Supervised Learning Methods

Reminder: Data: Nomenclature

Features/Attributes (input variables)

Targets/Labels (output variables)

Examples/Instances

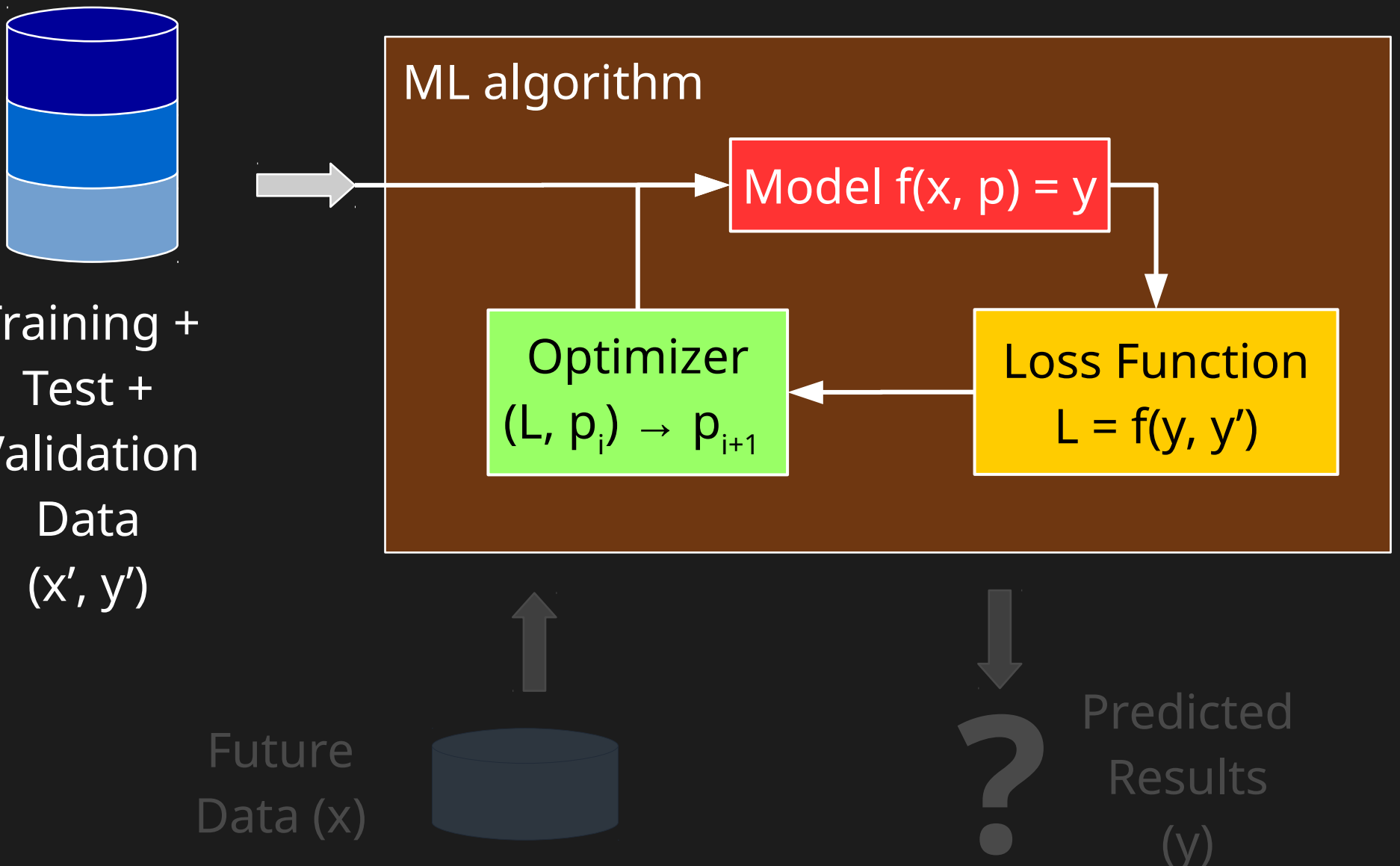
Weight	Height	Exterior	Wings?	Lika-bility	N_{legs}	Pet?	Type
0.1	0.1	feath.	true	1	2	true	bird
3.5	0.3	fur	false	1	4	true	cat
12.0	0.7	fur	false	1	4	true	dog
500	1.8	skin	false	2	4	false	rhinoceros
800	3.0	fur	true	3	4	false	chimera
2.5	0.5	fur	false	1	4	true	cat
...	...	...	...	...	...	...	...

