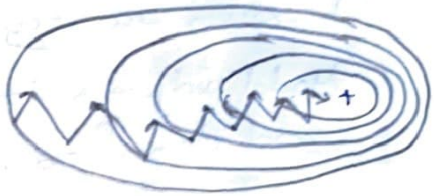
⊙ Stochastic gradient descent (SGD)

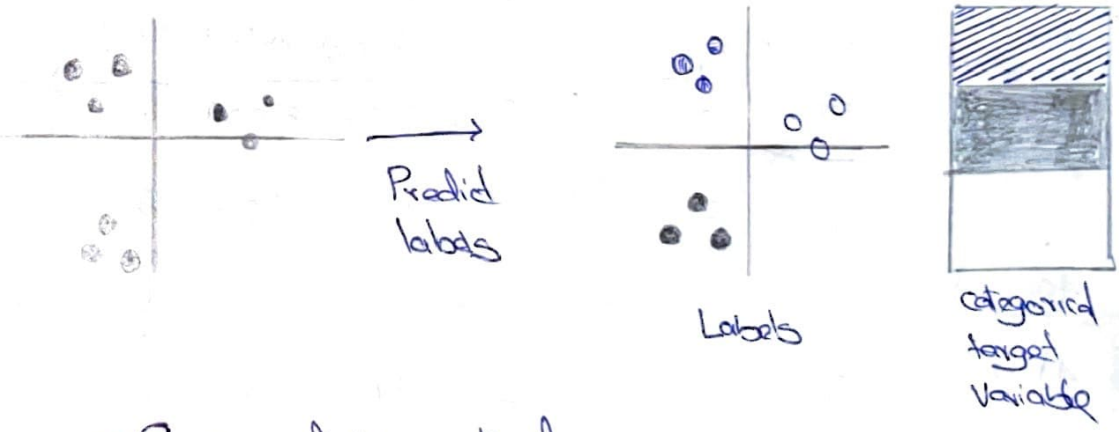
- Variation of the gradient descent
- Uses a random data subset and scales well
- Likely to overlook local and find global minima
- Converges quickly towards a global minimum



- Convergence can be improved by:
  - Decreasing learning rate
  - Gradually increasing sample size

Module 3

⊙ Classification



- Supervised ML method
- Uses fully trained models to predict labels on new data
- Labels from a categorical variable with discrete values.

- What is supervised learning
  - Understands data in context when answering a question
  - Ensures accuracy in Prediction
  - Model adjusts the data to fit the algorithm and classifies it accordingly
- Application of classification
  - Problems expressed as association between feature and target variables
  - Used to build apps for:
    - Email filtering
    - Speech-to-text
    - Handcorting recognition
    - Biometric identification
    - Document Classificatio

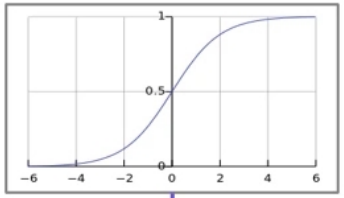
Load default Prediction.

Use cases of classification

Loan default prediction

age	ed	employ	address	income	debtinc	creddebt	othdebt	default
41	3	17	12	176	9.3	11.359	5.009	1
27	1	10	6	31	17.3	1.362	4.001	0
40	1	15	14	55	5.5	0.856	2.169	0
41	1	15	14	120	2.9	2.659	0.821	0
24	2	2	0	28	17.3	1.787	3.057	1
41	2	5	5	25	10.2	0.393	2.157	0
39	1	20	9	67	30.6	3.834	16.668	0
43	1	12	11	38	3.6	0.129	1.239	0
24	1	3	4	19	24.4	1.358	3.278	1
36	1	0	13	25	19.7	2.778	2.147	0

Model Training



age	ed	employ	address	income	debtinc	creddebt	othdebt	default
37	2	16	10	130	9.3	10.23	3.21	0

Trained Classifier

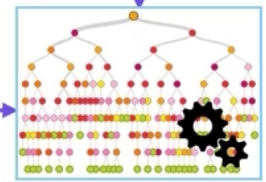
Prediction

Multiclass drug prescription

Age	Sex	BP	Cholesterol	Na	K	Drug
23	F	HIGH	HIGH	0.793	0.031	drugY
47	M	LOW	HIGH	0.739	0.056	drugC
47	M	LOW	HIGH	0.697	0.069	drugC
28	F	NORMAL	HIGH	0.564	0.072	drugX
61	F	LOW	HIGH	0.559	0.031	drugY
22	F	NORMAL	HIGH	0.677	0.079	drugX
49	F	NORMAL	HIGH	0.79	0.049	drugY
41	M	LOW	HIGH	0.767	0.069	drugC
60	M	NORMAL	HIGH	0.777	0.051	drugY
43	M	LOW	NORMAL	0.526	0.027	drugY

Categorical Variable

Modeling



Classifier

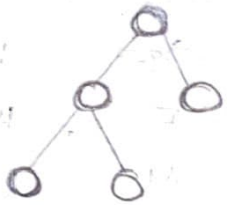
Age	Sex	BP	Cholesterol	Na	K	Drug
36	F	LOW	HIGH	0.697	0.069	DrugX

Prediction

Predicted Labels

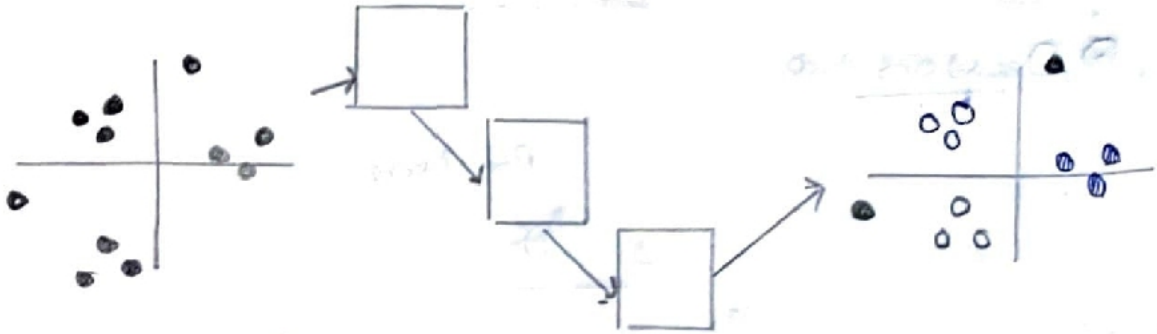
- Classification algorithms
  - Naive Bayes
  - Logistic regression
  - Decision trees
  - K-nearest neighbors
  - Support Vector Machine
  - Neural networks

