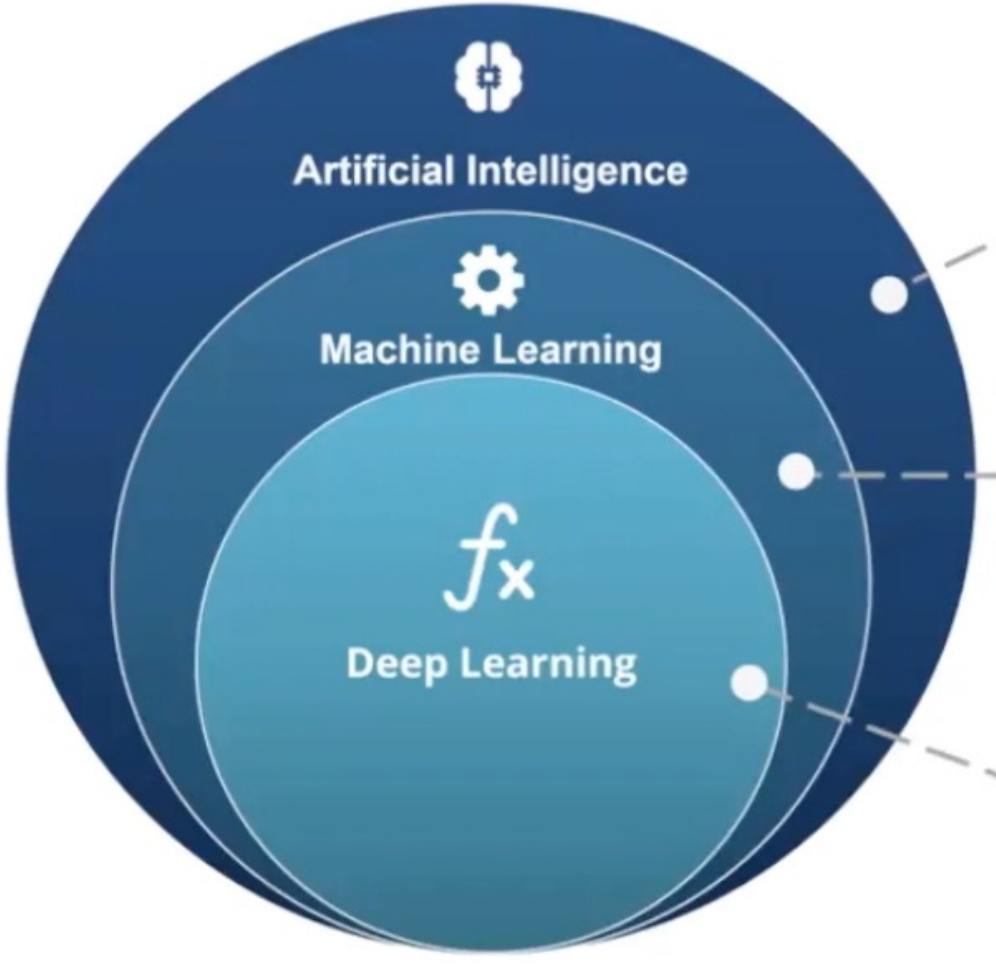
Machine learning –Basics (Lecture I)

Machine learning – Artificial Neural Networks (Lecture 2)

Tiam Heydari

Terminology:



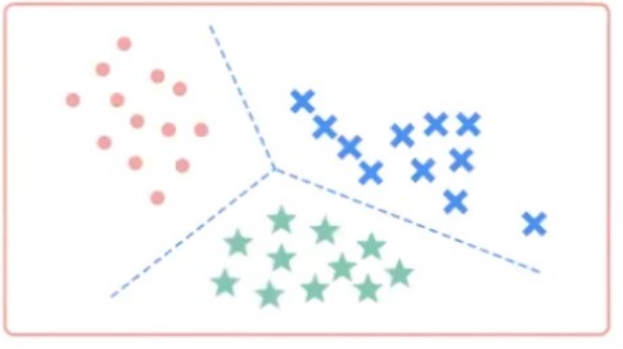
A technique that enables machines to **mimic human behavior**
Ex: Computer plays chess

Statistical methods that allow computer to **optimize models from data**
Ex: Given DNA sequence, predict that chance of type x cancer

Specific class of machine learning models **that uses multiple layers of processing**
Ex: Predicting gene expression from DNA sequence

Two main classes of machine learning:

Supervised ML



Have a labeled target

Aim: to learn outputs depend inputs

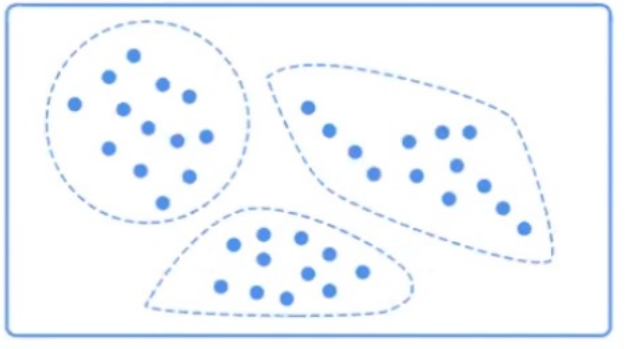
Regression: numeric target

-predict efficiency percentage of a drug

Classification: Categorical target:

-Predict if protein binds to a target or not

Unsupervised ML



No known target variables, unlabeled training data

Aim: to find and learn structure within data

Clustering

PCA

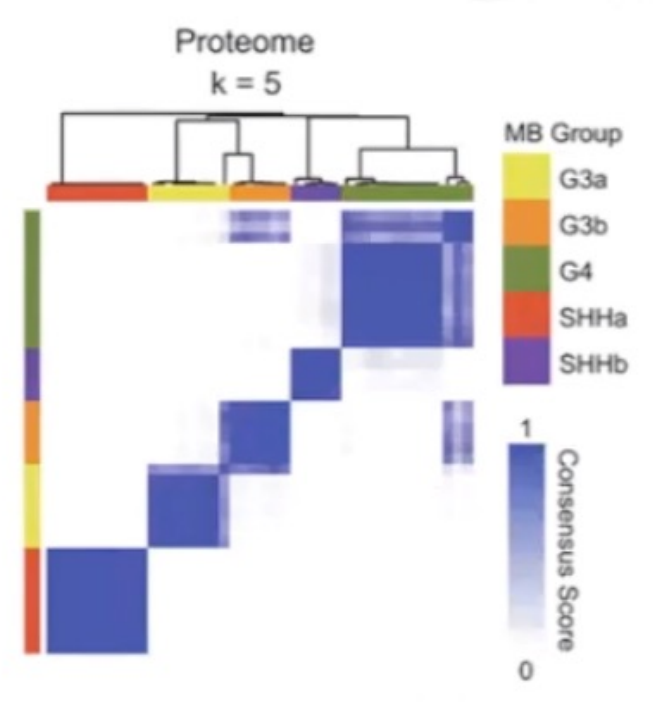
Autoencoders

…

Examples of ML in genomics

- Predicting true variation from read errors in sequencing data
  - DeepVariant
- Predicting chromatin states from ChIP data
  - Segway, chromHMM
- Predicting TF binding/chromatin structure/gene expression from DNA sequence
  - Basenji, DeepSEA, Enformer, BP-NET, DREAM-net (de Boer group)
- Finding hidden structure in single cell genomics data
  - Many
- Predicting cancer prognosis from observed mutations

Check your knowledge



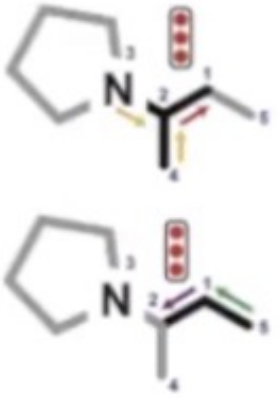
Identifying medulloblastoma subtypes through proteome clustering (Archer 2018)

Task: Given proteomic measurements, identify the group of patients that share similar markers

A) Supervised

B) Unsupervised

Check your knowledge



Selecting antibiotic candidates from large (>107M) chemical libraries (Stokes 2020)

Task: Given a database of antibiotics with associated bacterial kill rate:

For a new antibiotic, predict the associated kill rate

A)Supervised

B)Unsupervised

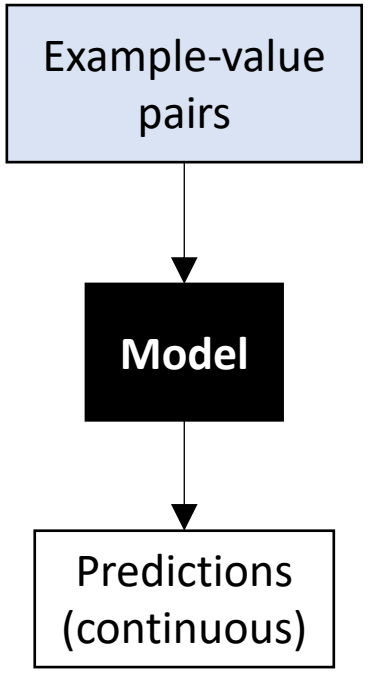
Example supervised models

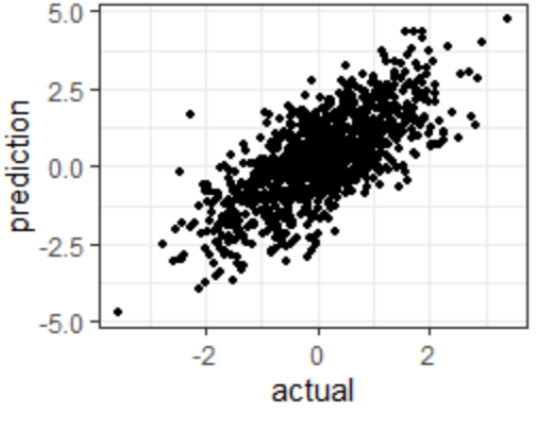
- Least squared regression
- Support vector machines
- Decision trees

- How these work in detail is important, but beyond our scope

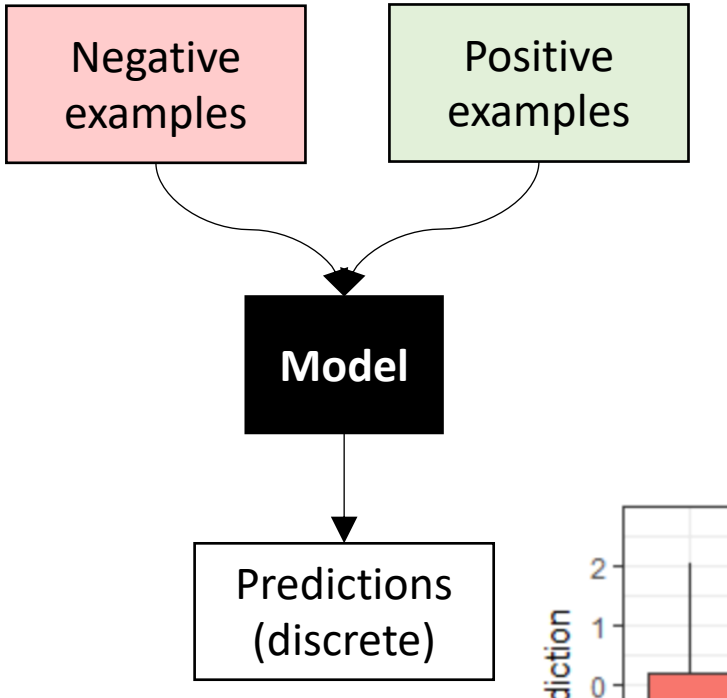
General workflow for training a ML model

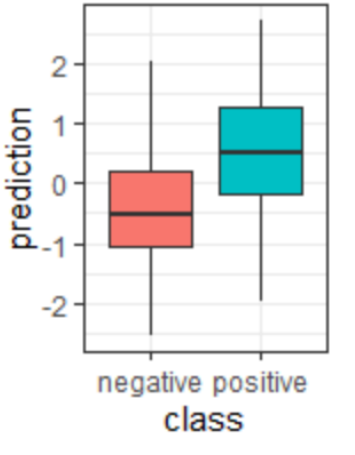
Regression





Classification

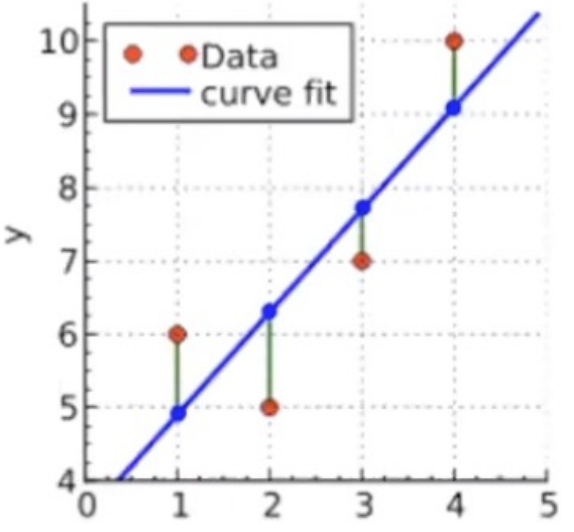




Example: Linear Regression

Y=mx+b

x: edit length	y: editing efficiency (%)
1	6
2	5
3	7
4	10

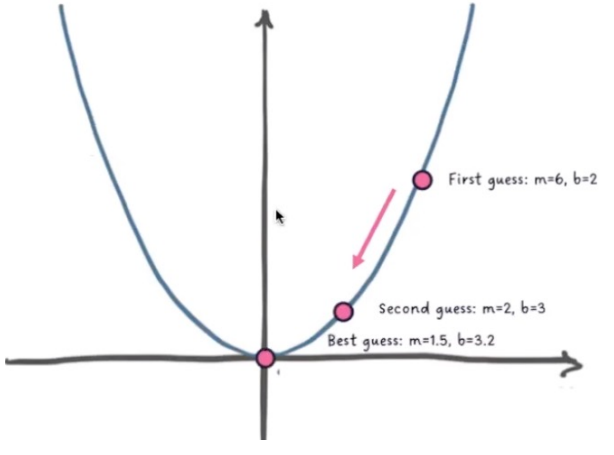


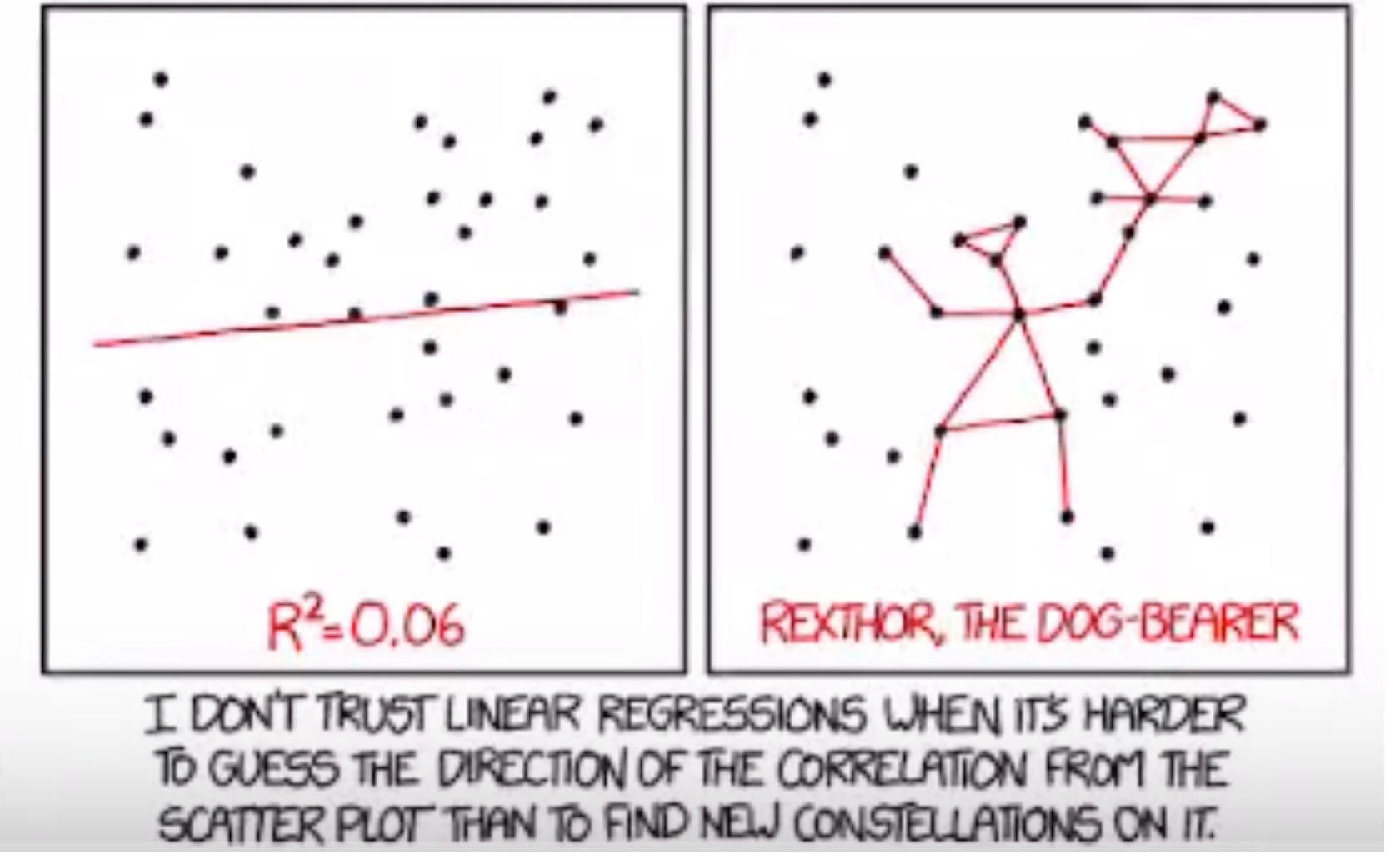
$$MSE = \frac{1}{N}\sum (y_i - \hat{y}_i)^2$$