classes of label "Type"

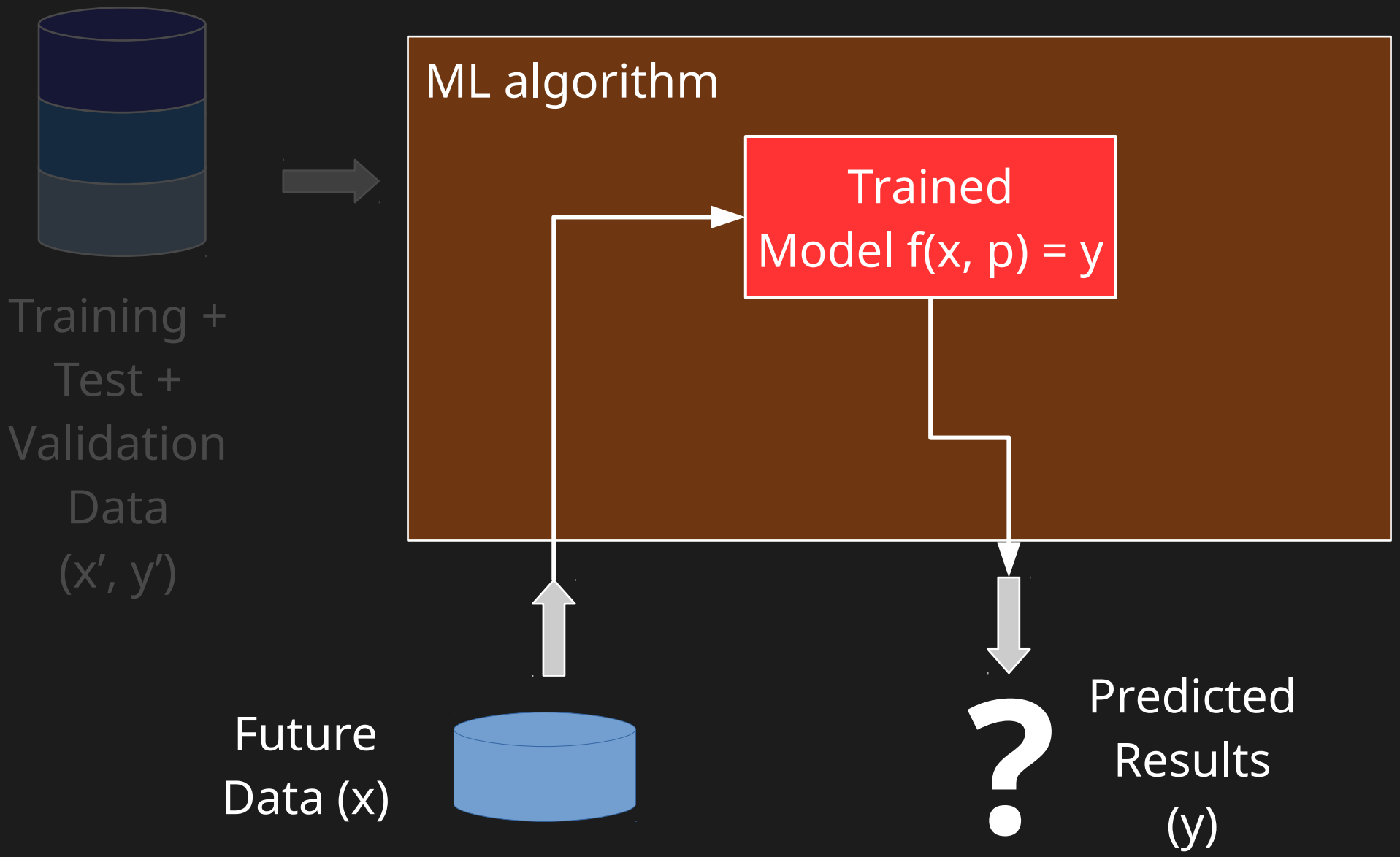
Data Type:

continuous	continuous	categorical	binary	ordinal	integer	binary	categorical (multi-class)

