AI/ML for prediction of biological properties of molecules

Module 3. Training an AI model for bioactivity prediction

Gemma Turon & Miquel Duran-Frigola
Ersilia Open Source Initiative (www.ersilia.io)
18th - 27th of September, 2023

Ersilia

Course overview

Day 1: M0 - Introduction to AI for DD
Day 2 - 3: M1 - Using AI models for DD
Day 4: M2 - Setting up your computational environment
Day 5: M3 - Building an AI model
— Steps to build an AI classifier (morning)
— Training your own AI model (afternoon)
Day 6: M4 - The Ersilia Model Hub
— Joint presentation with BTT students (morning)
— Model deployment & wrap up (afternoon)

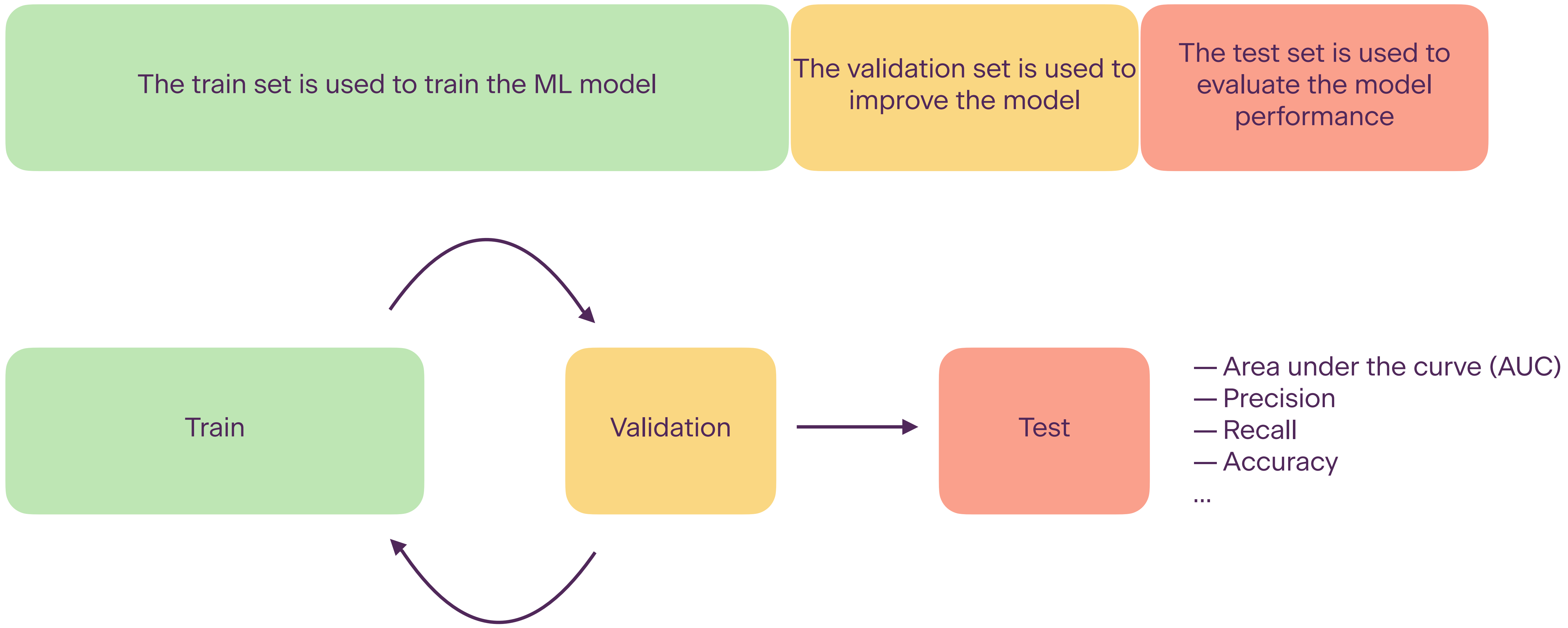
Classification tasks

Classification tasks

A classifier identifies the category of a new data point. In the case of bioactivity data, we usually find two categories (hence, a binary classifier):
— Active: 1
— Inactive: 0

To learn more about classifications, we will use a hands-on example with the Open Source Malaria data to create an AI model that predicts whether a drug will be active or not against the malaria parasite in vitro:
— Experiment: IC50
— Values: microMolar

Train, test and validation sets


The train set is used to train the ML model
The validation set is used to improve the model
The test set is used to evaluate the model performance
Train
Validation
Test
— Area under the curve (AUC)
— Precision
— Recall
— Accuracy
...

In our case-study, we will use an AutoML tool that internally performs the Train and Validation split, so we will only do a Train-Test split

Main steps to train a classifier

1. Observe our data. If the outcome is continuous, we will need to define a cut-off to binarize it
2. Divide the data into train and test sets, ensuring we keep balance between classes (active, inactive)
3. Featurize the molecules (convert the SMILES to vectors or embeddings)
4. Train the ML model: fit
5. Predict the results for the test set and evaluate the performance of the model

Main steps to train a classifier

1. Observe our data. If the outcome is continuous, we will need to define a cut-off to binarize it
2. Divide the data into train and test sets, ensuring we keep balance between classes (active, inactive)