How could we say if our model is performing well?

true editing efficiency	predicted efficiency by model	squared difference [loss]
6	5	1
5	6.2	1.44
7	7.7	0.49
10	9.2	0.64
Average (MSE)		**0.89**

Squared error:
$(y_{real} - y_{predicted})^2$




$R^2 = 0.06$
REXTHOR, THE DOG-BEARER
I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

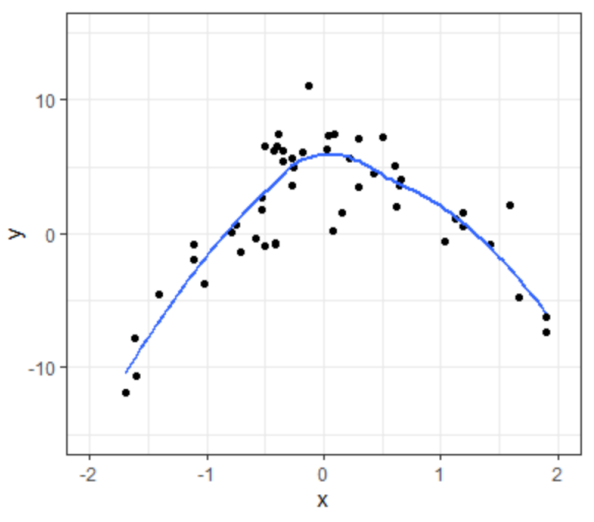
Test your intuition:

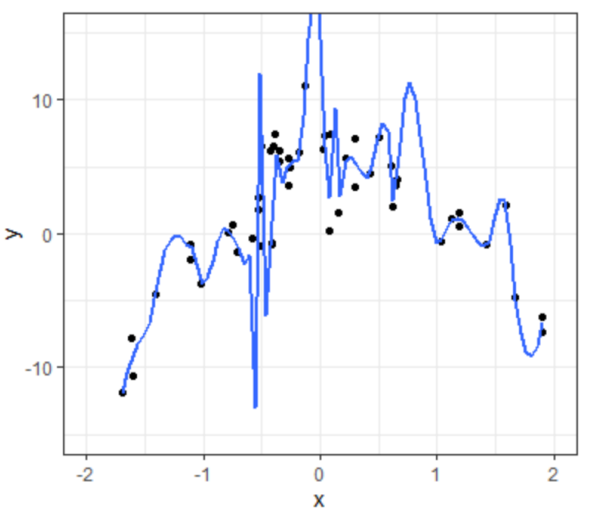
I have trained two different models on my dataset, which is more powerful to predict the correct values in another dataset (test dataset)?

Model A

Model B

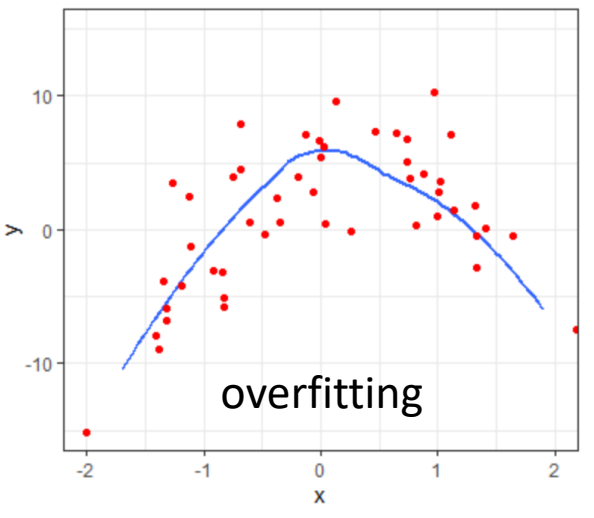
Training data

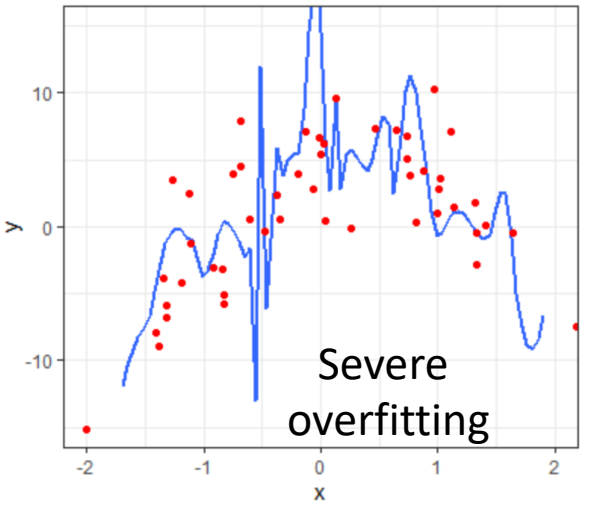




Let's see:

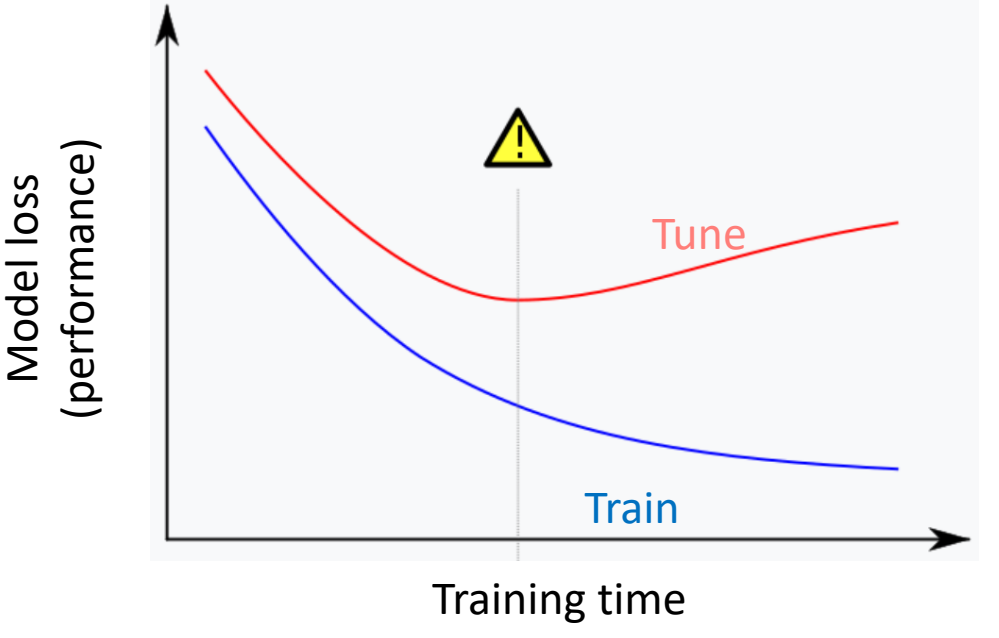
Test data





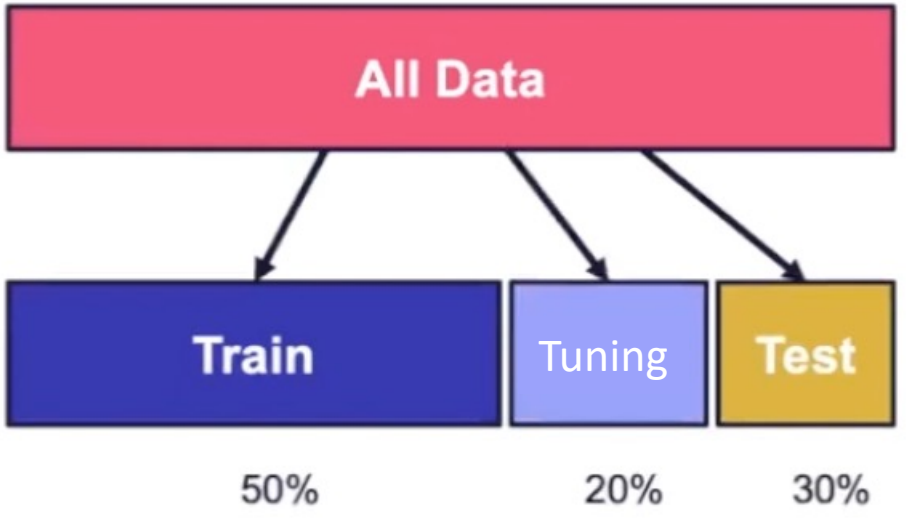
Overfitting

- Model learns features **specific** to the training data
- "**Memorizing**" training examples: Common in models with many parameters
- Manifests as improved performance on training data with no improvement (or even degradation) of performance on tuning/test data
- Essentially **all** models will be overfit at least a little to training data
- Model optimization is a "race" between fitting and overfitting
  - Stop when overfitting>fitting