Reminder: Learning Stage

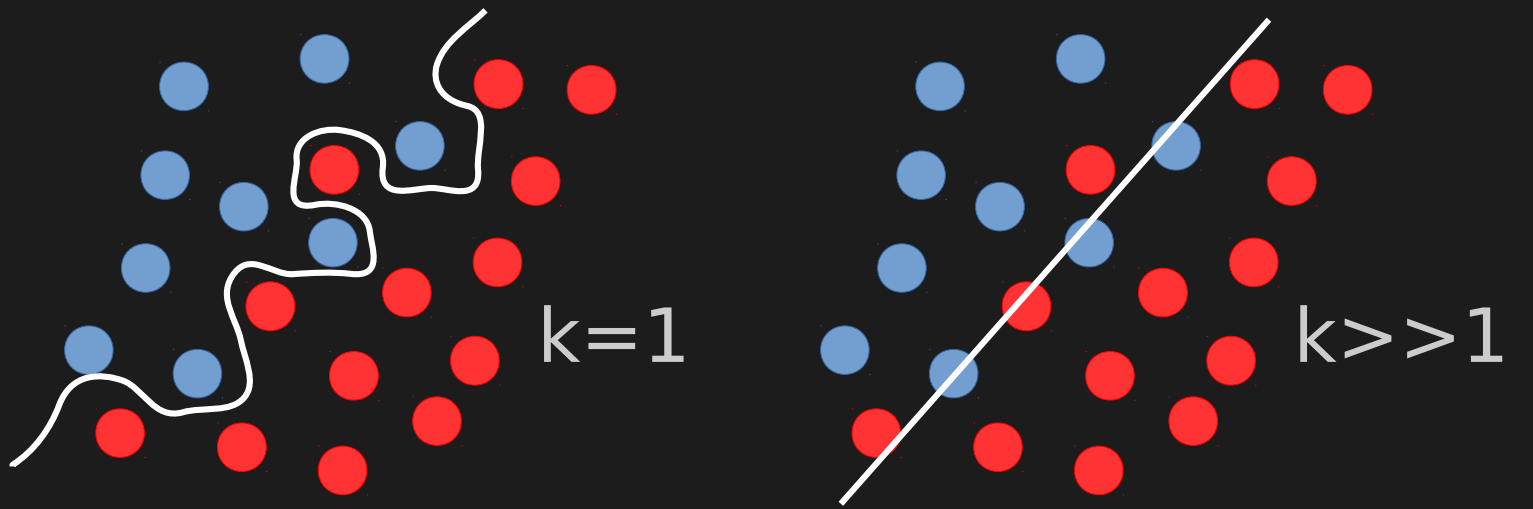


Reminder: Prediction Stage

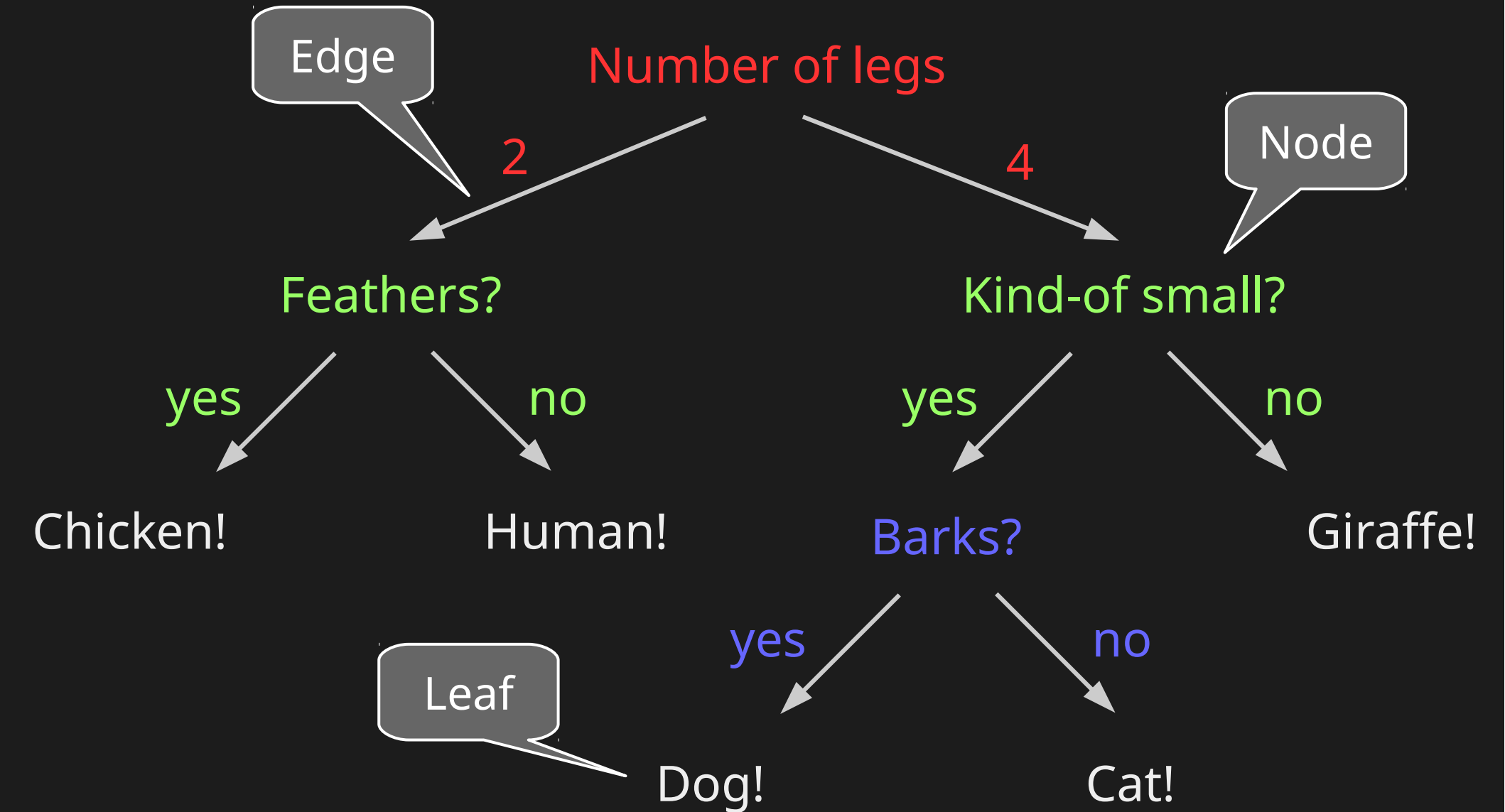


k Nearest Neighbors

- Non-parametric classifier/regressor
- Classification: output is class affiliation based on class majority among k nearest neighbors
- Regression: output is mean value of k nearest neighbors
- Hyperparameter k controls regularization