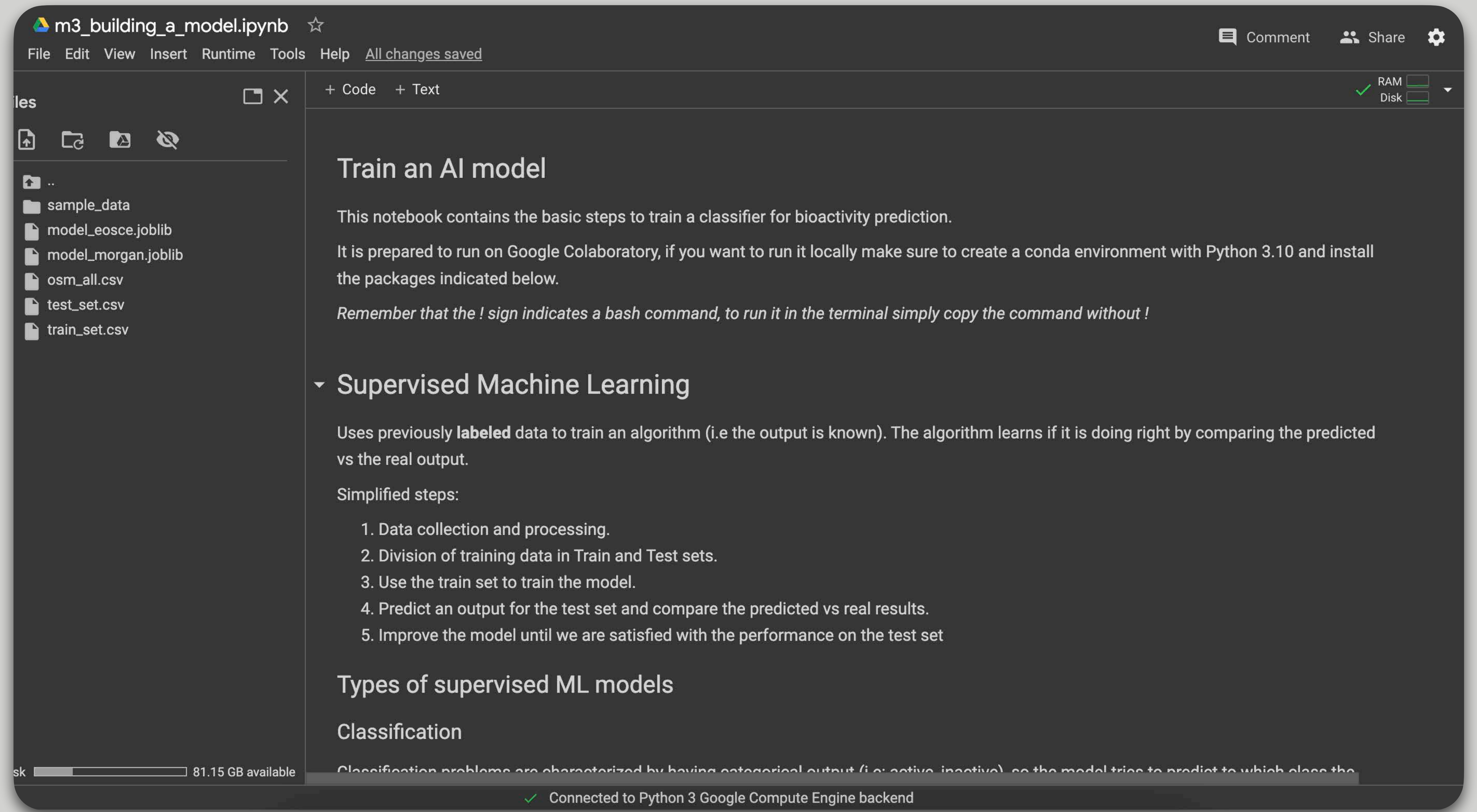
👉 Go to the course repository https://github.com/ersilia-os/ersilia-intro-workshop

👉 Go to notebooks

👉 Open the m3-building-a-model.ipynb

👉 Click on “Open with Colab”

Our Colab Notebook



Train an AI model

This notebook contains the basic steps to train a classifier for bioactivity prediction.

It is prepared to run on Google Colaboratory, if you want to run it locally make sure to create a conda environment with Python 3.10 and install the packages indicated below.

Remember that the ! sign indicates a bash command, to run it in the terminal simply copy the command without !

Supervised Machine Learning

Uses previously **labeled** data to train an algorithm (i.e the output is known). The algorithm learns if it is doing right by comparing the predicted vs the real output.

Simplified steps:

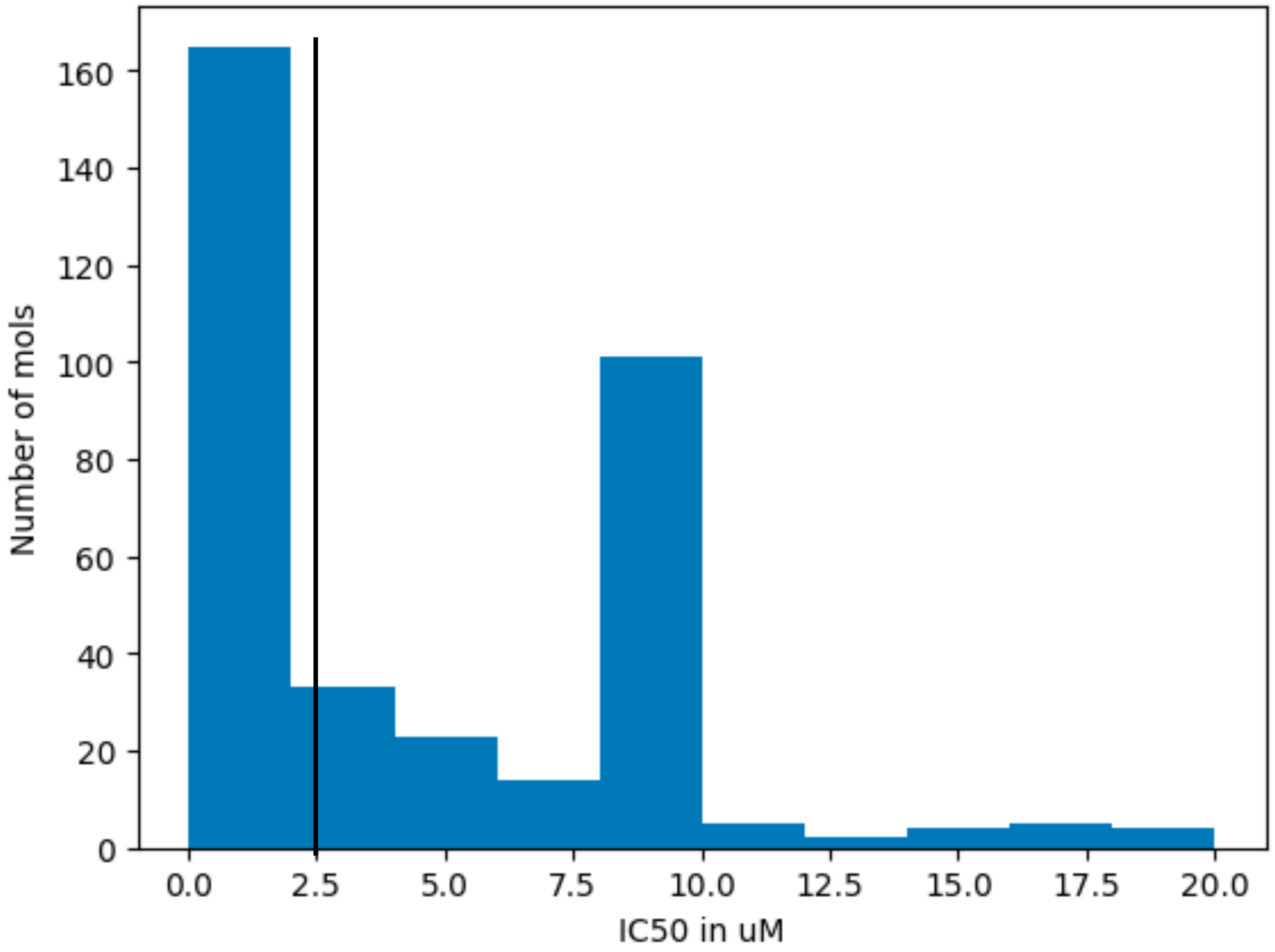
1. Data collection and processing.
2. Division of training data in Train and Test sets.
3. Use the train set to train the model.
4. Predict an output for the test set and compare the predicted vs real results.
5. Improve the model until we are satisfied with the performance on the test set