From:wikipedia

Train/Tune/Test Datasets



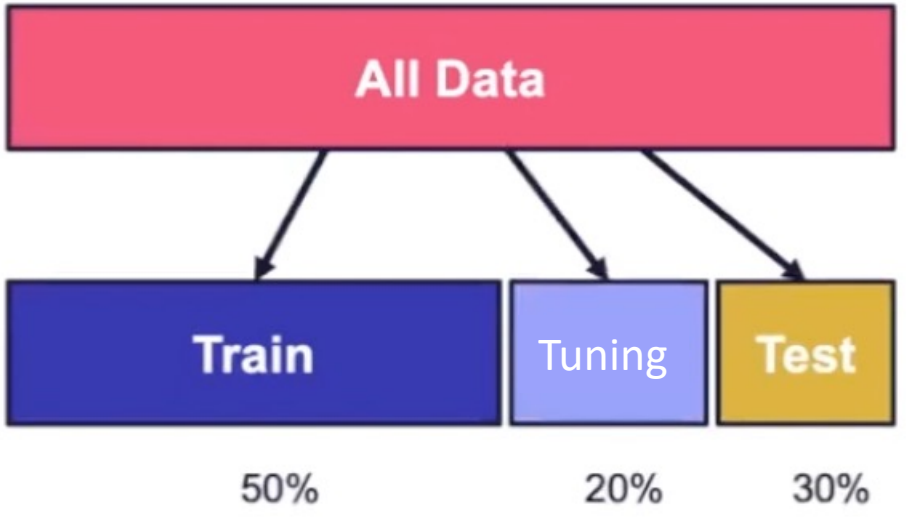
- Training:
  - This is the dataset the model **learns from**.
  - The model adjusts its internal parameters (weights) based on this data.
  - It represents the majority of the data used during training.

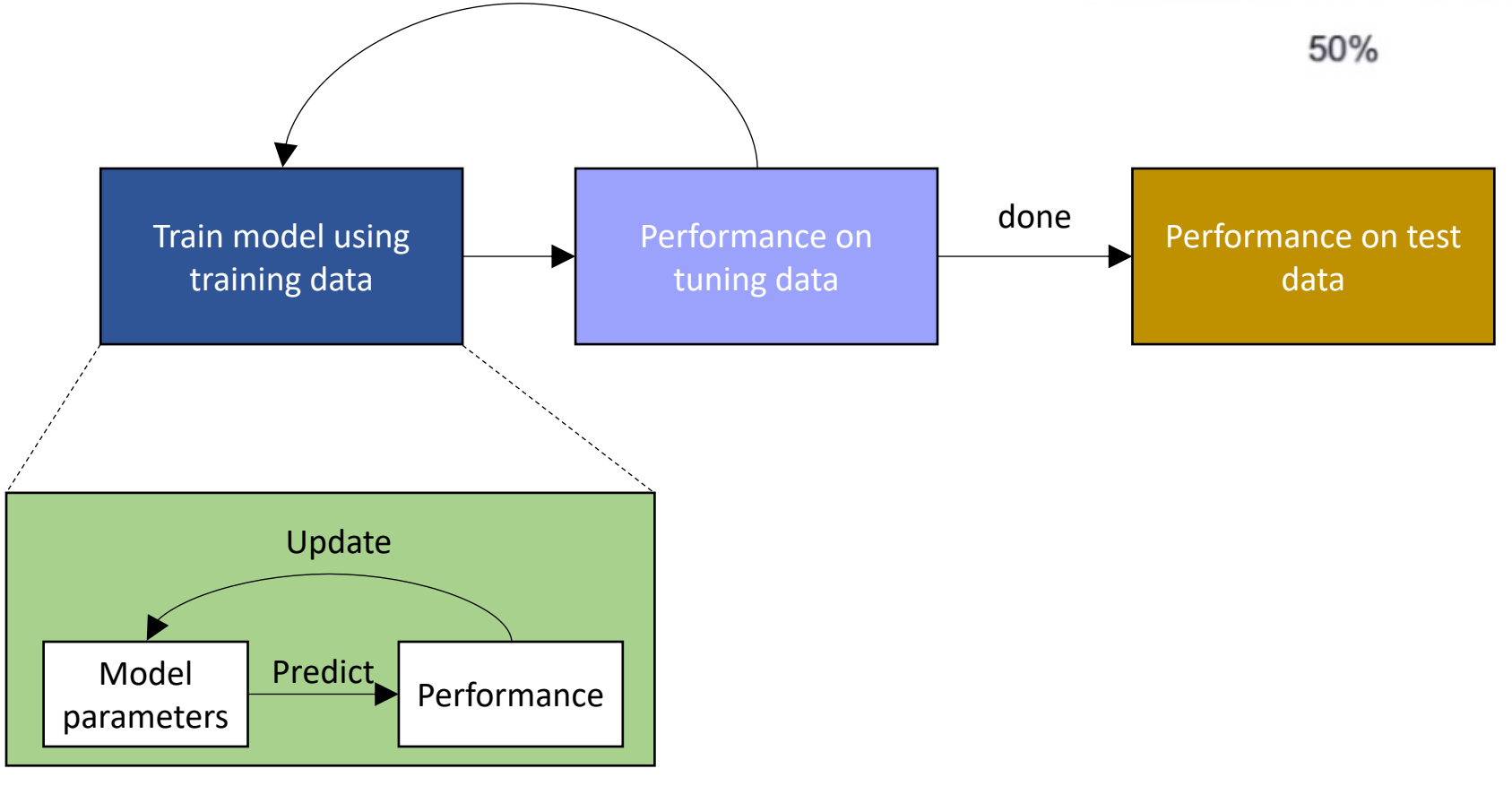
Tuning (Validation Set):

- This dataset is used to **fine-tune** the model, particularly its **hyperparameters** (e.g., learning rate, number of layers, regularization strength).
- Helps to detect overfitting, which occurs when a model performs well on training data but poorly on new data.
- It is NOT used to train the model itself, only to guide improvements.

- Testing:
  - Confusingly, also sometimes known as “validation” data
  - The final dataset used to evaluate model performance
  - Don’t look at this until the end

Training a model





What metric(s) can we use to evaluate performance?