- Multiclass Prediction:
  - Classification algorithms used as components for multiclass classifi
  - strategies:
    - One-versus-all
    - One-versus-one

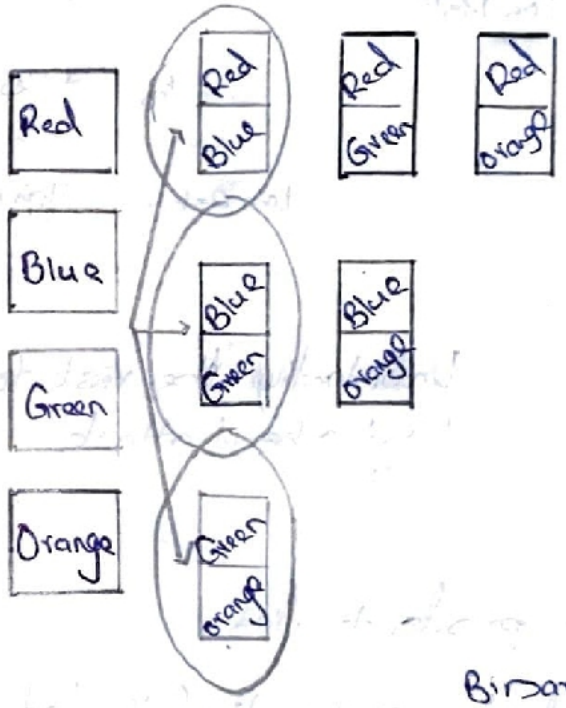


- One-versus-all strategy
  - Binary classifier: One for each class label
  - Assigned a single label that define target class.
  - Task: Binary Prediction for every data point for a one-versus-others trained classifier
  - K-classes = k binary classifiers

Red	→ Red classifier	→ Is it Red?
Blue	→ Blue classifier	→ Is it Blue?
Green	→ Green classifier	→ Is it Green?
Orange	→ Orange classifier	→ Is it Orange?
classes		Binary classifier



2. One-versus-one strategy



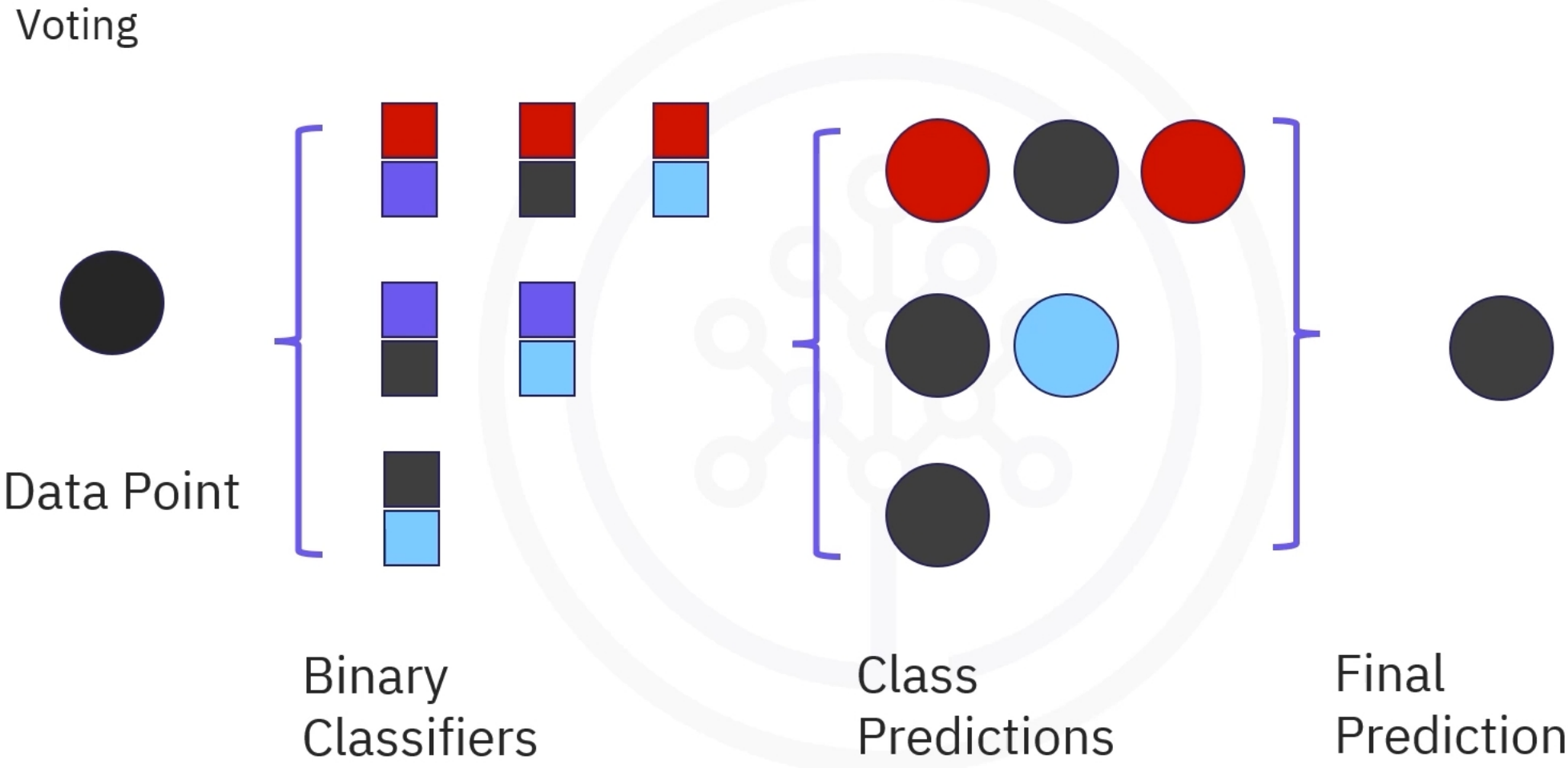
→ Is it Red or is it Blue?

→ Is it Blue or is it Green?