Types of supervised ML models

Classification

Classification problems are characterized by having categorical output (i.e: active, inactive), so the model tries to predict to which class the

Data processing



Total molecules: 415
Active molecules: 177
Inactive molecules: 238
Frequency of Actives (%): 42.65

The direction of the assay is important.
👉 In which case we might want to select high values as active?

Train set: 332 molecules

Test set: 83 molecules

Random
Keeping the class balance

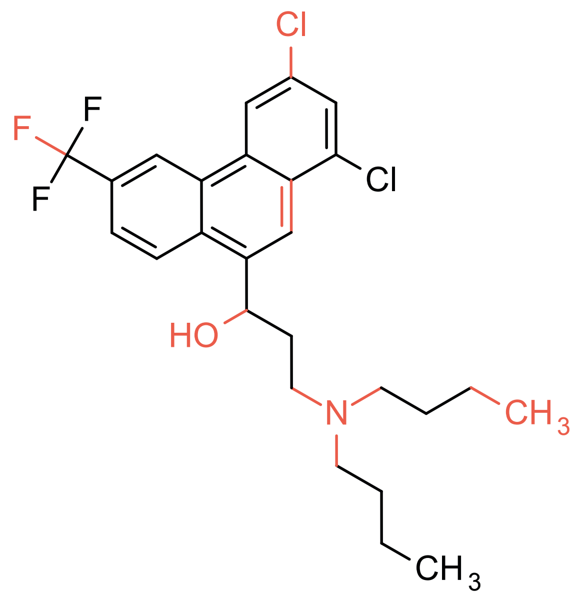
Main steps to train a classifier

3. Featurize the molecules (convert the SMILES to vectors or embeddings
4. Train the ML model: fit

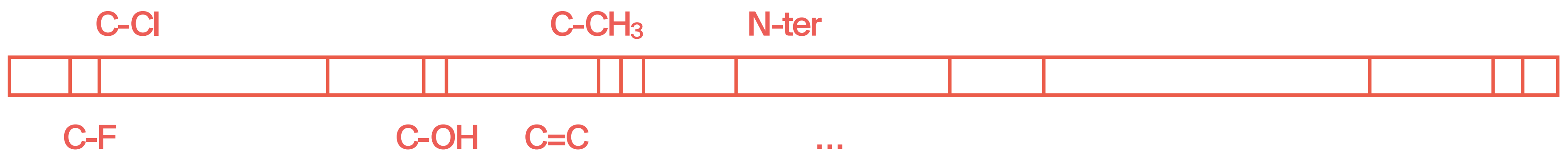
👉 Which are some featurizers we might use?

👉 What are good packages to train ML models?

Molecular featurization



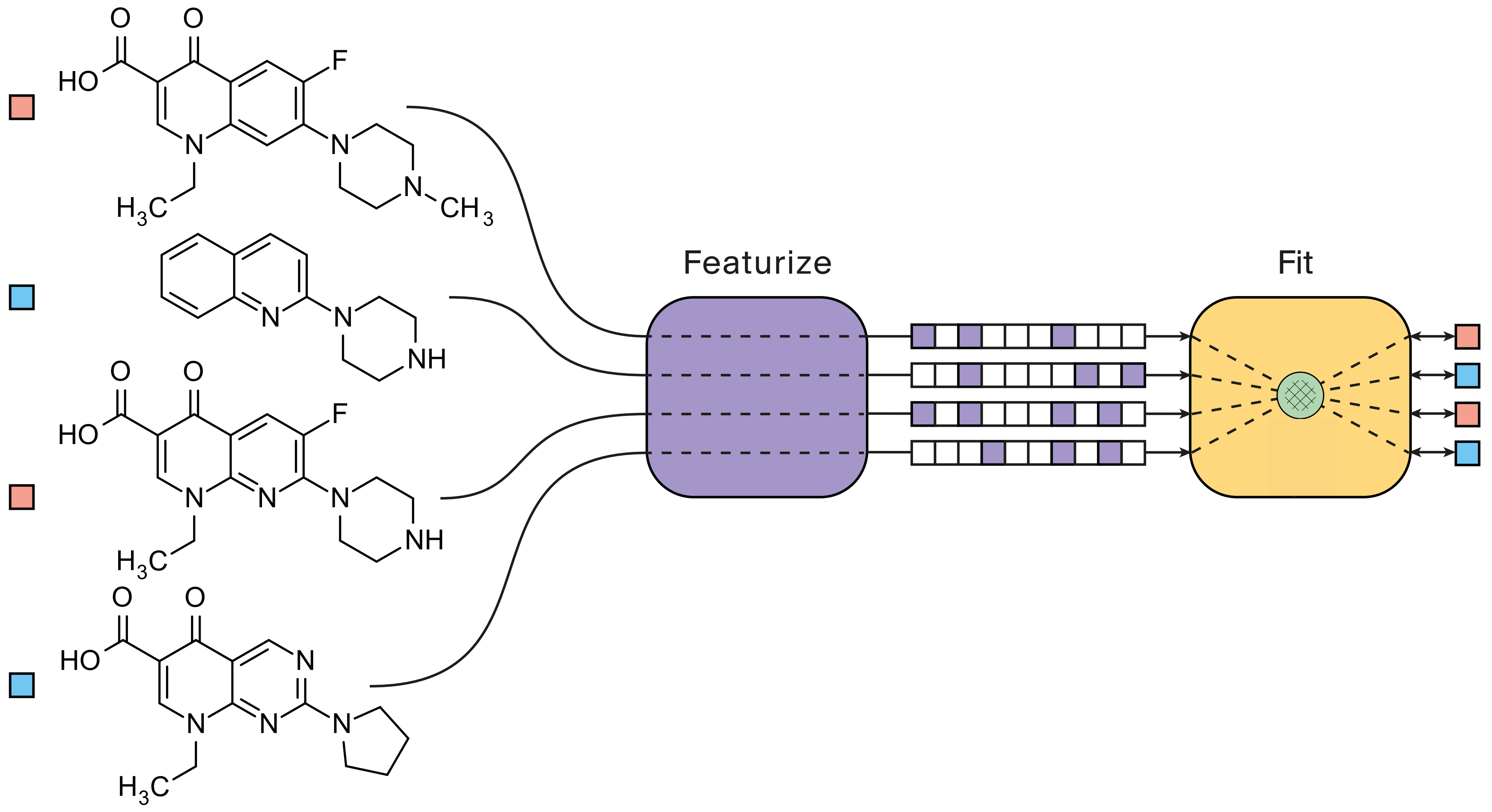
Calculated properties
— Molecular weight
— LogP or LogD
— Hydrogen bonds
— pKa
— Topological surface area (TPSA)
— ...