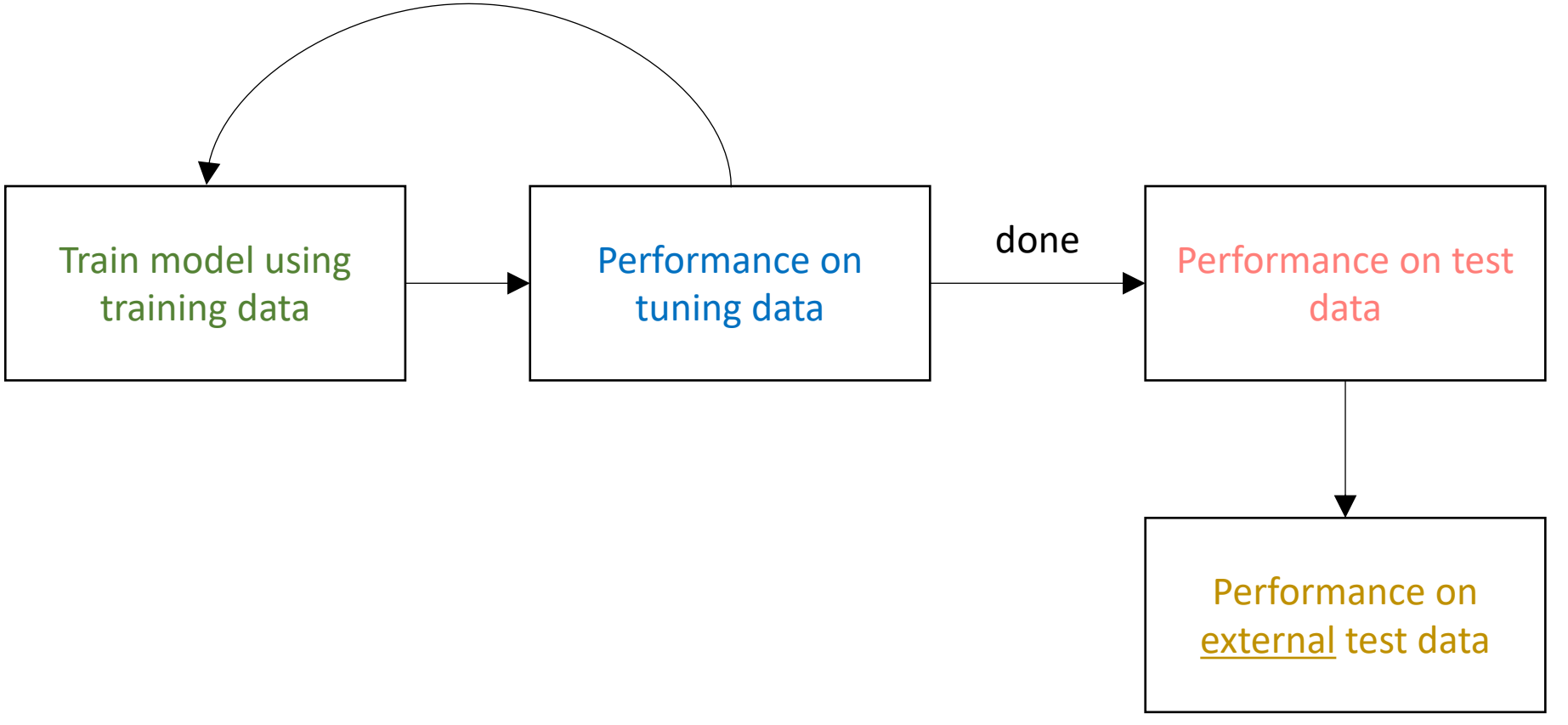
Training a model

The model learns from the training dataset by adjusting its internal parameters to minimize loss.

Once the model is tuned, it is evaluated on a separate **test set** to assess how well it generalizes to unseen data.

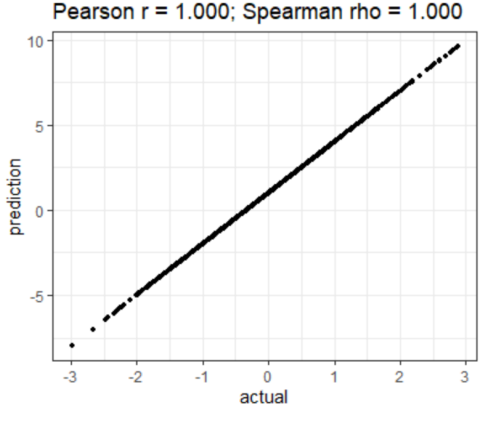
If the model's performance is poor, adjustments are made (shown by the feedback loop arrow back to training).

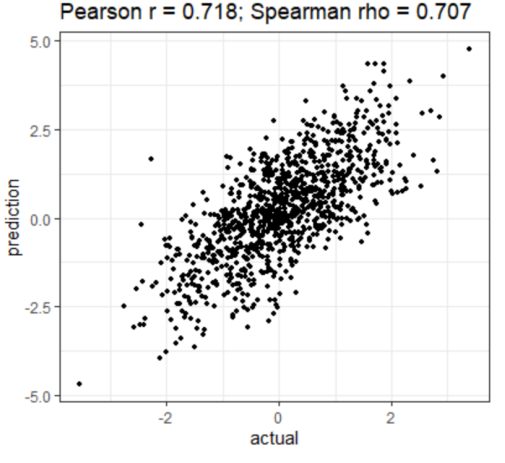
Apply to a brand new dataset: external dataset for real-world performance evaluation and ensure model generalize well

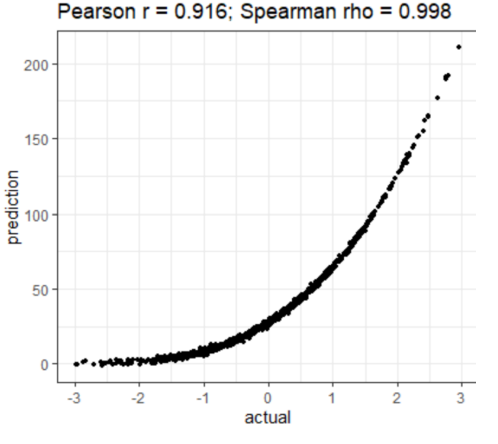


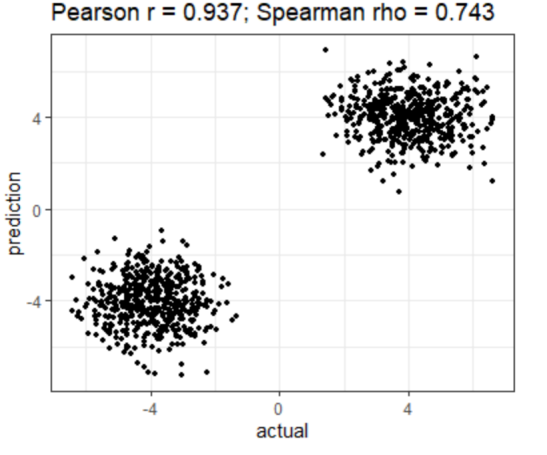
Evaluation metrics: regression

- **Pearson correlation (*r*)**
- **R squared** (This is a metric that indicates how well model predictions approximate the true values, where 1 indicates perfect fit, and 0 would be R squared of a DummyRegressor that always predicts the mean of the data.)
- **MAE (Mean Absolute Error)** More robust to outliers, measures average absolute difference.
- **MSE (Mean Squared Error)** Penalizes large errors more than MAE**.**
- **RMSE (Square root of MSE**) interpretable in original units.









Quiz

I am trying to predict ATAC-seq reads using the DNA sequence. I try a neural network configuration, train it, tune the hyper parameters, and finally run it on the test data. It performed fairly well (r=0.6), but I think it can be improved so I go back and update the neural network. Through 10^6 or so iterations until I get the test-data performance up to r=0.8! What is the true performance of this final model?

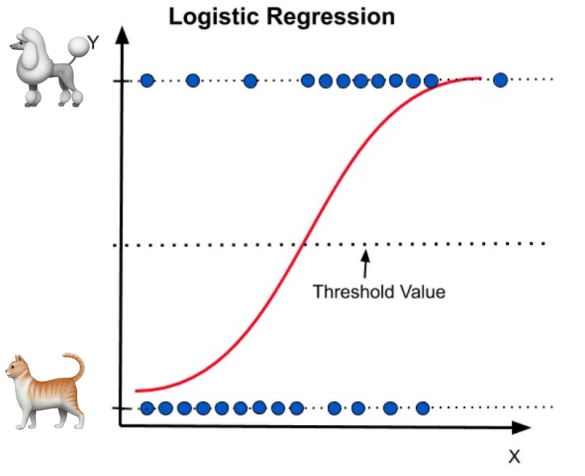
A) 0.6

B) 0.8

C) <0.8

D) >0.8

Some Classification methods



- **Logistic Regression** → Simple, interpretable, works well for linear separability.
- **Decision Trees** → Easy to understand, prone to overfitting without pruning.
- **Random Forest** → Ensemble of decision trees, reduces overfitting, robust.
- **Support Vector Machine (SVM)** → Effective for high-dimensional data, uses the kernel trick.
- **Neural Networks (DNNs, CNNs)** → Handles complex patterns, requires larger data.

Example: Classification

We have a group consist of two types

We trained a sophisticated classification model, and looks like it performed well based on accuracy:

Accuracy= .0964

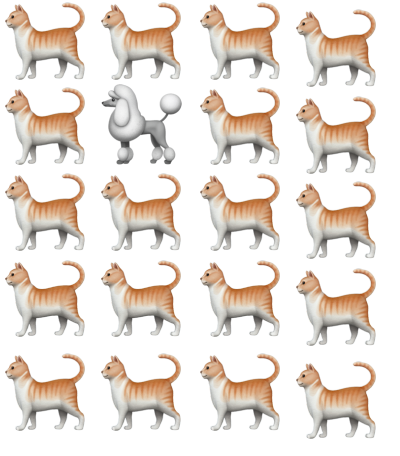
$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

But we trained a **dummy classifier** (follows a given strategy) and looks like it performed well too:

Accuracy= .095

What is happening?

My data:
10000 samples,
95% type1, %5 type two



Class imbalance!

accuracy alone doesn't give enough information!

Classification: Selecting a negative set

- Shapes **what your model will learn**
- Not always obvious
- May have its own biases
- E.g. Negative set for predicting enhancers
  - Nearby sequences?
  - Sequences not known to be enhancers?
  - Sequences that are enhancers in other cell types?
  - Random sequences?
    - GC-matched?
    - Dinucleotide matched?
    - Trinucleotide matched?

