|y_A) = \frac{\#\, y_A|A}{\#\, y_A}$$

Number of A outputs (TP + FP)

Confusion matrix

	Actual A (100)	Actual B (200)	Actual C (100)	Actual D (20)
Predicted A	**100**	**10**	**30**	**0**
Predicted B	**0**	**150**	**10**	**0**
Predicted C	**0**	**0**	**60**	**0**
Predicted D	**0**	**40**	**0**	**20**

We can use confusion matrix for:

- Estimation of probability of correct classification

- Estimation of probability of class presence

- Experiment planning (which classes require more samples)

- Model tuning (which classes require higher accuracy)

What about regression?

Regression is much simpler – you just measure how far your model's prediction is from the actual target:

MAE (Mean absolute error) ← *Easy interpretation (same unit)*

MAPE (Mean absolute percentage error) ← *Easy interpretation (relative)*

MSE (Mean squared error) ← *Focuses on „important errors"*

RMSE (Root mean squared error) ← *Focuses on „important errors" but is also easy to interpret*

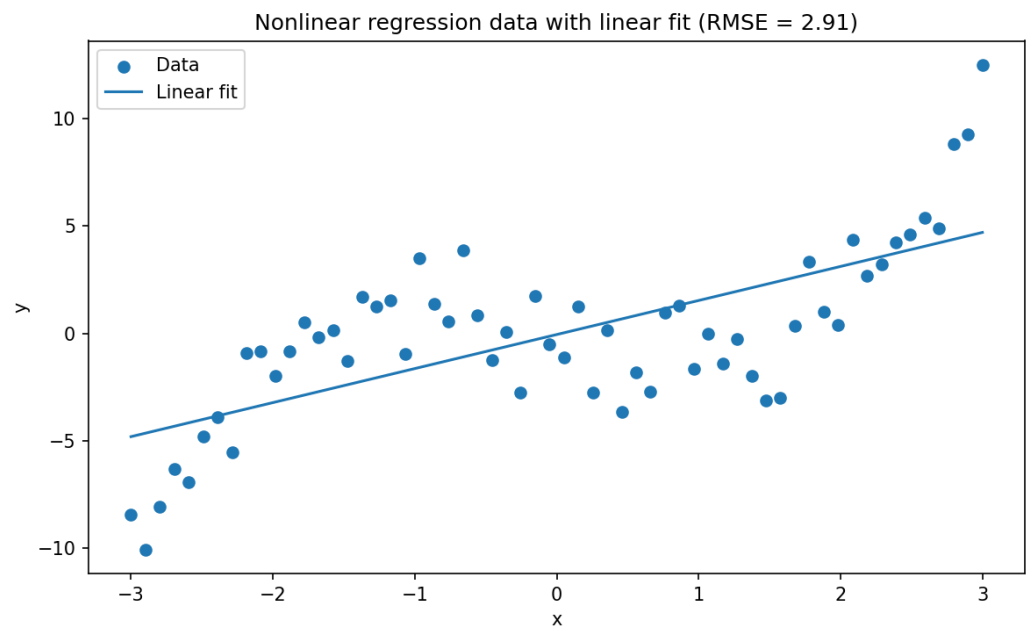
R2 (R Squared, *coefficient of determination*) ← *Allows comparison of models using different datasets*

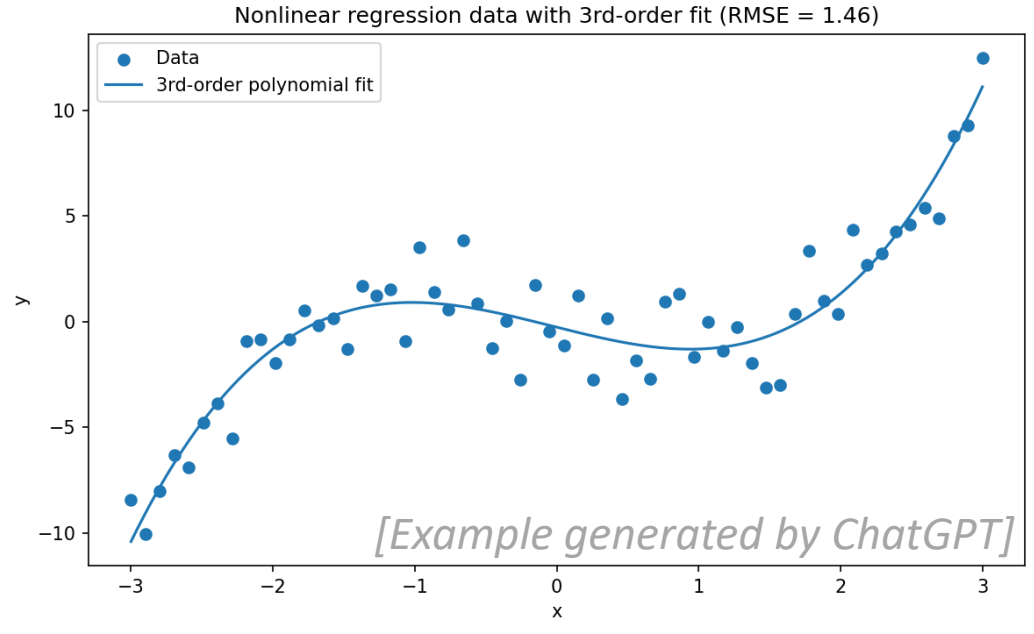
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Squared errors of your model

Squared errors of baseline model (mean)

What about regression?





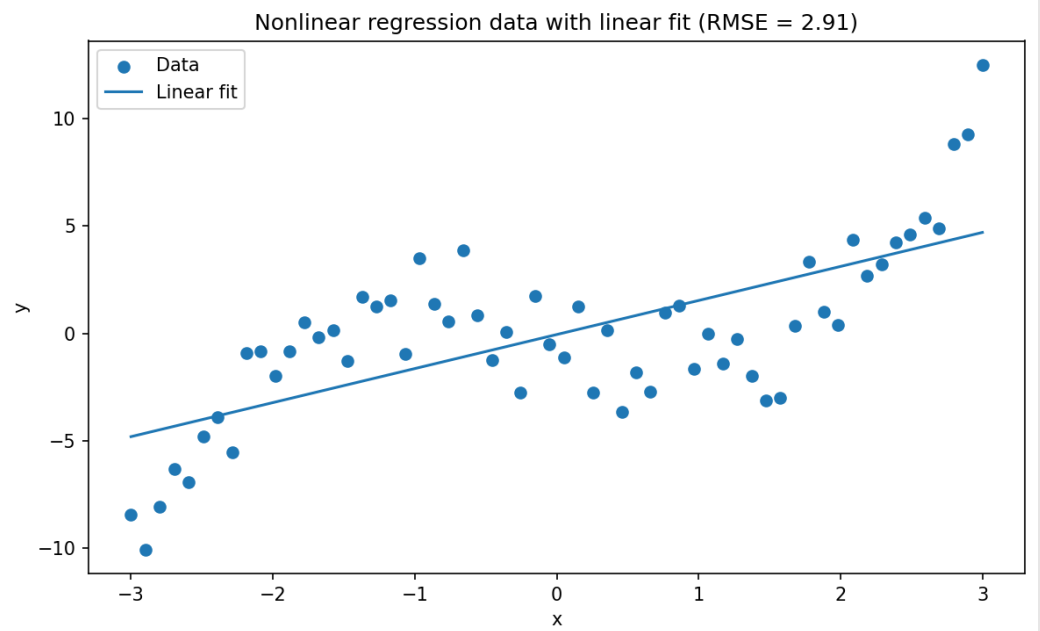
Regression:

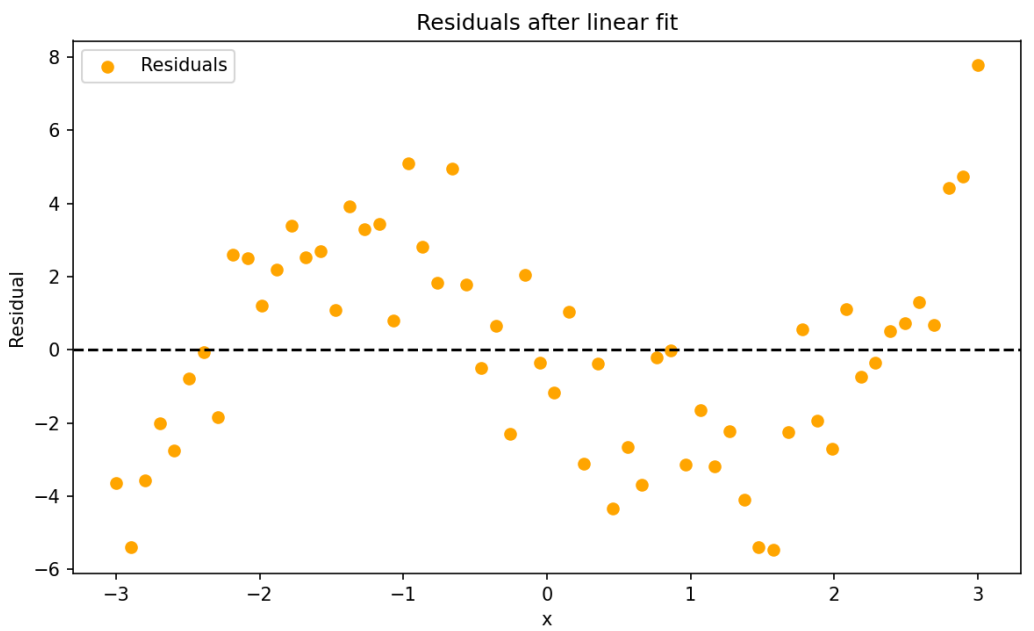
Residual analysis

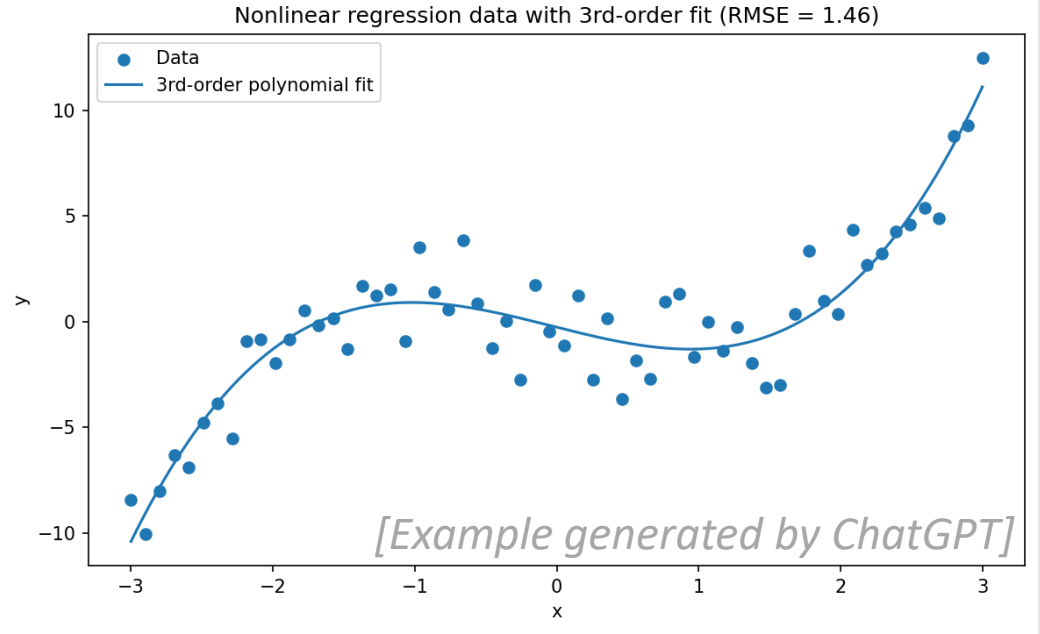
Residuum is a difference between model response and its target

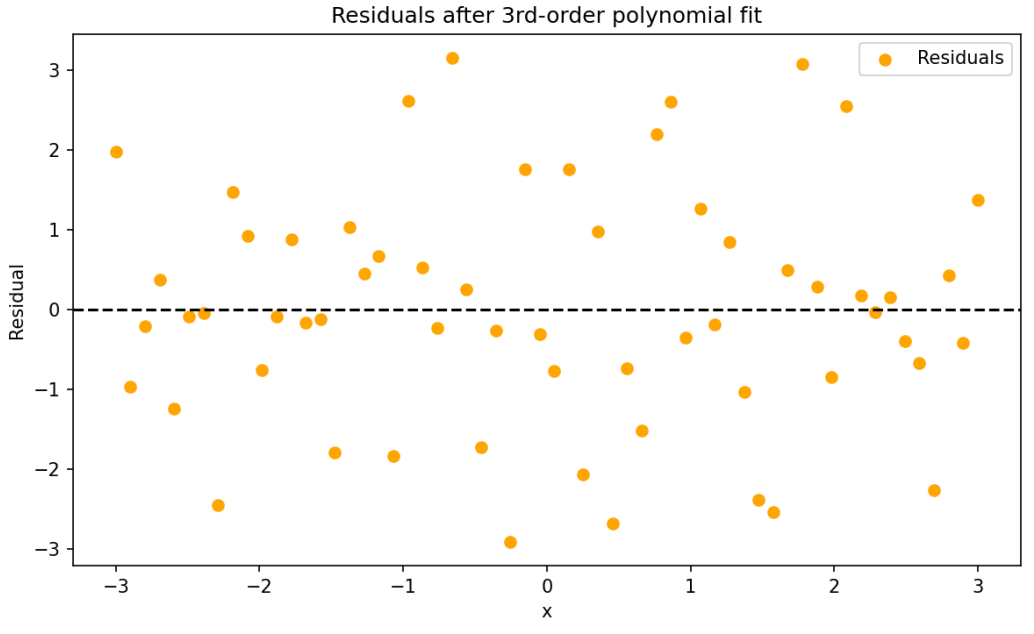
Can be done with respect to various features showing in what cases the model struggles the most
Shows **systematic errors**

What about regression?