Halofantrine belongs to the class of organic compounds known as phenanthrenes and derivatives. These are polycyclic compounds containing a phenanthrene moiety, which is a tricyclic aromatic compound with three non-linearly fused benzene. Halofantrine is a synthetic antimalarial which acts as a blood schizonticide. It is effective against multi drug resistant (including mefloquine resistant) P. falciparum malaria. The mechanism of action of Halofantrine may be similar to that of chloroquine, quinine, and mefloquine; by forming toxic complexes with ferritoporphyrin IX that damage the membrane of the parasite. It appears to inhibit polymerisation of heme molecules (by the parasite enzyme 'heme polymerase'), resulting in the parasite being poisoned by its own waste. Halofantrine has been shown to preferentially block open and inactivated HERG channels leading to some degree of cardiotoxicity. Side effects include coughing noisy, rattling, troubled breathing, loss of appetite, aches and pain in joints, indigestion, and skin itching or rash, et cetera, et cetera.

Duran-Frigola et al. Nat Commun, 2014
Duran-Frigola et al. Nat Biotech, 2020

Building a AI model



SMILES

Morgan fingerprint

Binary activity

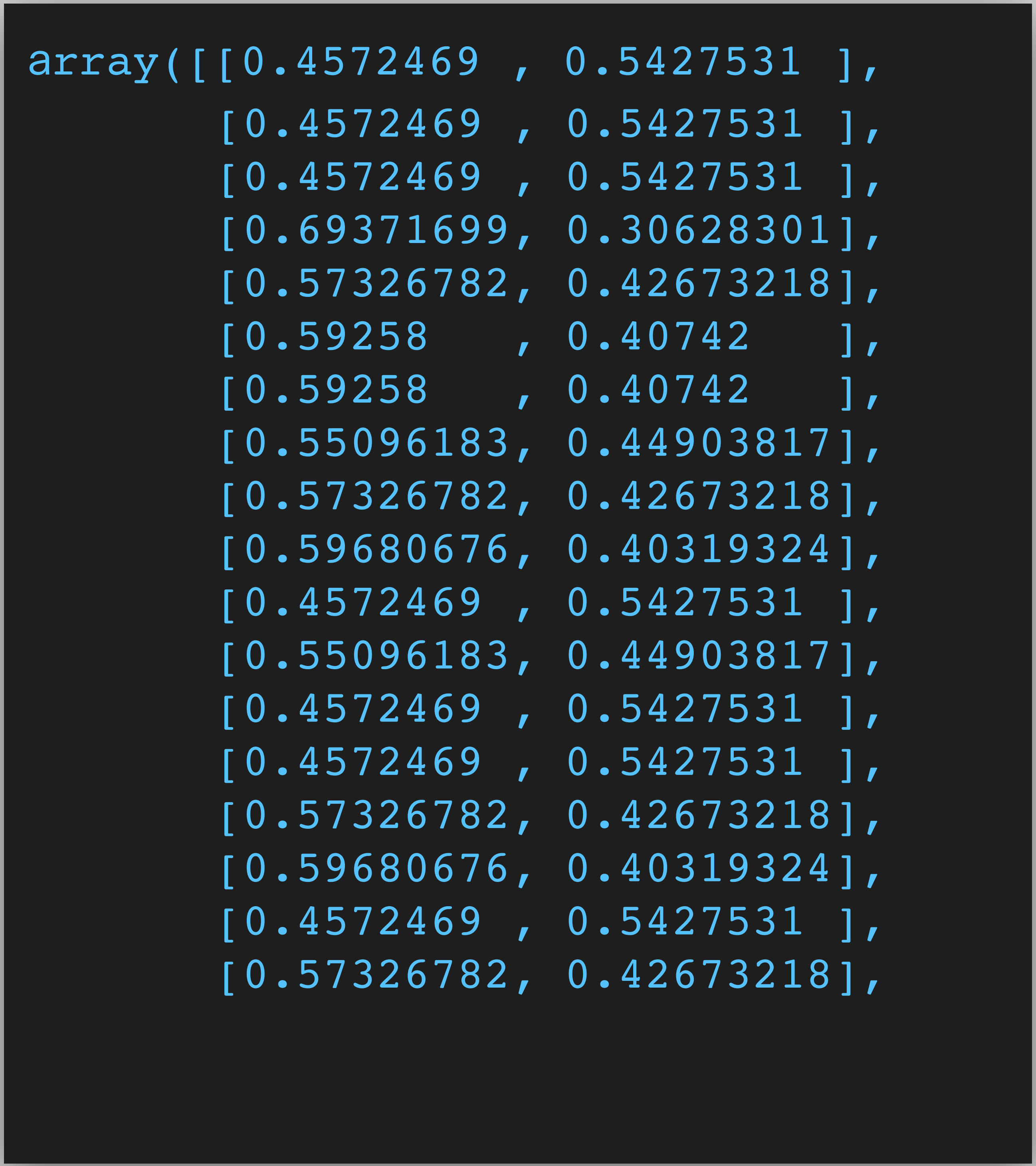
Main steps to train a classifier

5. Predict the results for the test set
6. Evaluate the performance of the model

👉 What will be the output of our model?

👉 What evaluation metrics we can use for a classification task?