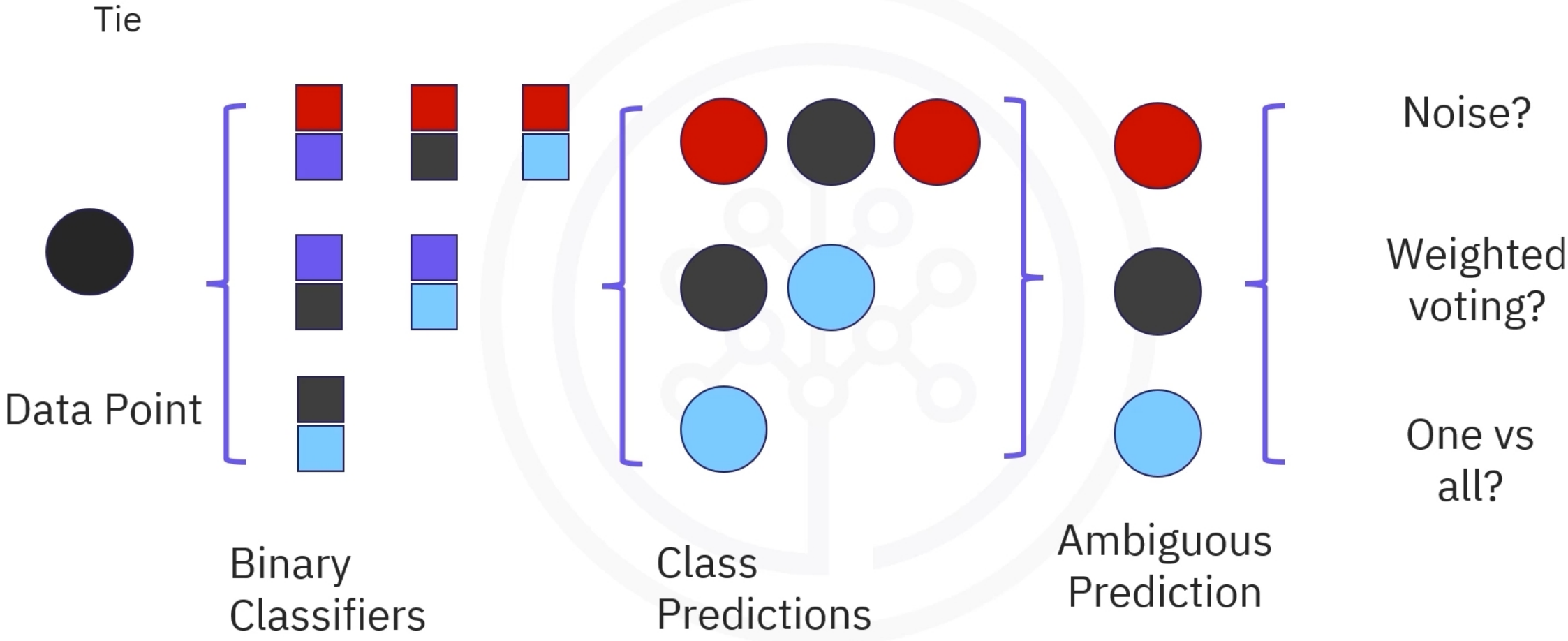
→ Is it Green or is it Orange?

Binary classifiers

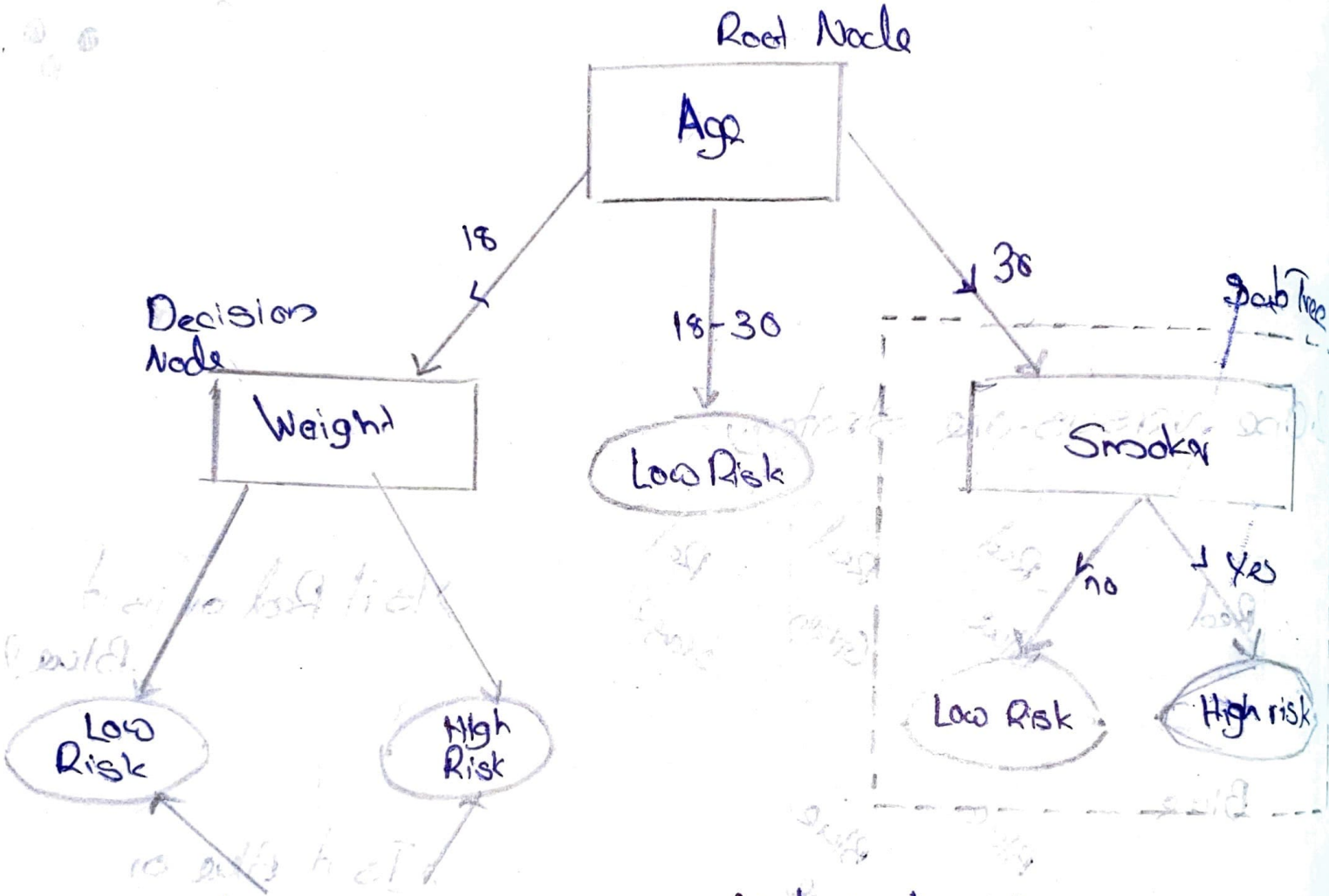
One-versus-one strategy



One-versus-one strategy



① Decision tree

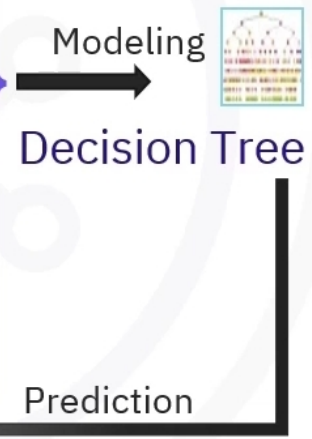


Understanding the risk to Prevent a heart attack.