Classification: Selecting a negative set

- Shapes **what your model will learn**
- Not always obvious
- May have its own biases
- E.g. Negative set for predicting enhancers
  - Nearby sequences? These may share features with enhancers and could be misleading.
  - Sequences not known to be enhancers? Might include unannotated enhancers.
  - Sequences that are enhancers in other cell types? if cell-type specificity is a factor.
  - Random sequences?
    - GC-matched? Controls for GC content bias.
    - Dinucleotide matched? Preserves local sequence dependencies.
    - Trinucleotide matched? Further refines sequence similarity control.

If the negative set is not well chosen, the model might not actually learn what makes an enhancer unique—it might just learn dataset-specific biases!

classification : Evaluation metrics

Some definitions:

True Positives (TP): a sample was positive, and the model predicted a positive class for it.
True Negatives (TN): a sample was negative, and the model predicted a negative class for it.
False Positives (FP): a sample was negative, but the model predicted positive.
False Negatives (FN): a sample was positive, but the model predicted negative.

Confusion matrix
-is not a metric
-diagnostic tool

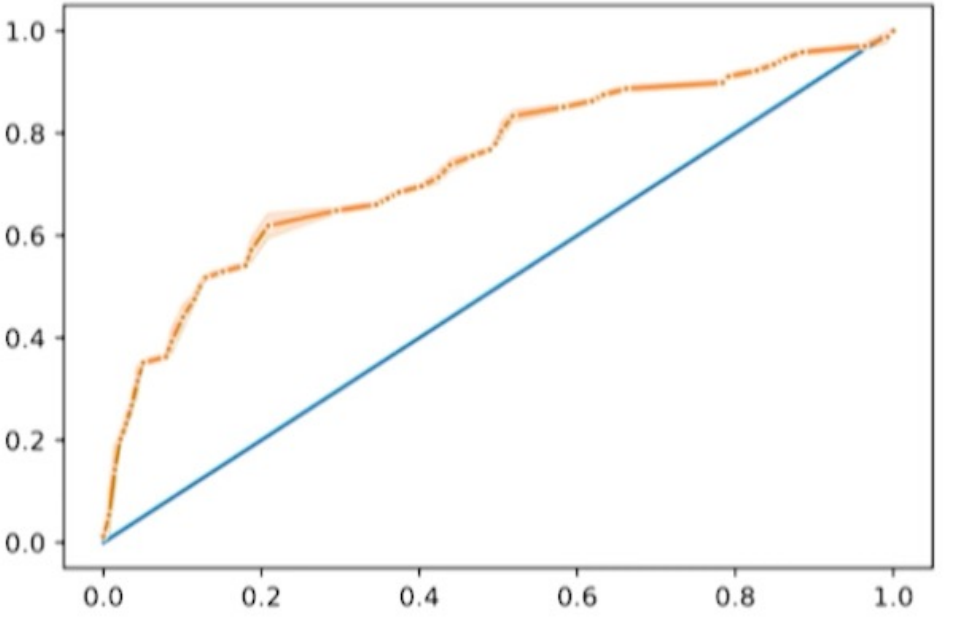
	Predicted: 0	Predicted: 1
True: 0	TN = 126	FP = 13
True: 1	FN = 24	TP = 60