Classification outputs



```
array([[0.4572469 , 0.5427531 ],
       [0.4572469 , 0.5427531 ],
       [0.4572469 , 0.5427531 ],
       [0.69371699, 0.30628301],
       [0.57326782, 0.42673218],
       [0.59258   , 0.40742   ],
       [0.59258   , 0.40742   ],
       [0.55096183, 0.44903817],
       [0.57326782, 0.42673218],
       [0.59680676, 0.40319324],
       [0.4572469 , 0.5427531 ],
       [0.55096183, 0.44903817],
       [0.4572469 , 0.5427531 ],
       [0.4572469 , 0.5427531 ],
       [0.57326782, 0.42673218],
       [0.59680676, 0.40319324],
       [0.4572469 , 0.5427531 ],
       [0.57326782, 0.42673218],
```

Once the model is fitted, we can predict the category of each molecule in the validation and test sets. A classifier outputs two numbers per each prediction:

— Probability of 0 (first column)

— Probability of 1 (second column)

Classification outputs

In a classification, the output is a probability. We need to define a cut-off or threshold to transform the results into a binary output (0 or 1) again. The threshold is typically set at 0.5 by default

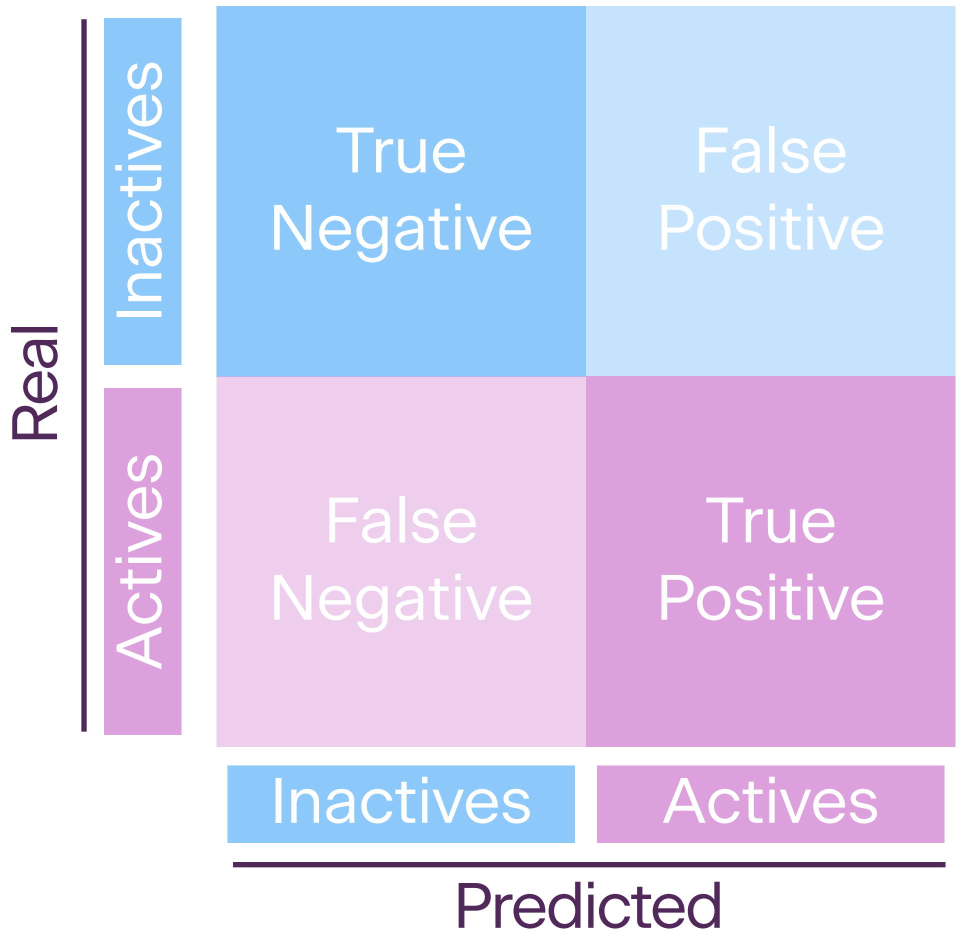
Proba 0	Proba 1	Cut-off: 0.5	Cut-off: 0.7	Cut-off: 0.3
0.39	0.61	1	0	1
0.3	0.7	1	1	1
0.69	0.31	0	0	1
0.21	0.79	1	1	1

Classification outputs

Original datasets: train, test
— X datasets: X_train, X_test
— Y datasets: Y_train, Y_test

Predictions:
— y_hat: output of the classifier, expressing the probability that a molecule is inactive (0) or active (1).
— y_hat_bin: binarized activity based on the predicted probability, the cut-off is set by the researcher.