- Each internal node corresponds to a test
- Each branch corresponds to the result of the test
- Each terminal, or leaf node, assigns it data to a class

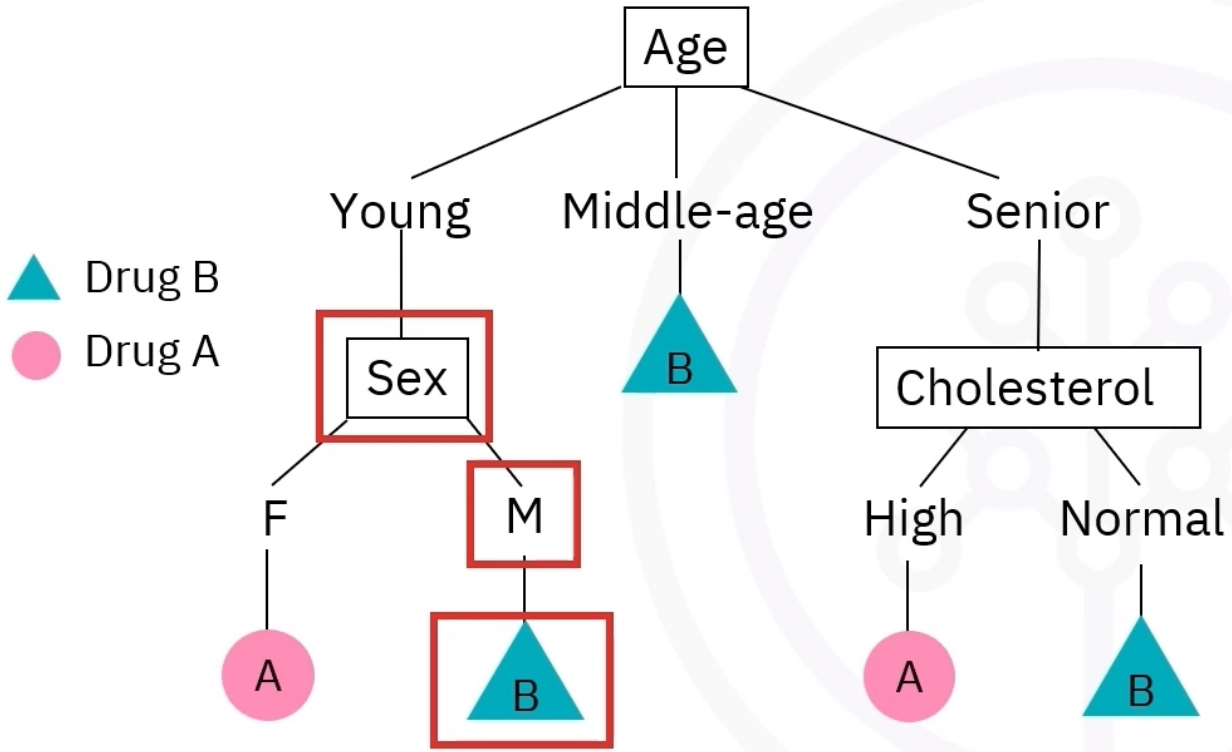
How to build a decision tree?

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	High	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A
p15	Middle-age	F	Low	Normal	?



- Consider features of a data set
- Example in medical study: Age, sex, blood pressure, and cholesterol
- Use training part of data set to build decision tree
- Use decision tree to predict class of unknown patient

Patient classifier example



Decision to prescribe Drug A or B based on historical data

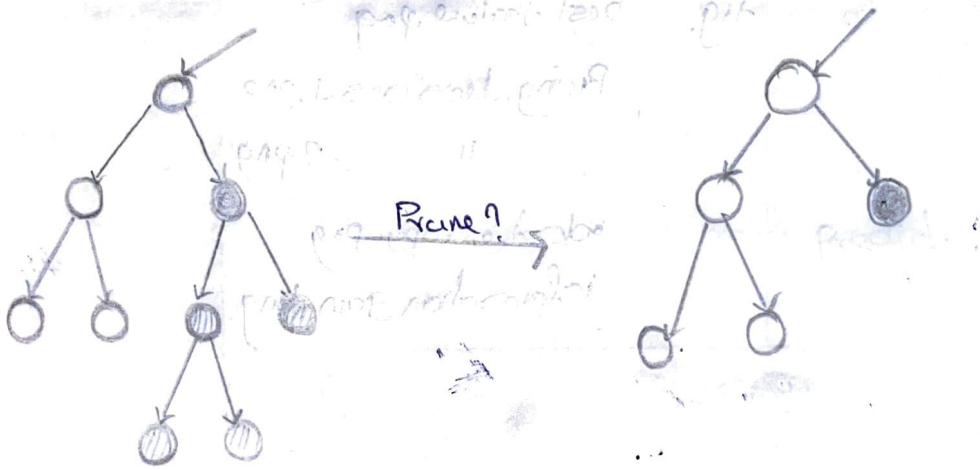
For middle-aged patient, decision tree suggests Drug B

For young female, decision tree suggests Drug A

o Training a decision Tree.