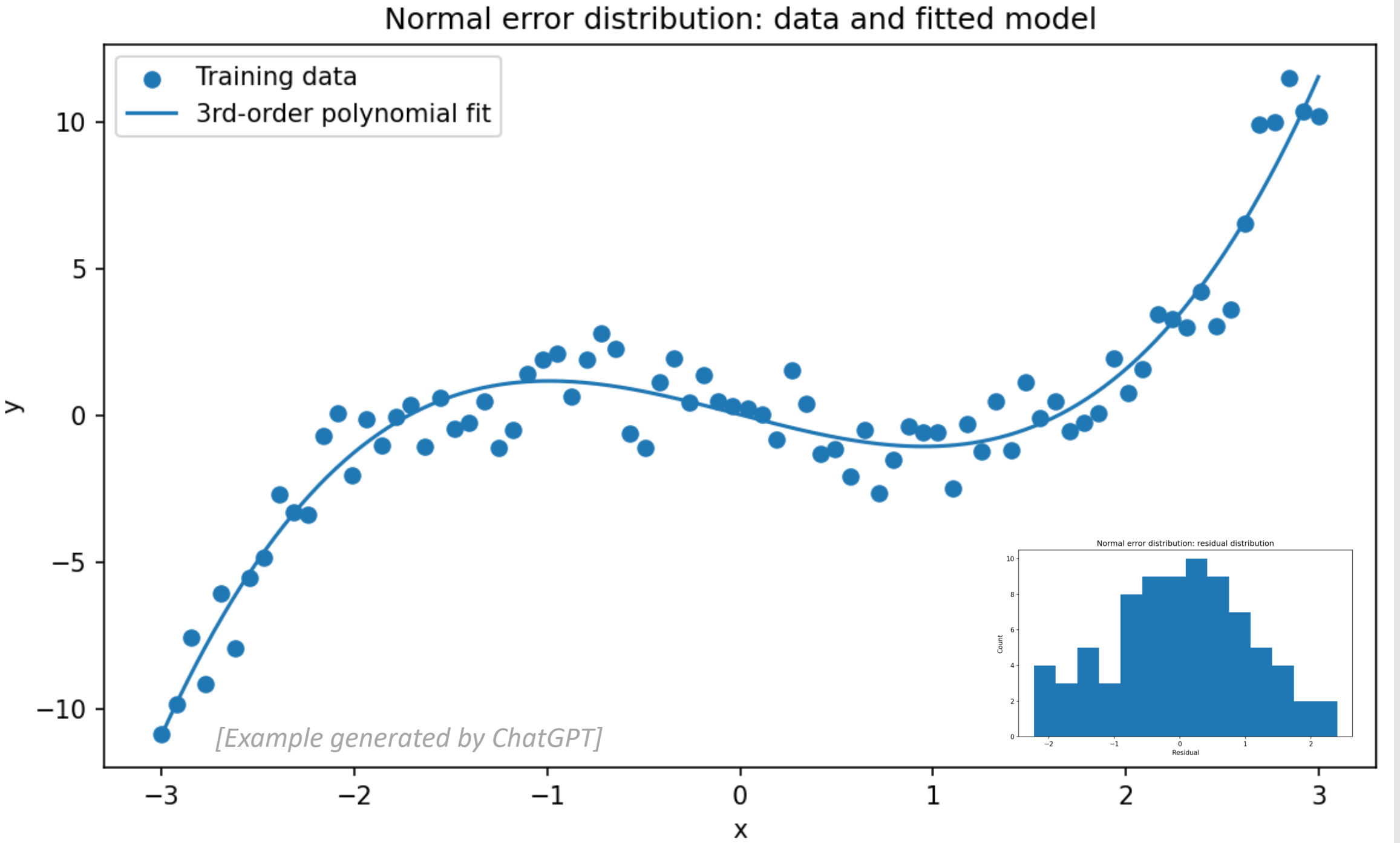
What about regression?

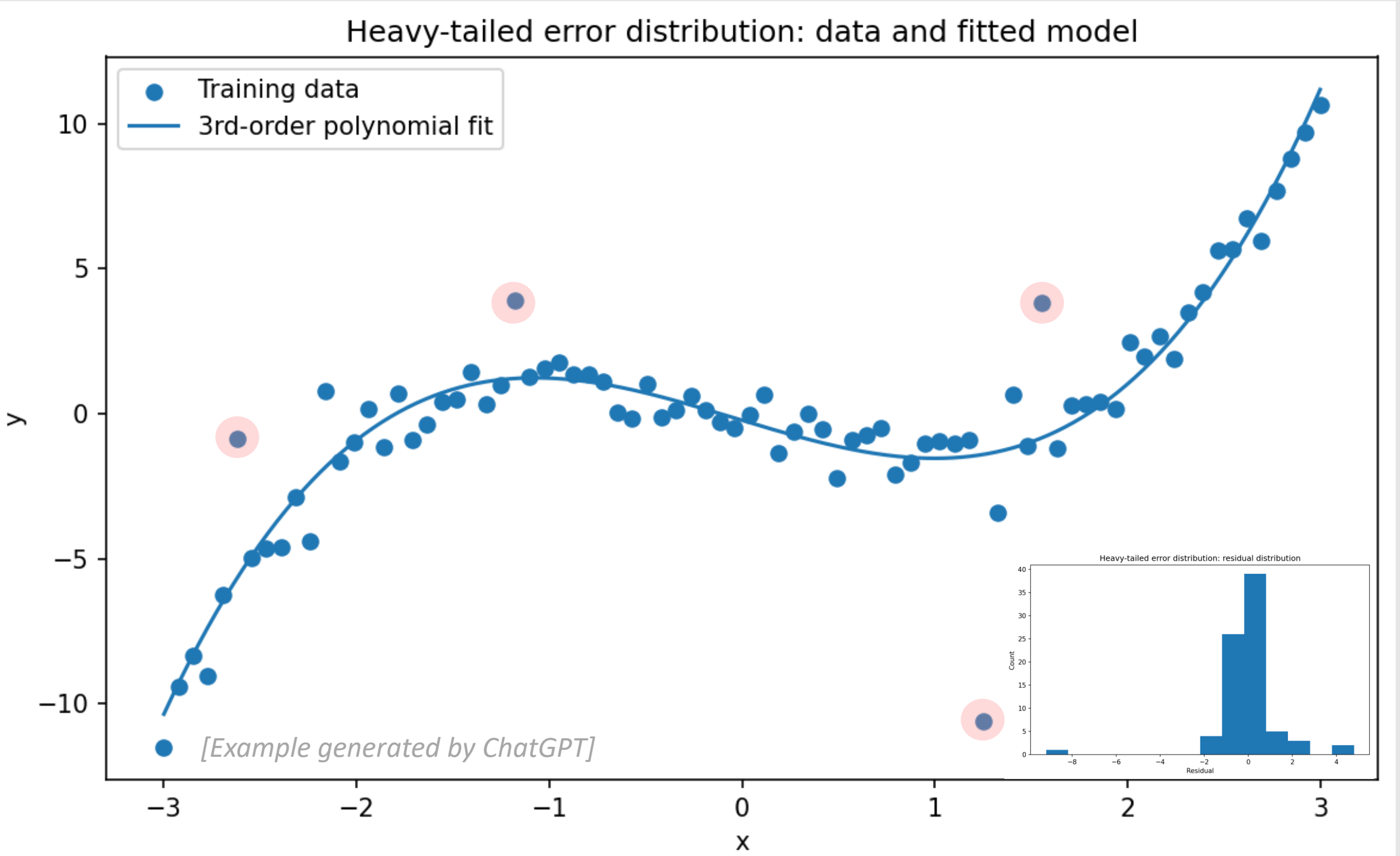
Regression:

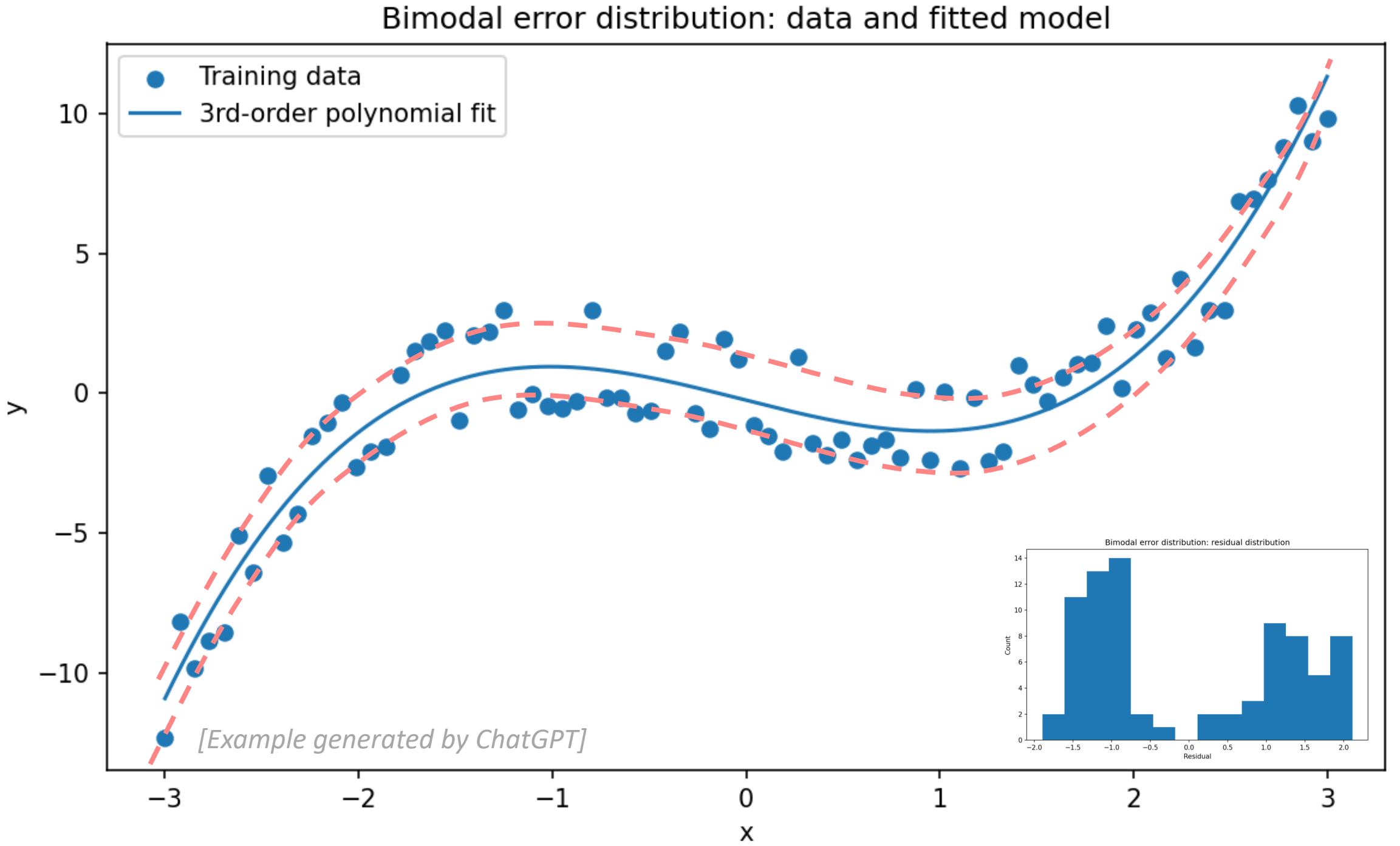
Residual analysis

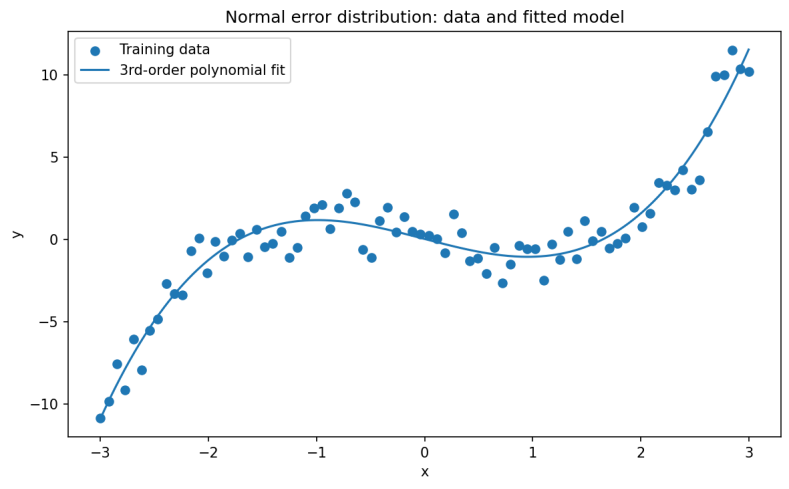
Error distribution

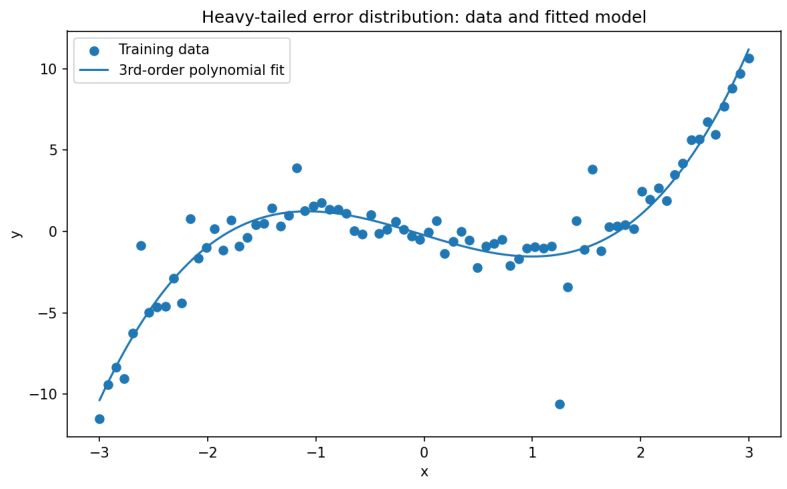
Basically: its the aggregation of the residuum – but can reveal interesting problems with models (or data)

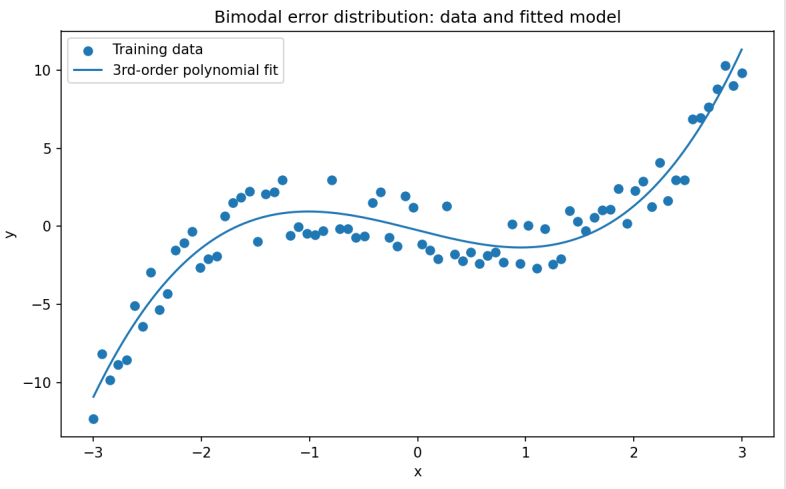

Normal error distribution: data and fitted model
Training data
3rd-order polynomial fit
y
x
[Example generated by ChatGPT]
Normal error distribution: residual distribution
Count
Residual