Model Evaluation: Confusion Matrix

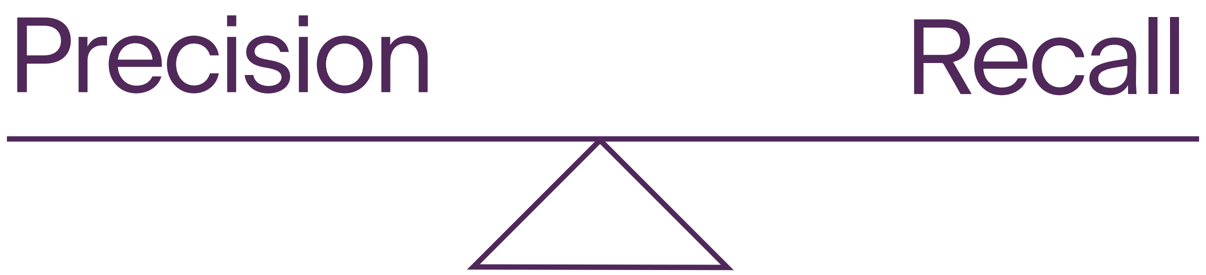


Precision: how many positives are actually positive

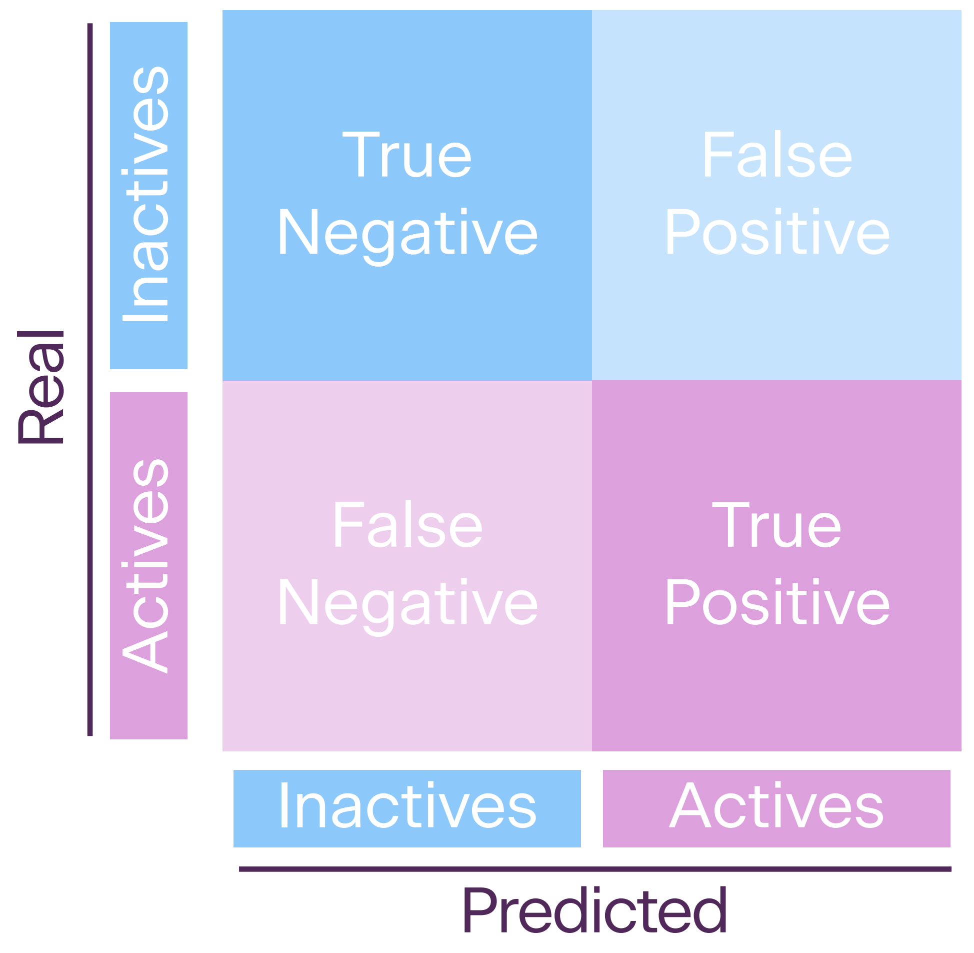
`TP / (TP+FP)`

Recall: how many positives are we able to identify

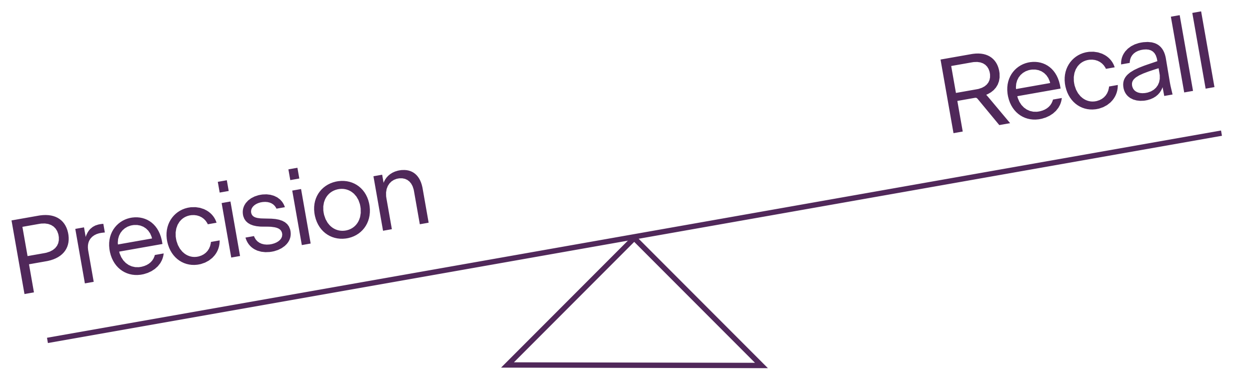
`TP / (TP+FN)`



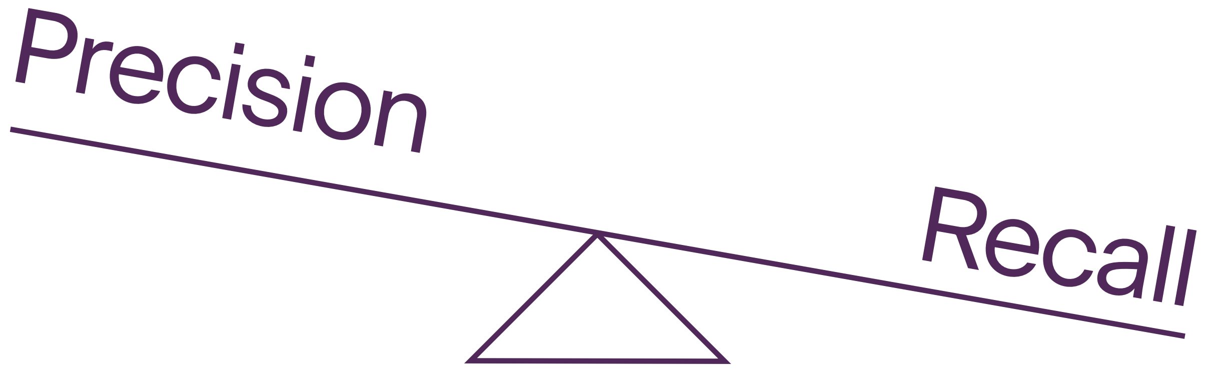
Model Evaluation: Confusion Matrix



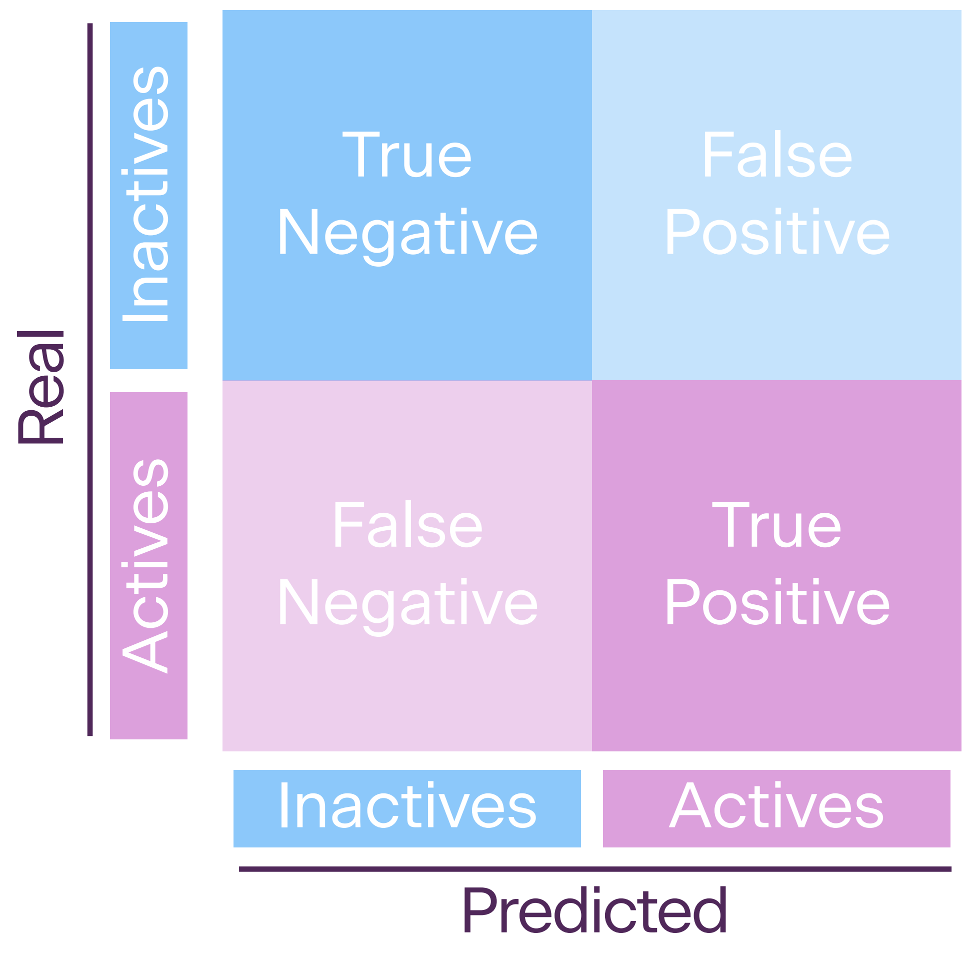