1	• Start with a seed node and labeled training data
2	• Find the feature that best splits the data
3	• Each split partitions the node's input data
4	• Repeat the process for each new node

⊙ Tree Pruning:



Stop growing the tree when:

→ Maximum tree depth is reached

→ Minimum number of datapoints in a node has been exceeded

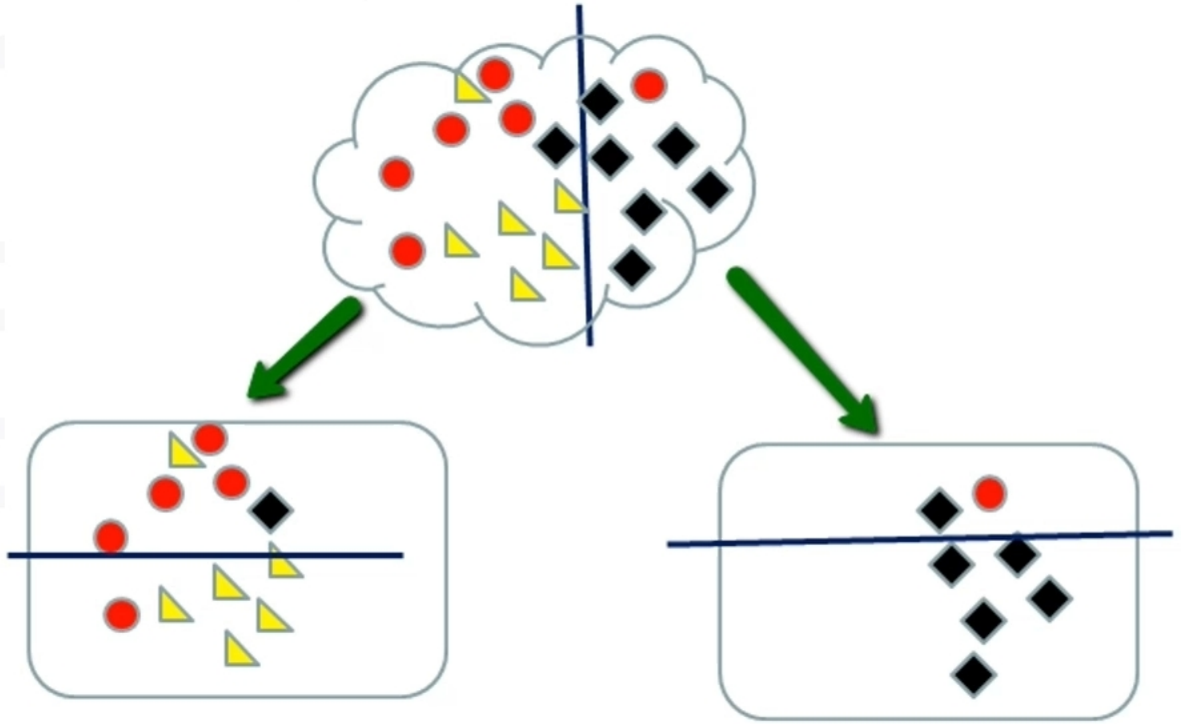
→ Minimum number of samples in a leaf has been exceeded

→ Decision tree has reached maximum number of leaf nodes

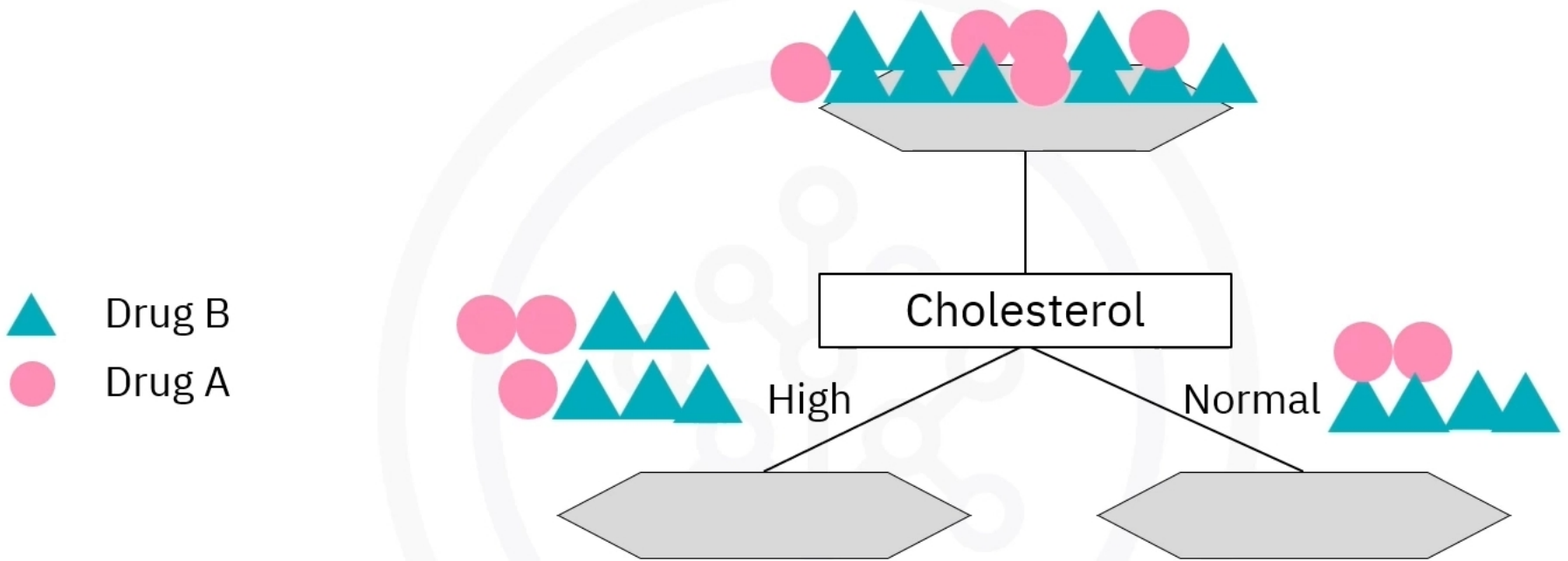
- Why Prune?
  - Pruning simplifies decision tree
  - Pruned tree is more concise and easier to understand
  - Pruning results in better Predictive accuracy

Which is the best feature?