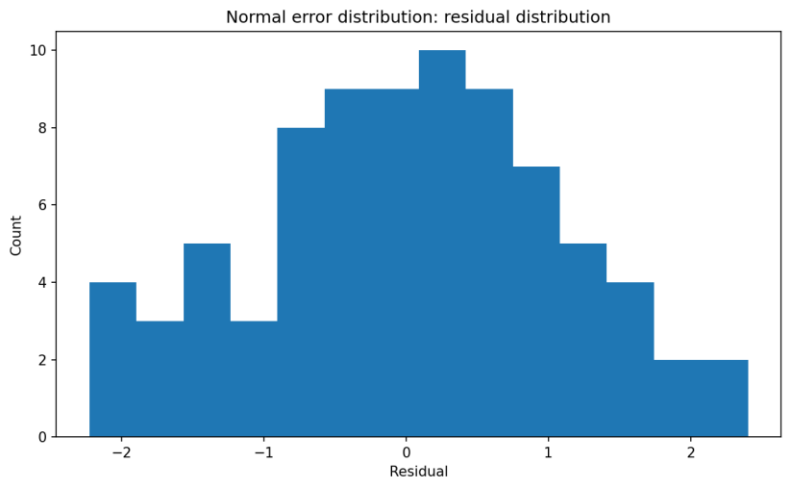

Heavy-tailed error distribution: data and fitted model
Training data
3rd-order polynomial fit
[Example generated by ChatGPT]
x
y
Heavy-tailed error distribution: residual distribution
Count
Residual

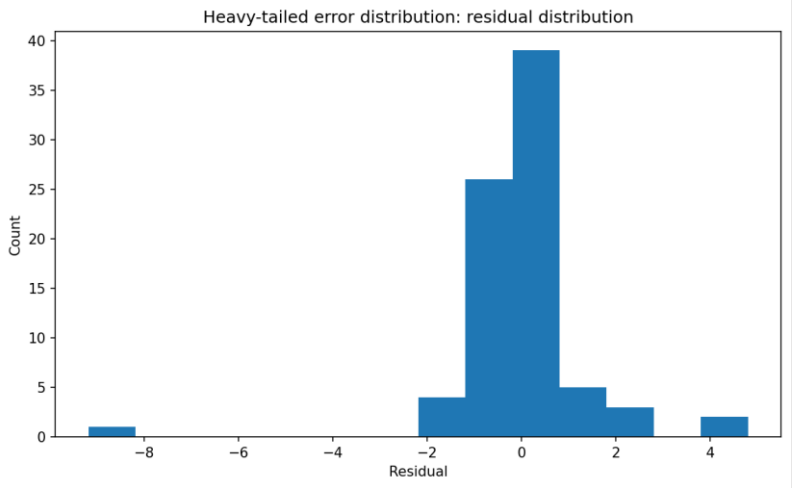

Bimodal error distribution: data and fitted model
Training data
3rd-order polynomial fit
10
5
0
−5
−10
y
−3
−2
−1
0
1
2
3
x
[Example generated by ChatGPT]
Bimodal error distribution: residual distribution
Count
Residual

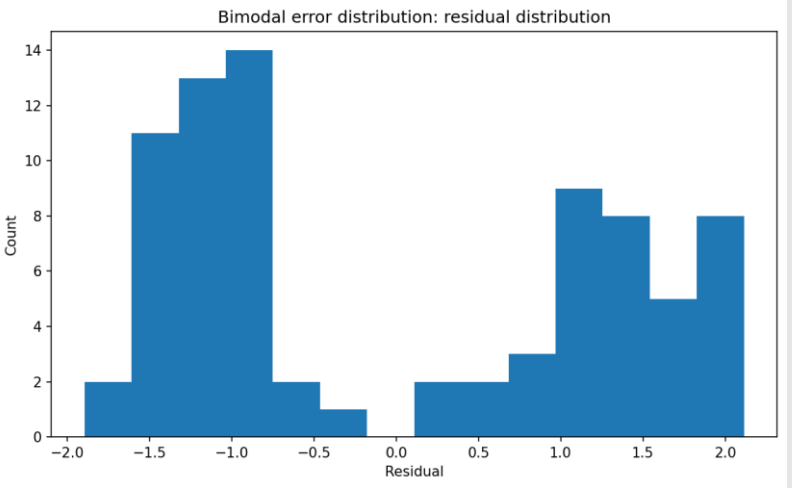










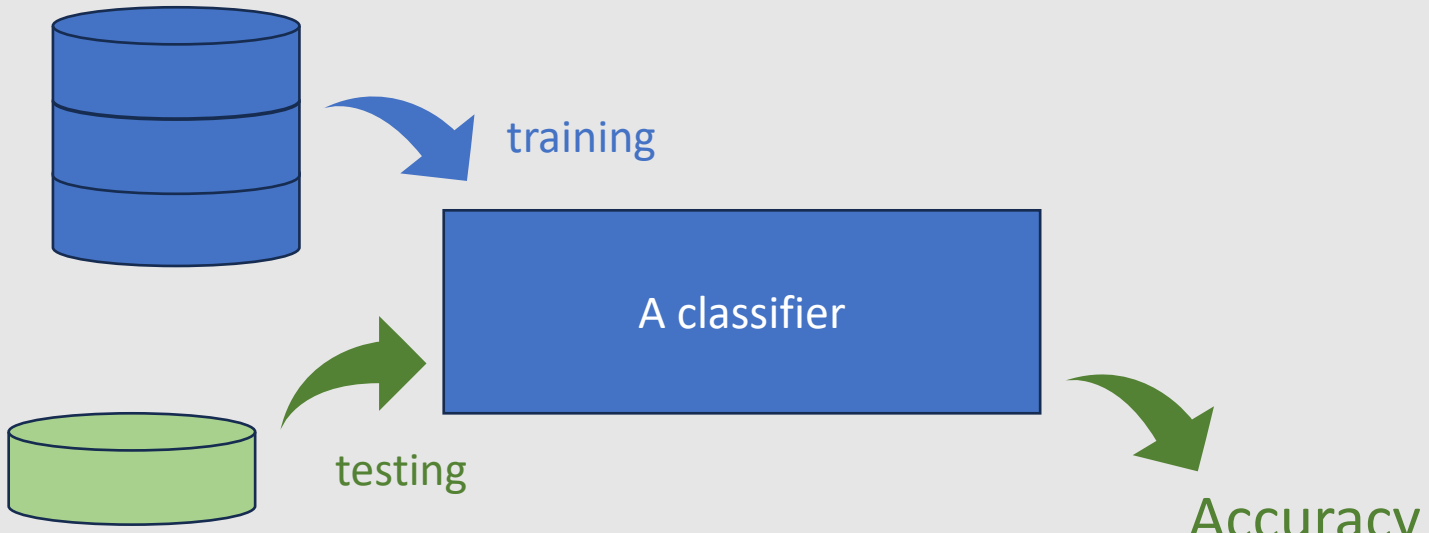


Its OK, maybe improve setup a bit to get better results?

We have huge errors sometimes. I Wonder why?

There's a process underneath we don't recognize yet…

Metrics are estimates!