Higher, more restrictive threshold in proba1



Lower, less restrictive threshold in proba1



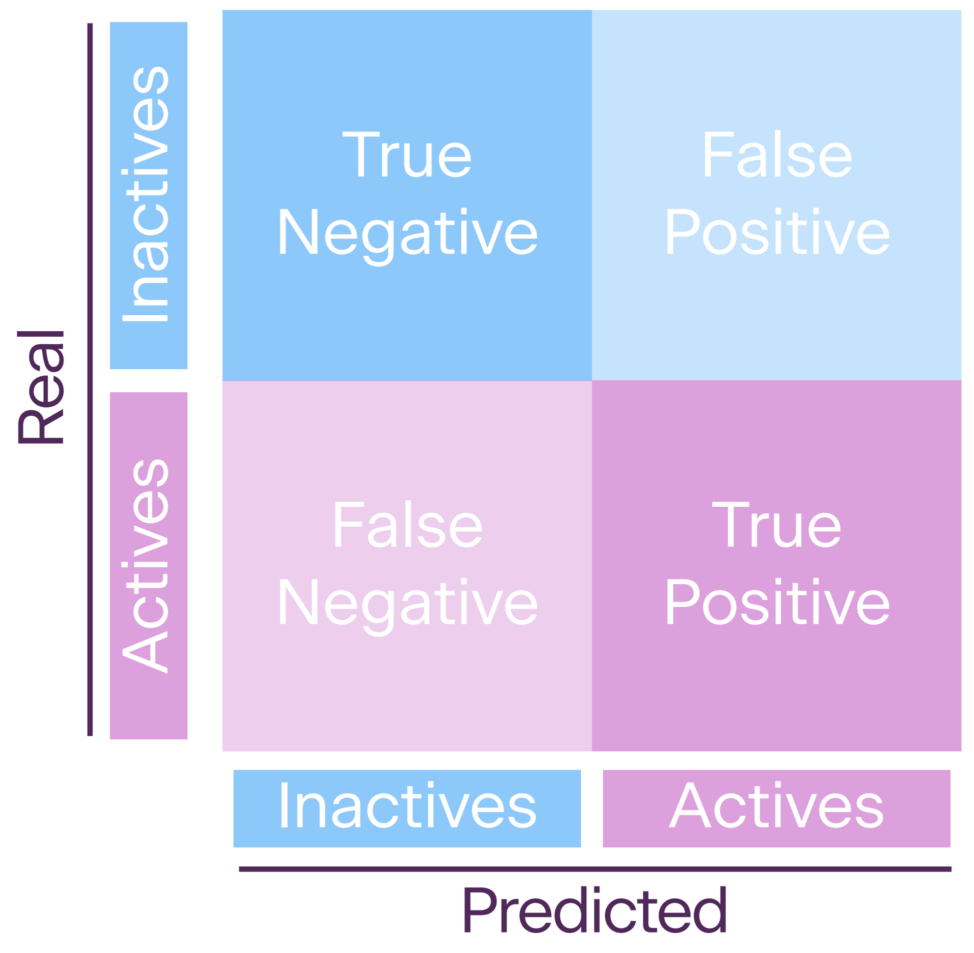
Model Evaluation: Confusion Matrix



👉 In which scenarios we might want a high precision?

👉 In which scenarios we might want a high recall?

Model Evaluation: ROC Curves



True Positive Rate (sensitivity or recall): proportion of correctly predicted positives of all positive observations