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

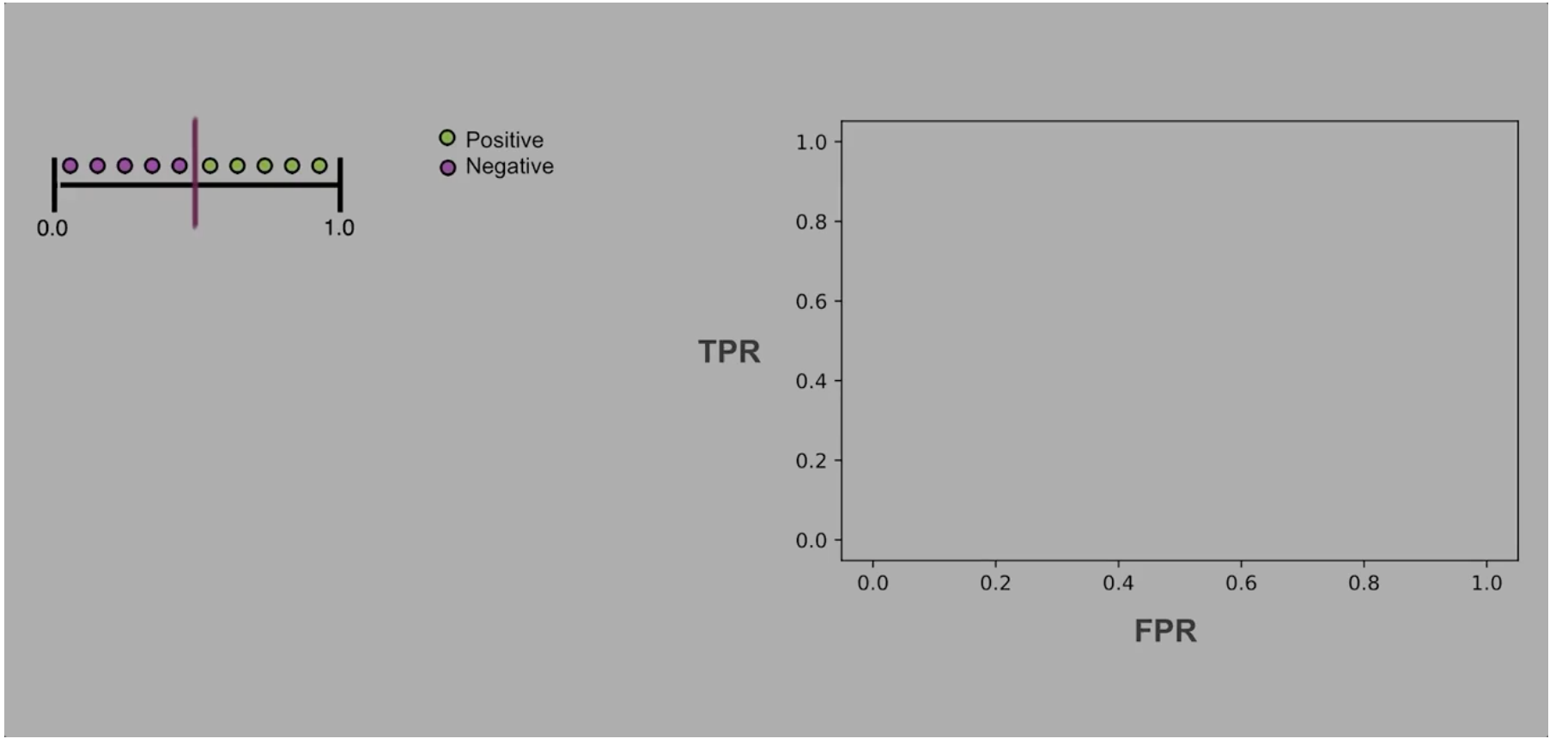
classification : ROC Curve

$$\text{True Positive Rate} = \frac{TP}{TP + FN}$$

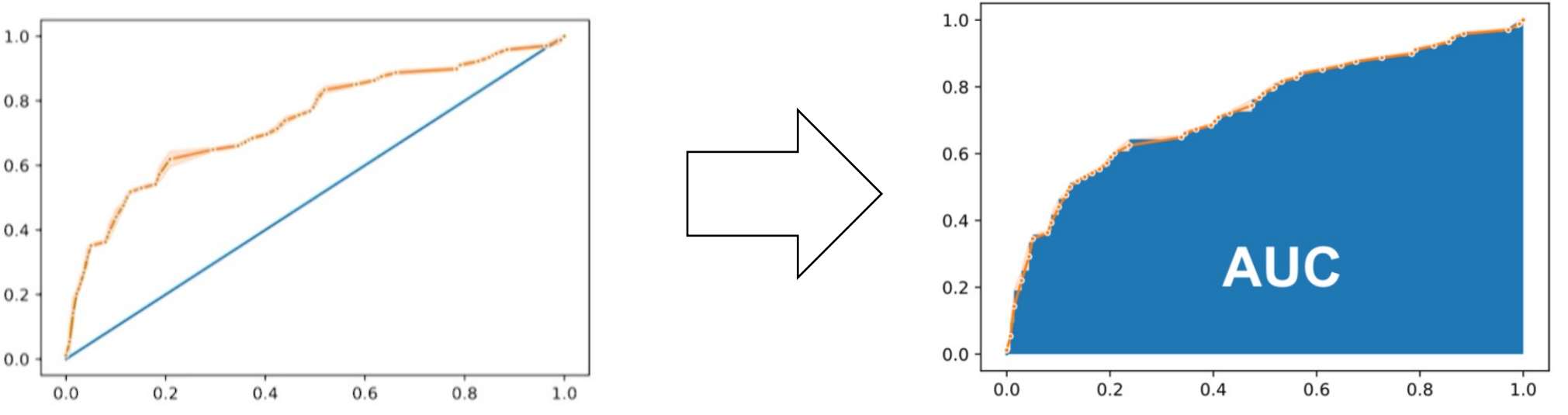


$$\text{False Positive Rate} = \frac{FP}{FP + TN}$$

classification : ROC Curve meaning



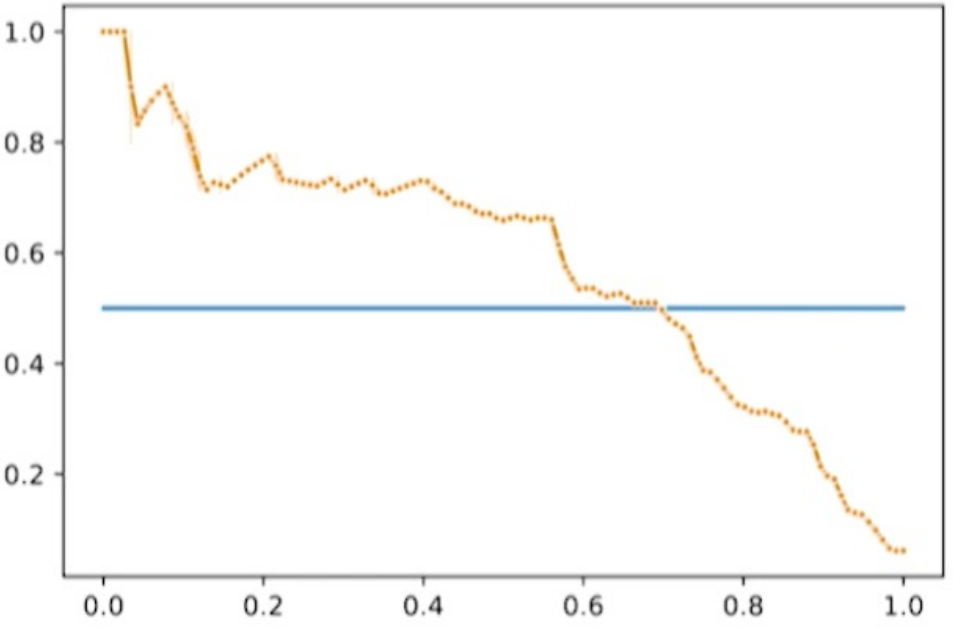
classification : AUC score



The closer it is to 1 (100%), the better

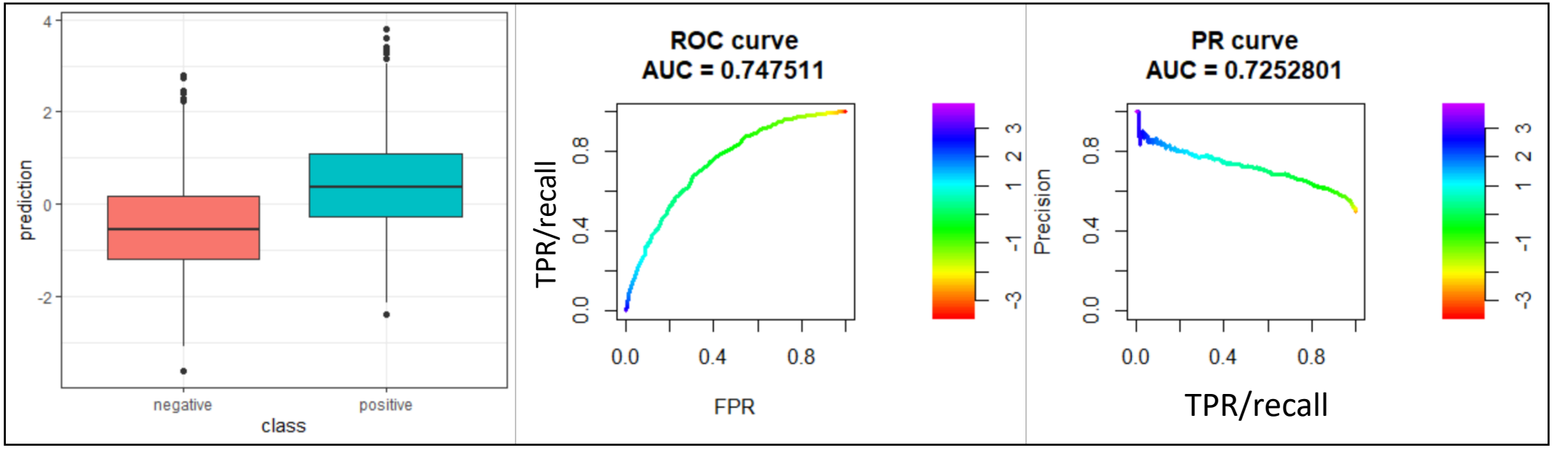
classification : Precision/Recall Curve

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

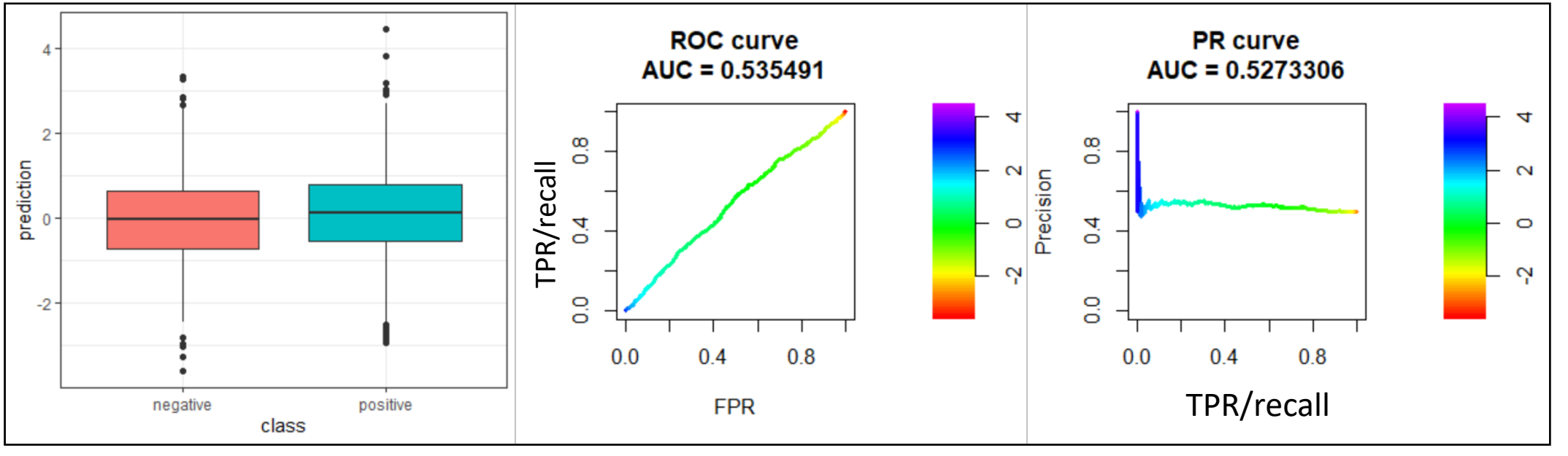


$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

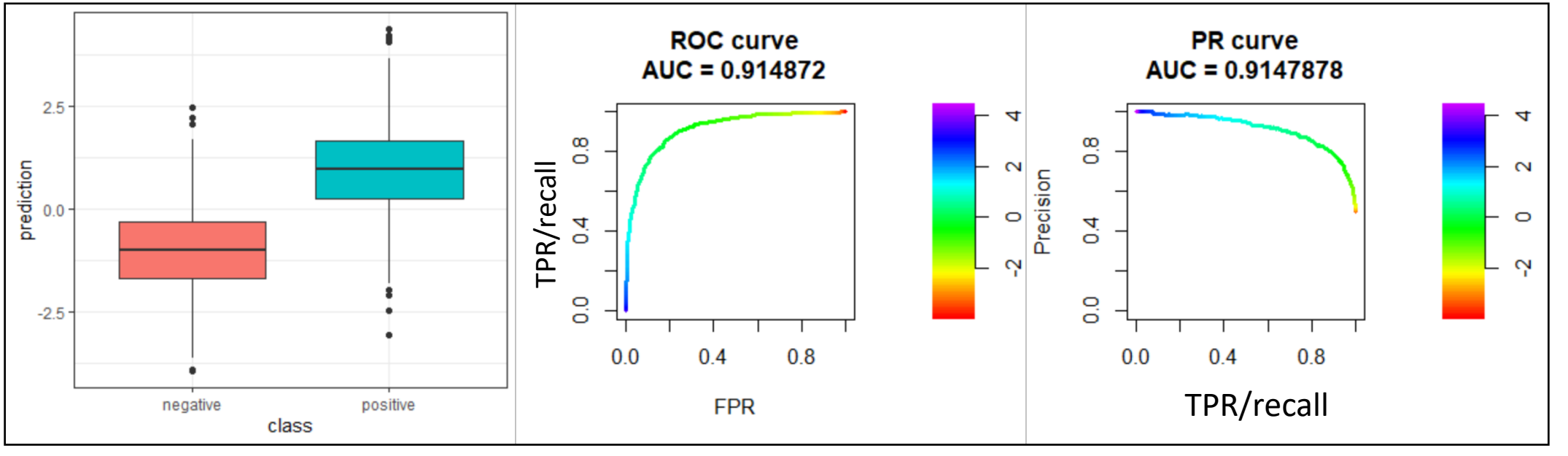
1 SD difference, 1:1 negative:positive



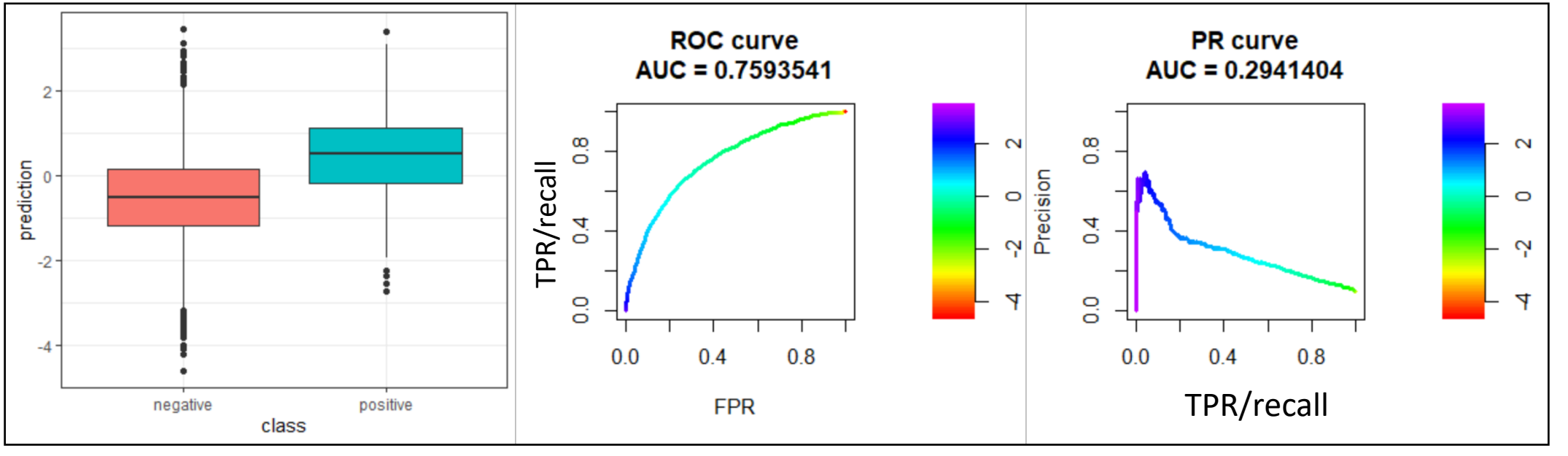
0 SD difference, 1:1 negative:positive



2 SD difference, 1:1 negative:positive



1 SD difference, 9:1 negative:positive



Quiz

I am looking at two papers, both of which claim to build models that identify enhancer regions from sequence. Both quantified performance by AUPRC. Paper 1's model had an AUPRC of 0.6 and Paper 2's model 0.82. Which model is better?

A) Paper 1

B) Paper 2

C) Can't say

Quiz

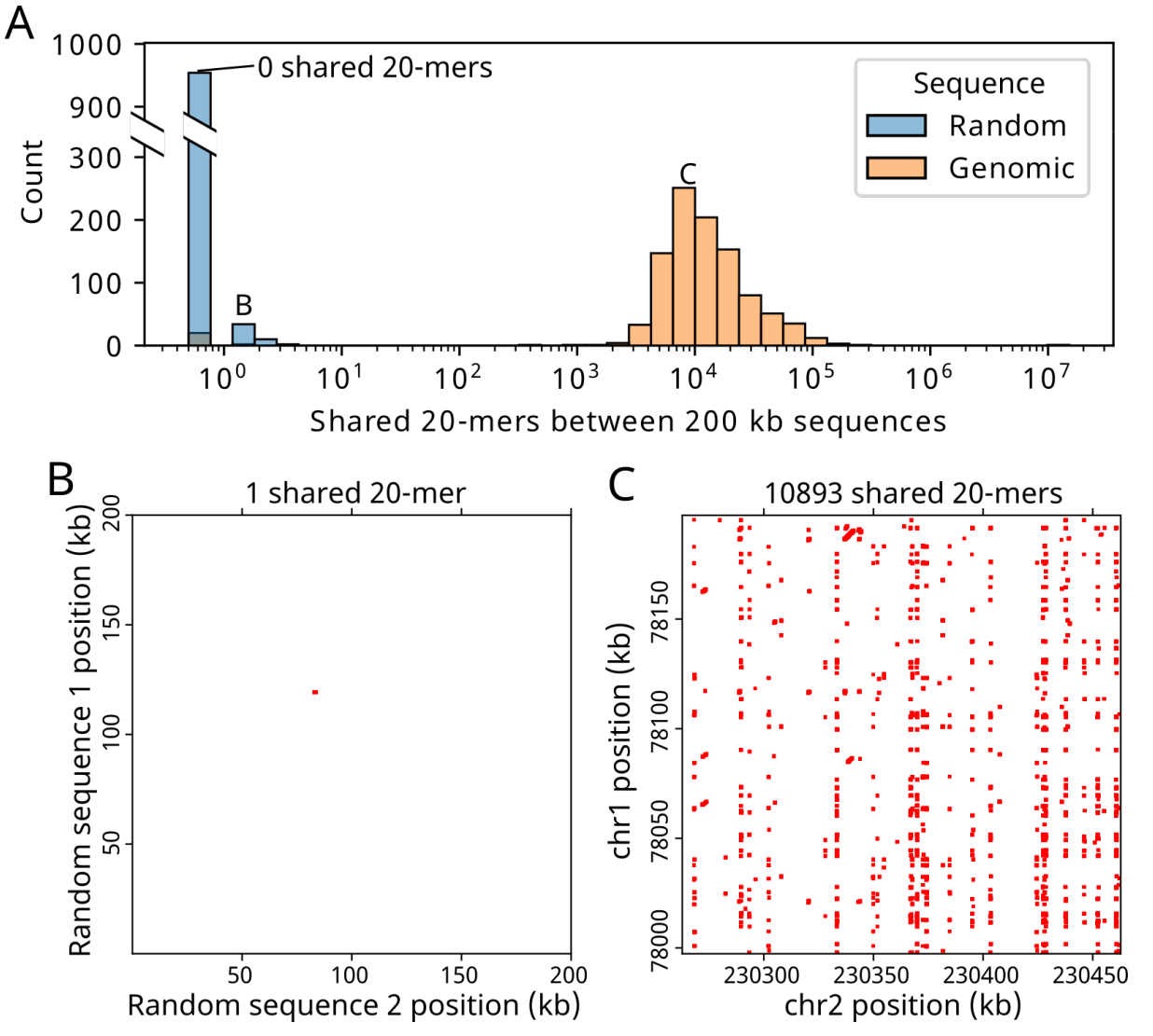