- Decision trees are trees built using recursive partitioning to classify data
- Select feature that best split data to train the tree
- Common split measures are:
  - Information gain
  - Gini impurity

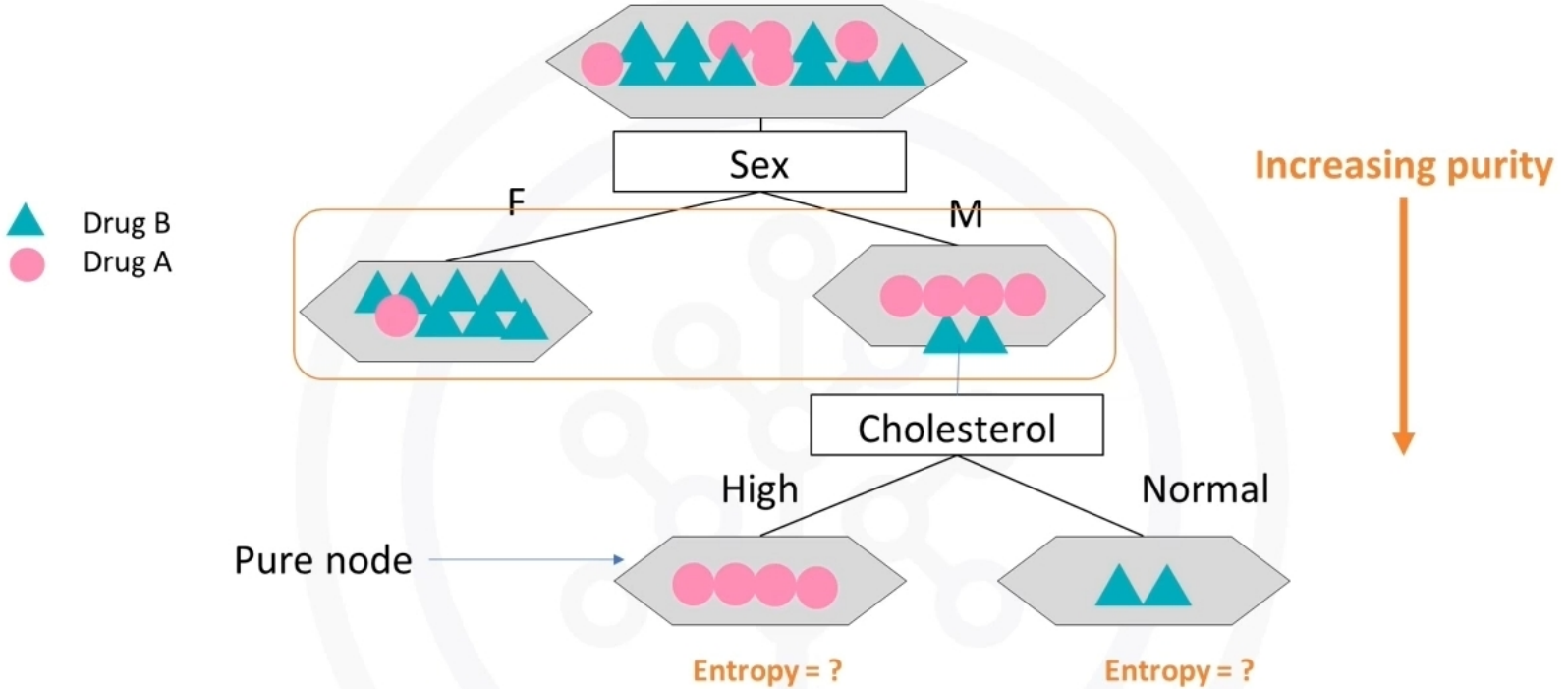


Pruning decision tree example



Test cholesterol as the first feature

The tree assigns patients to two nodes: High and Normal



Test patient sex as the second feature

Continue branching until stopping criterion

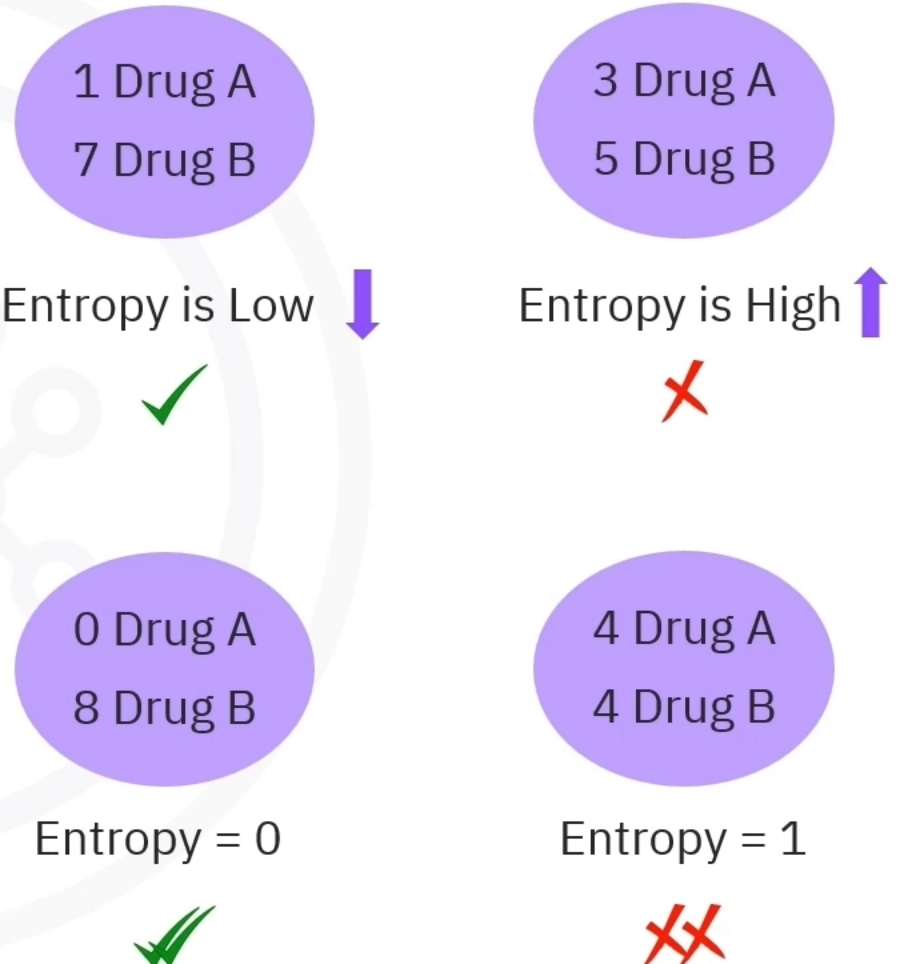
What is entropy?

Measure of information disorder in a data set

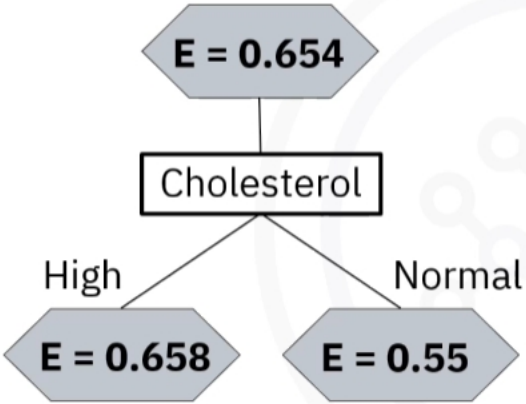
Measures how random the classes in a node are

If the classes are completely homogeneous, entropy = 0

If the classes are equally divided, entropy = 1



What is information gain?