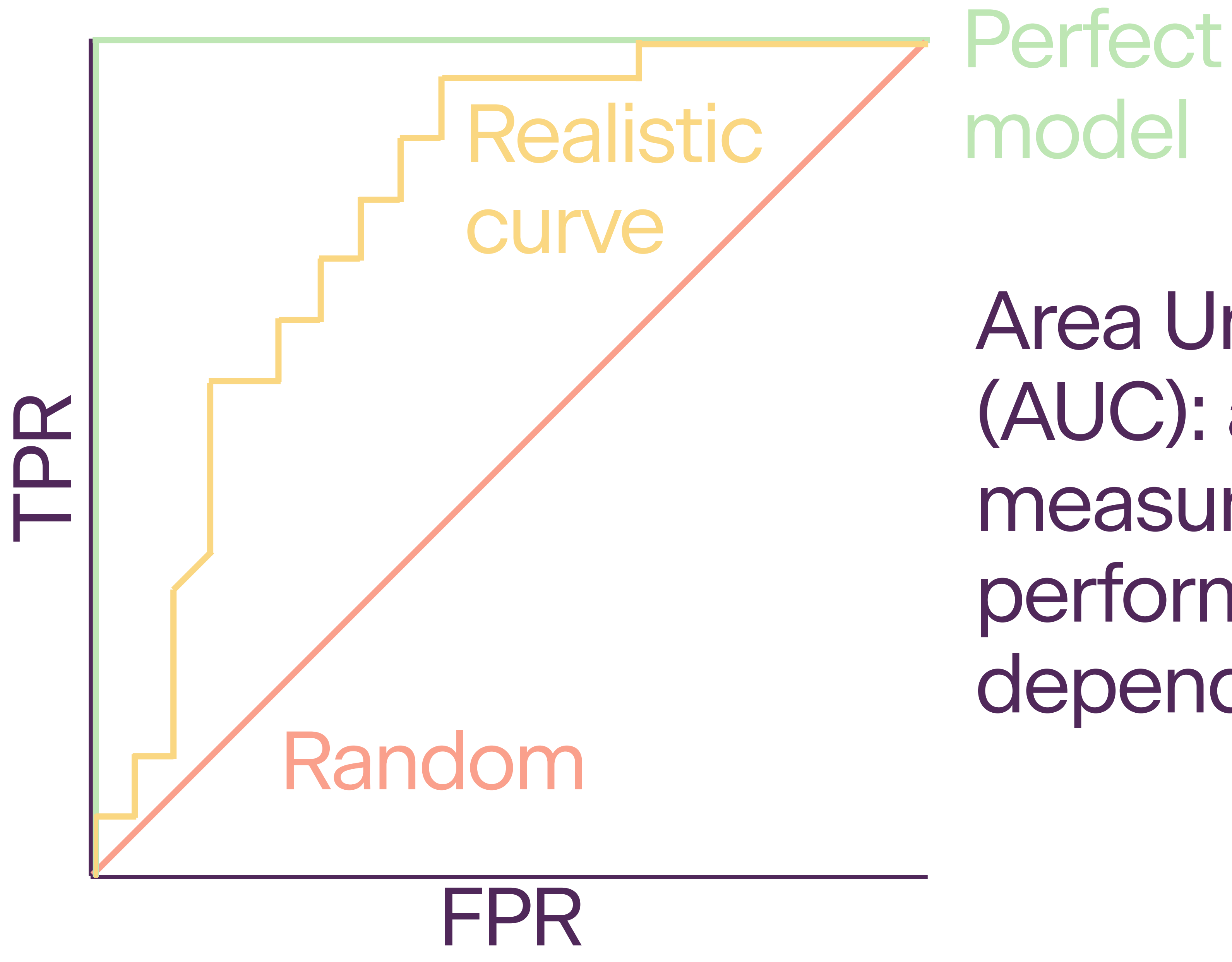
`TP / (TP+FN)`

False Positive Rate (100-sensibility): proportion of incorrectly predicted positives of all negative observations

`FP / (TN+FP)`

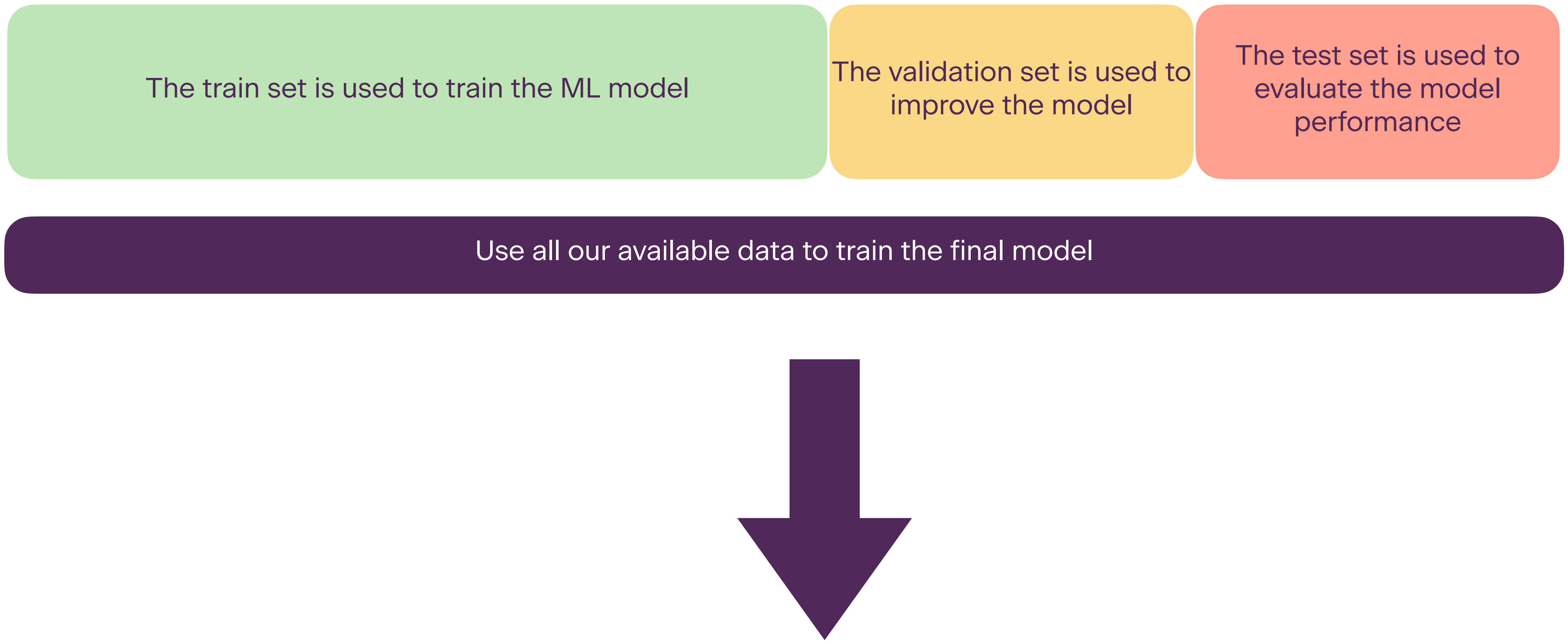
Model Evaluation: AUROC

ROC Curve: performance of the model at all classification thresholds (from 0 to 1)



Area Under the Curve (AUC): aggregate measures of model performance. It does not depend on the threshold

At the end of the pipeline, we will...



The train set is used to train the ML model

The validation set is used to improve the model

The test set is used to evaluate the model performance

Use all our available data to train the final model

Save the model to run predictions on new data. We can use it locally or deploy it online, for example in the Ersilia Model Hub

menti.org
4663 2789

https://ersilia.io
hello@ersilia.io
@ersiliaio