I am looking at two papers, both of which claim to build models that identify enhancer regions from sequence. Both quantified performance by AUPRC. Paper 1's model had an AUPRC of 0.6 **with a 50:50 positive:negative split**, and Paper 2's model had an AUPRC of 0.82 **with a 10:90 positive:negative split**. Which model is better?

A) Paper 1

B) Paper 2

C) Can't say

Cryptic homology across the genome



GIGO Principle

- Garbage in: Garbage out
- You can’t make a good model out of poor quality data
- You can’t learn a relationship that doesn’t exist in data
- Example:
  - Training a model on data from a failed experiment
  - Training a model to predict cancer prognosis from lines on the palm of your hand

Further reading

- ROC curves, PR curves, and terminology
  - https://en.wikipedia.org/wiki/Receiver_operating_characteristic
  - https://en.wikipedia.org/wiki/Precision_and_recall
- Predicting cancer outcome (an example):
  - https://www.nature.com/articles/s41598-020-76025-1
- Overfitting:
  - https://en.wikipedia.org/wiki/Overfitting

End

Model interpretation

- Approach depends on the model
  - Linear model: magnitude and sign of weights
- What to watch out for:
  - Dilution of influence by multiple redundant factors
  - Evidence of confounders (e.g. G/C-rich motifs)
  - Correlated but not causal signals
  - Long motifs with high information content (a sign of overfitting)
  - Many weak signals (a sign of normalization or batch issues)