Entropy of a tree before split - Weighted entropy after split

Opposite of entropy

Increases with the decrease in entropy

- Advantages of decision trees.
  - Model Visualization
  - Interpretability
  - Analysis and Prediction

⊙ Conclusion.

- In a decision tree.
  - Each internal node corresponds to a test
  - Each branch corresponds to the result of the test
  - Each terminal, or leaf node, assigns its data to a class

- A decision tree is an algorithm for classifying data points
- Decision tree are built by considering data set features

⊙ Regression tree

- Analogous to a decision tree that predicts continous values
- Classification: Target is categorical
- Regression:

  Target is continous

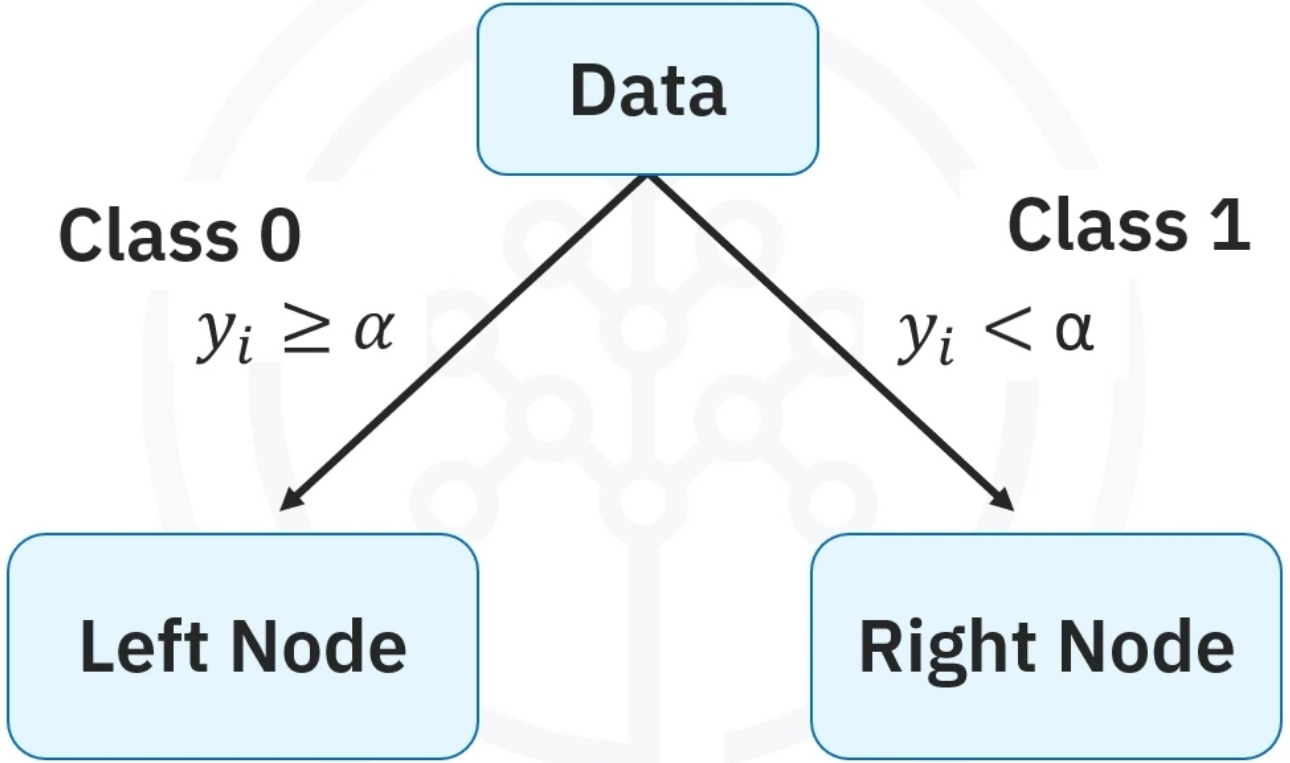
- Regression tree: Decision tree adapted to solve regression Problems

Classification versus regression trees

	Classification Trees	Regression Trees
Objective	Classify data into discrete sets	Predict continuous target variable
Target Variable	Categorical	Float
Splitting Criterion	Gini impurity or entropy	Variance reduction
Prediction at Leaf Nodes	Class label majority vote	Average value of target values
Example Use Cases	Spam detection, image classification, medical diagnosis	Predicting revenue, temperatures, wildfire risk

- Creating regression trees.
  - Recursively split data set into subset to maximize information gain
  - Generate a tree like structure
  - Minimizes random of classes assigned to split nodes

Creating regression trees



Predicting values

Predicted value:

- Mean of target values

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Alternative value:

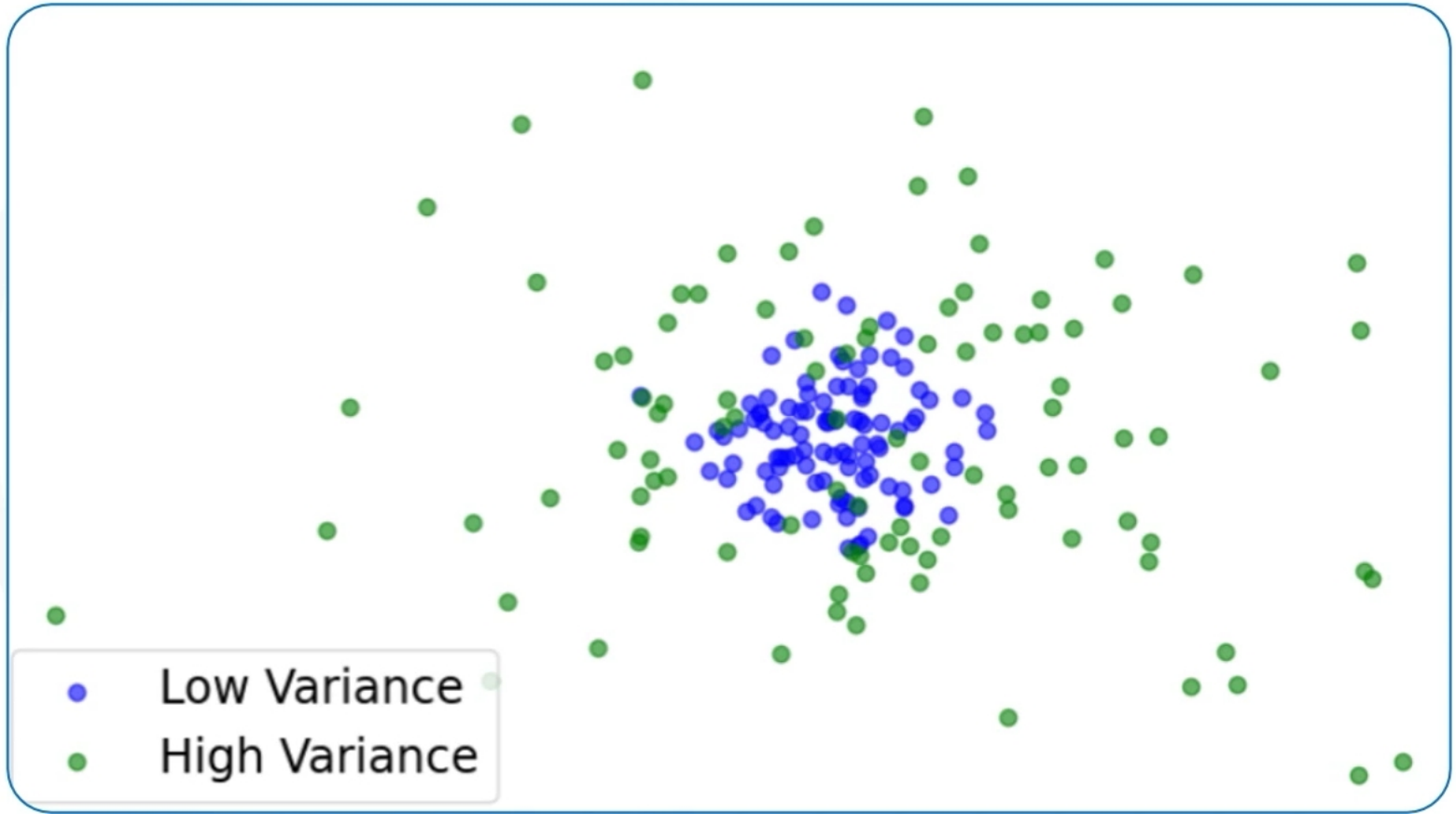
- Median of target values
- Better for skewed data
- More expensive

Splitting criterion

Features that minimize error between actual and predicted value

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$$

A measure of target variance



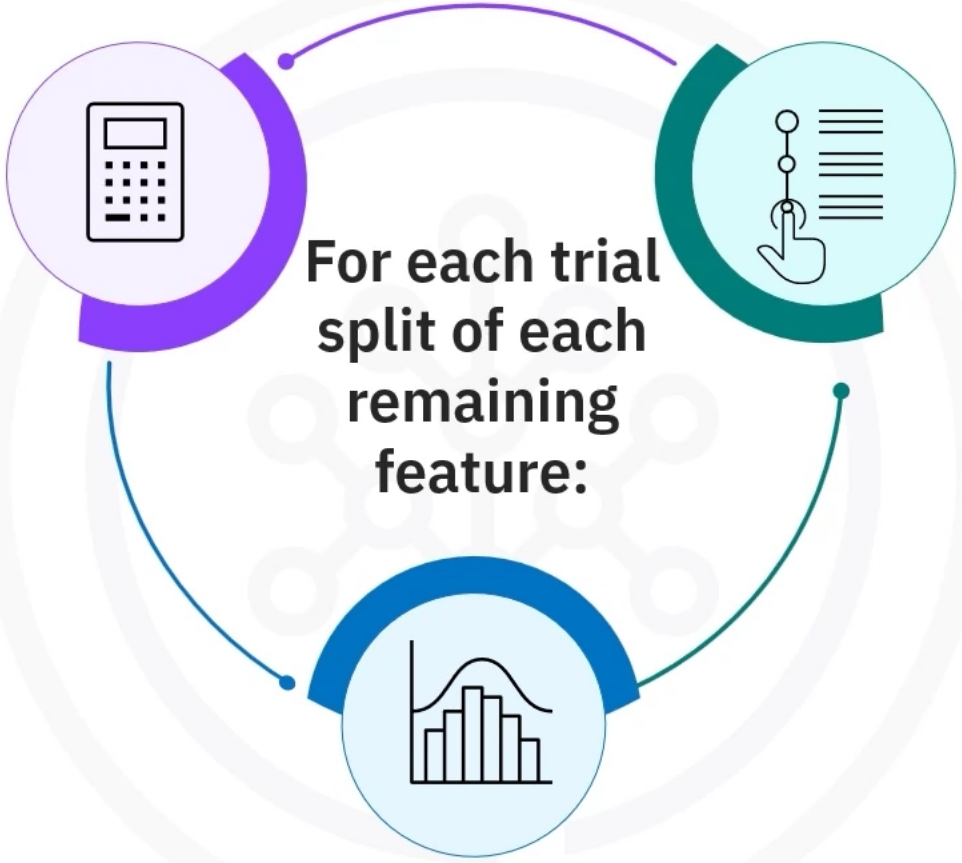
Quality of split

Weighted average of MSEs of each split:

$$\mathrm{MSE}_{Avg} = \frac{1}{\mathrm{N}_{Total}} \left(\mathrm{N}_{Left} * \mathrm{MSE}_{Left} + \mathrm{N}_{Right} * \mathrm{MSE}_{Right} \right)$$

Choosing the best split

Calculate MSE for left and right nodules



Select split with smallest value

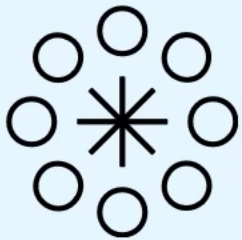
Calculate weighted average of MSEs

Categorical feature splits

Binary feature

- Separate into two classes
- Calculate weighted average of MSEs
- No minimization needed

Multiclass feature



- Use one-vs-one or one-vs-all
- Calculate weighted average for each binary split
- Select split that minimizes weighted MSE

Continuous feature trial thresholds

Sort feature values:
$x_i \leq x_j$ for all $i < j$

Drop duplicates:
$x_i < x_j$ for all $i < j$

Define midpoint thresholds:
$\alpha_i = \frac{x_i + x_{i+1}}{2}$

Choose α that minimizes weighted MSE

For large datasets, select a sparse subset

Assumption: Target values are uniformly distributed

Consider distribution when sampling thresholds