Decision Trees



Decision Trees

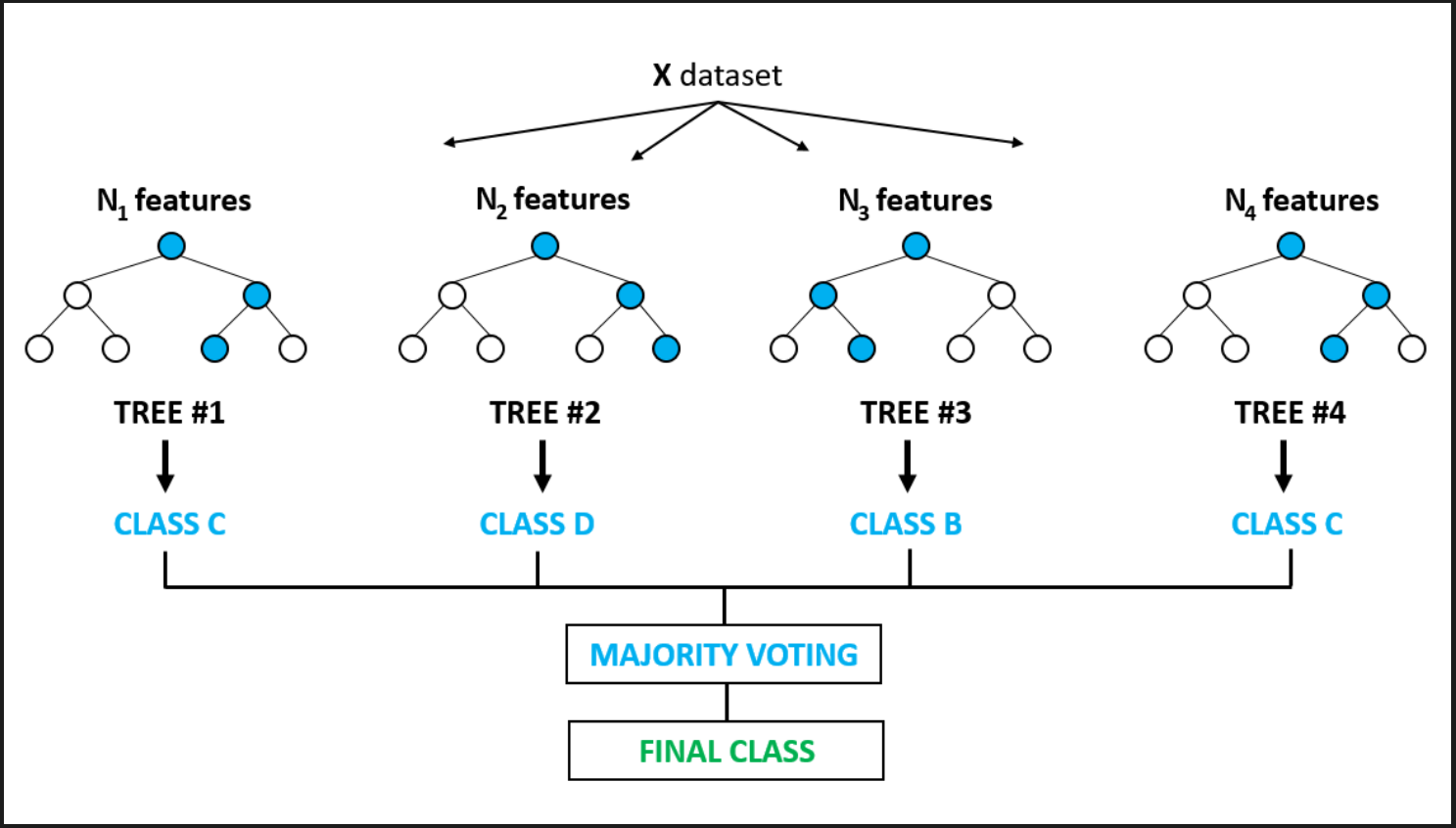
- Non-parametric model for classification and regression
- Hyperparameters (Complexity/Regularization):
  - maximum depth
  - maximum number of features
  - minimum number of samples for split
  - minimum number of samples for leaf
- Great interpretability but unlikely to generalize well

Ensemble Models

- Combine the results from different models to compensate for shortcomings
- Final results through averaging, weighted averaging, or majority vote
- Individual models are “weak models” with low complexity

Random Forest

- Combination of N decision trees
- Good generalization by keeping depths and number of features low



The Fancy Stuff: Gradient-Boosted Tree-based Models

- Build individual trees in a random forest such that THEIR SUM minimizes the loss function
- Models like xgboost or lightGBM – if tuned properly – can perform as well as deep learning models
- Python implementations for both models have sklearn interfaces – you can use them like all the other models we used