10 samples → 80%

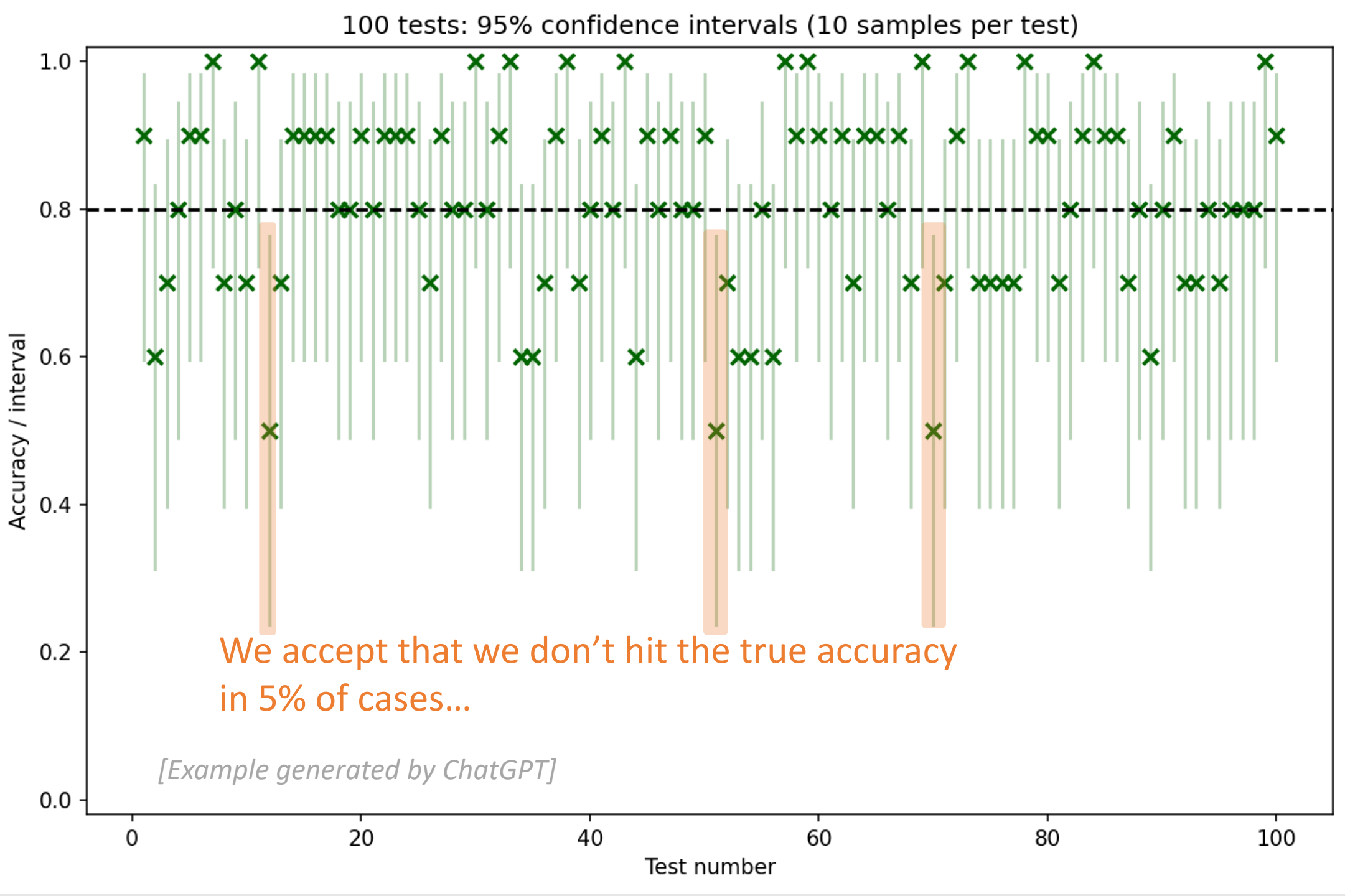
100 samples → 80%

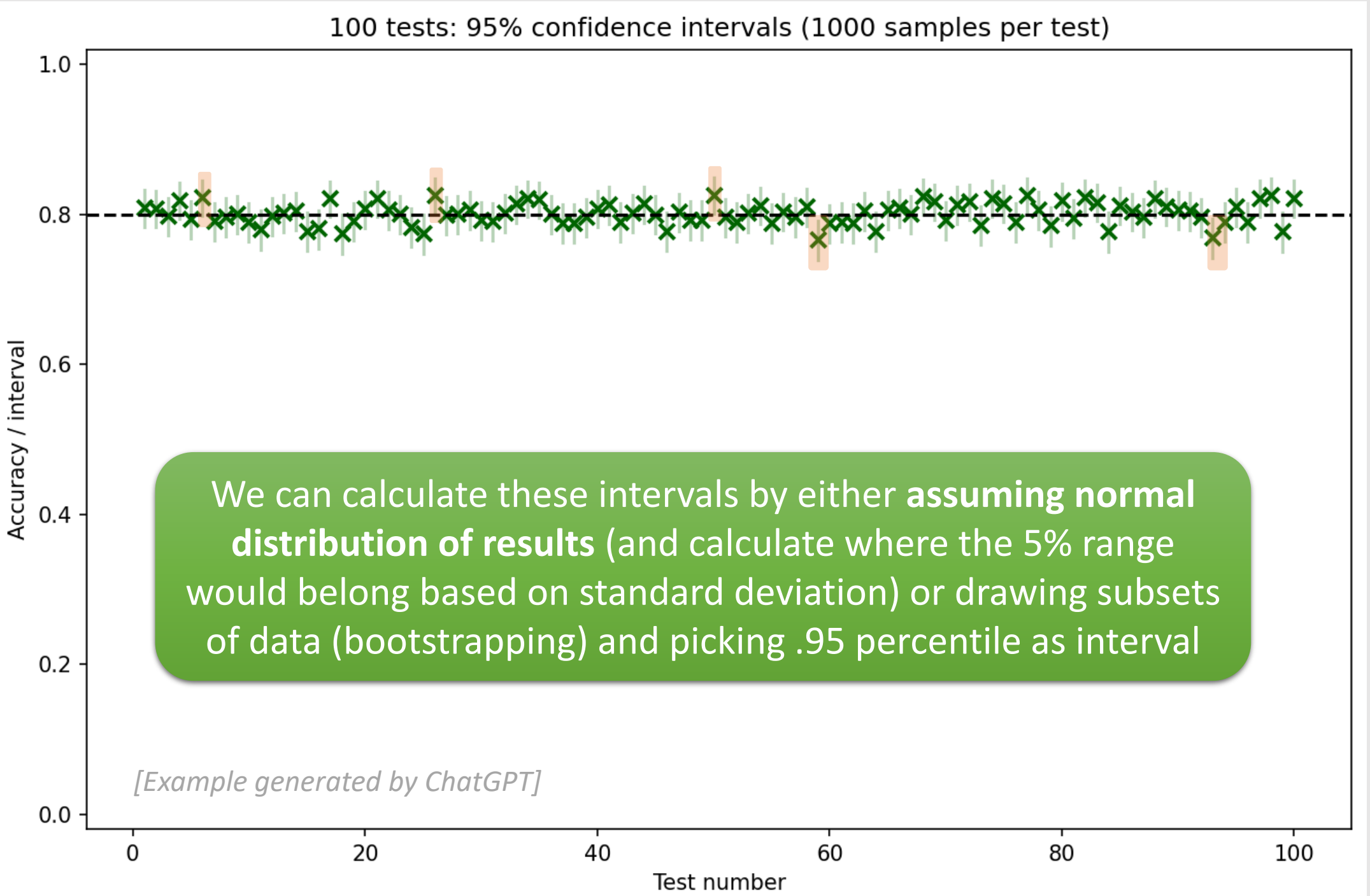
Feels more reliable…

We could ask:

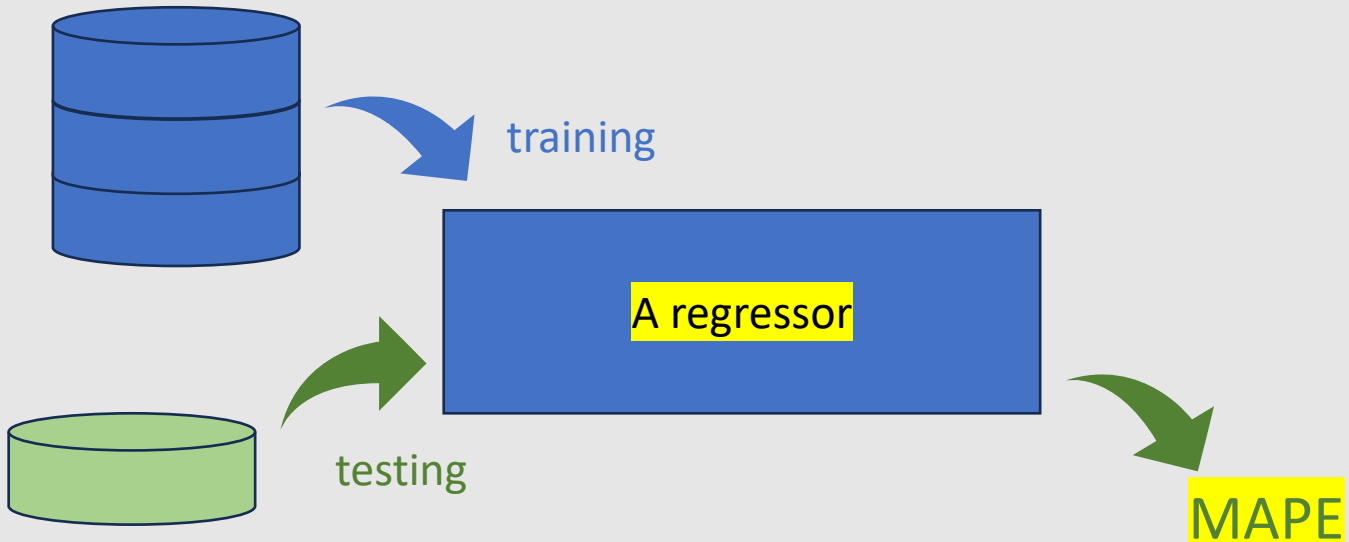
~~**„How likely we are to get this accuracy estimate while true accuracy is below 70%?”**~~

„What accuracy we can assume with 95% confidence?”


100 tests: 95% confidence intervals (10 samples per test)
Accuracy / interval
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
Test number
We accept that we don’t hit the true accuracy in 5% of cases…
[Example generated by ChatGPT]


100 tests: 95% confidence intervals (1000 samples per test)
Accuracy / interval
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
Test number
We can calculate these intervals by either **assuming normal distribution of results** (and calculate where the 5% range would belong based on standard deviation) or drawing subsets of data (bootstrapping) and picking .95 percentile as interval
[Example generated by ChatGPT]

Metrics are estimates!



10 samples → 6%

1000 samples → 6%

Feels more reliable…

We could ask:

~~**„How likely we are to measure this while true MAPE is above 10%?”**~~

„In what range we will be 95% likely to find true MAPE?”

The procedure is similar as before…

Gentle introduction to bayes rule

Imagine that we have a weather forecast classifier
that calculates probability of snow the next day

The classifier is really good, it has overall 95% accuracy,
Snow prediction is as well 95% specific and 95% sensitive
(snow is predicted for 5% of not-snow days and 5% of snow days is not predicted)

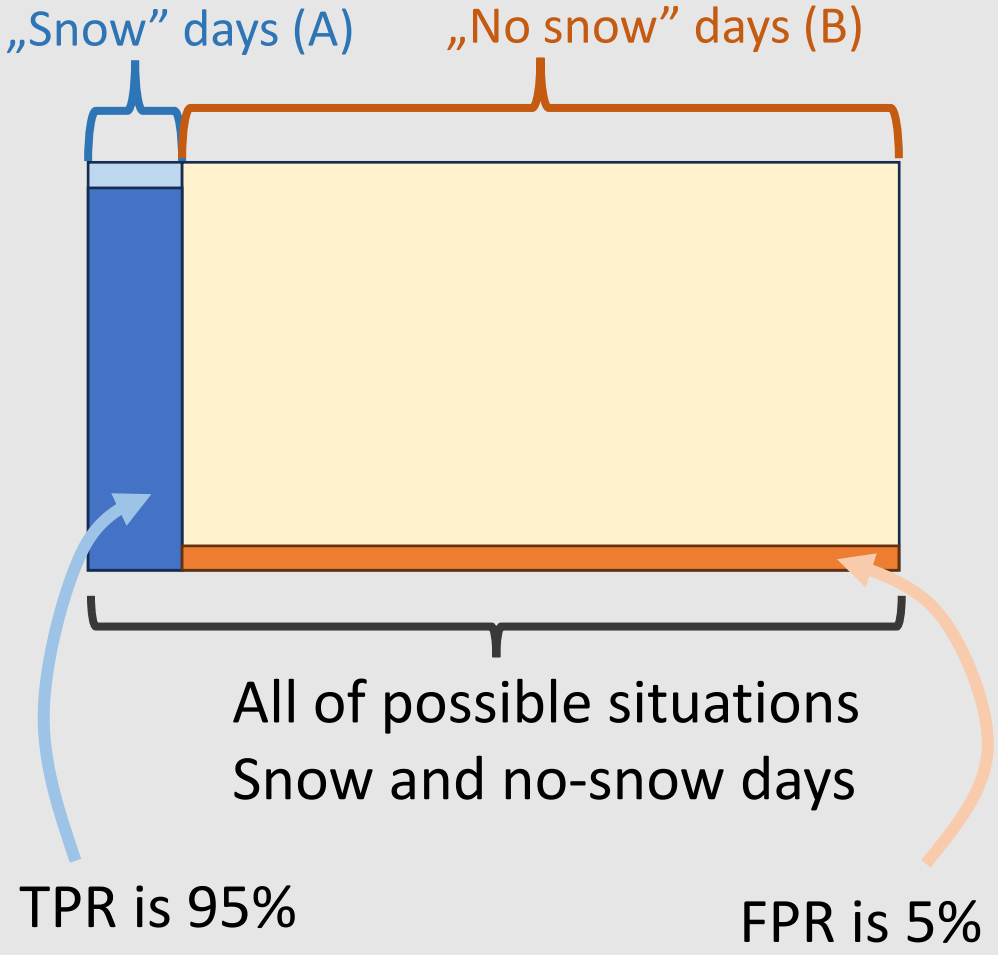
The classifier detects snow the next day. What is the probability of this output being correct?

What is the answer if we know that only 10% of days are snowy?

What if we have this answer in the middle of June?

Gentle introduction to bayes rule

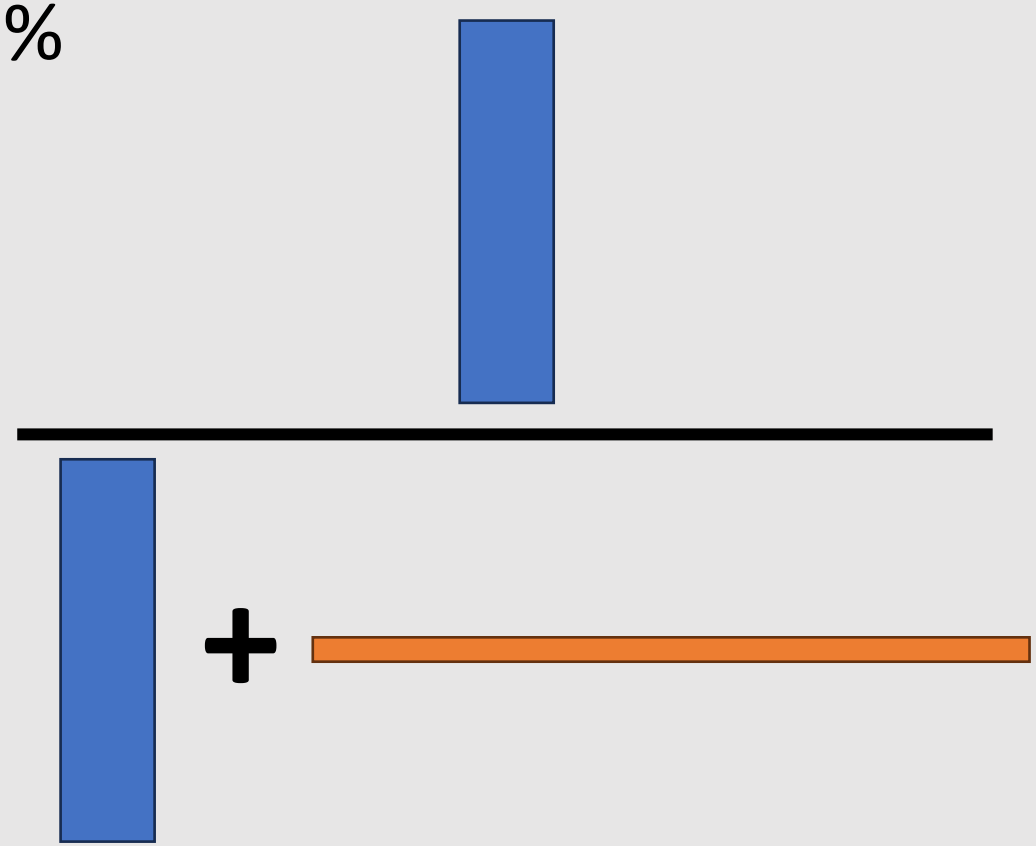
We know that there is some underlying probability of class existence. Let's say that its 1:9:



The classifier detects snow the next day. What is the probability of this output being correct?

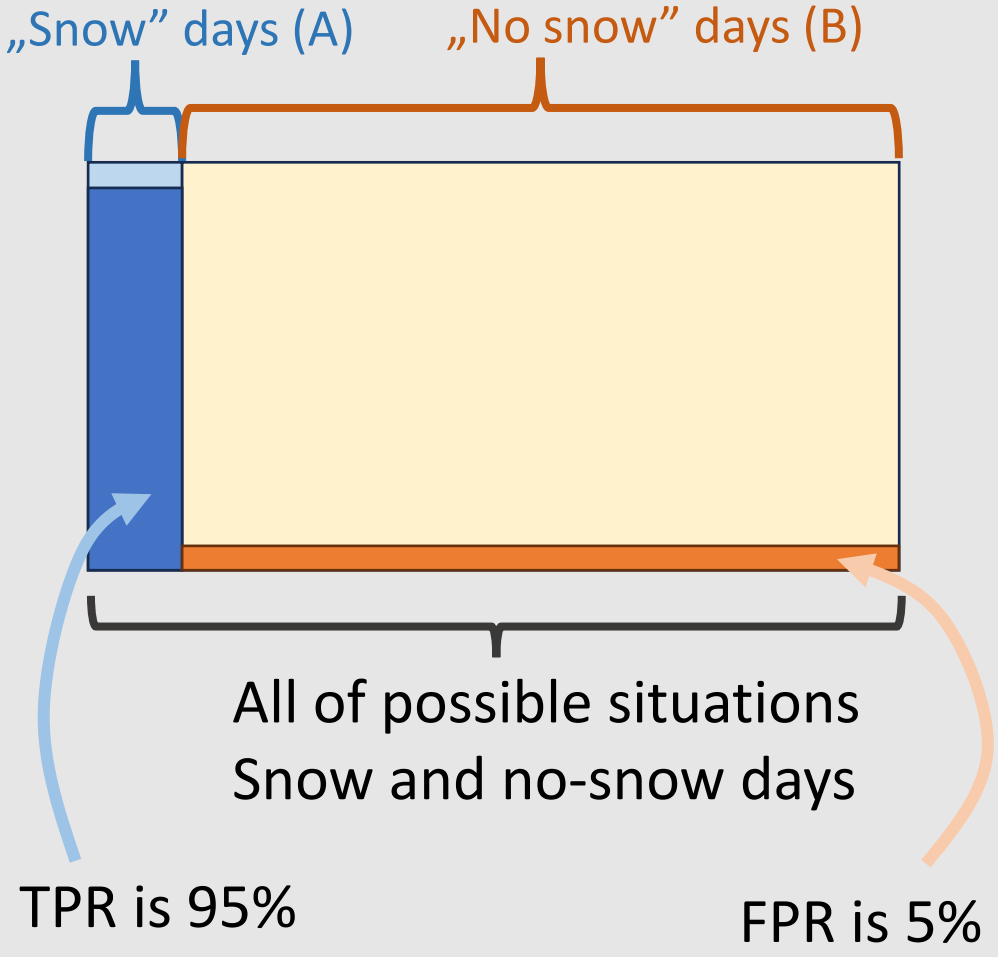
$$P(A|y_A) = \frac{P(y_A|A) \cdot P(A)}{P(y_A|A) \cdot P(A) + P(y_A|B) \cdot P(B)}$$

68%



Gentle introduction to bayes rule

We know that there is some underlying probability of class existence. Let's say that its 1:9:



The classifier detects snow the next day. What is the probability of this output being correct?

$$P(A|y_A) = \frac{P(y_A|A) \cdot P(A)}{P(y_A|A) \cdot P(A) + P(y_A|B) \cdot P(B)}$$

68%

In a generalized form this equation is called Bayes conditional probability (Bayes rule):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

In this equation „A" and „B" refer to general events, not classes in our example! We were denoting B as y_A

A hypothesis interpretation to bayes rule

The classifier detects snow the next day. What is the probability of this output being correct?

1. Consider how likely we are to see snow in the first place (10%)



*This is our **prior** probability (assumption before observation)*

2. Consider that we actually have a classifier that tells us „there will be snow” (95% correctness)



This is new evidence (actual observation). We expect it to UPDATE our prior belief

3. We combine new evidence with prior assumption to update our belief - we are **more likely** than before to see snow because new evidence tells us so (68%)



*This is our **posterior** probability (updated with new evidence)*

A perfect introduction to graphical interpretation of Bayes rule:
https://youtu.be/HZGCoVF3YvM

Things to remember:

1. Define what is a false positive, true positive, false negative and true negative indication, provide a practical example
2. Define classifier metrics: FPR, TPR, FNR, TNR, Precision, Accuracy, Recall, Sensitivity, Specificity
3. Draw examples of ROC and PR diagrams, describe elements and show how threshold setup affects the curves, explain how two different classifiers can be compared on one diagram
4. Explain how ROC and PR diagrams are different from usage perspective and how are they similar (what purpose do they serve)
5. Draw an example of a confusion matrix, explain how can it be used to improve experiment or understand classification outcomes
6. Explain metrics used for regressor evaluation
7. Explain (also with graphical interpretation) what residua analysis and error distribution analysis is
8. Explain what a confidence interval for metric calculations are
9. Draw a graphical interpretation of a Bayes rule, explain probability of correct classification given TPR, FPR and class prevalence