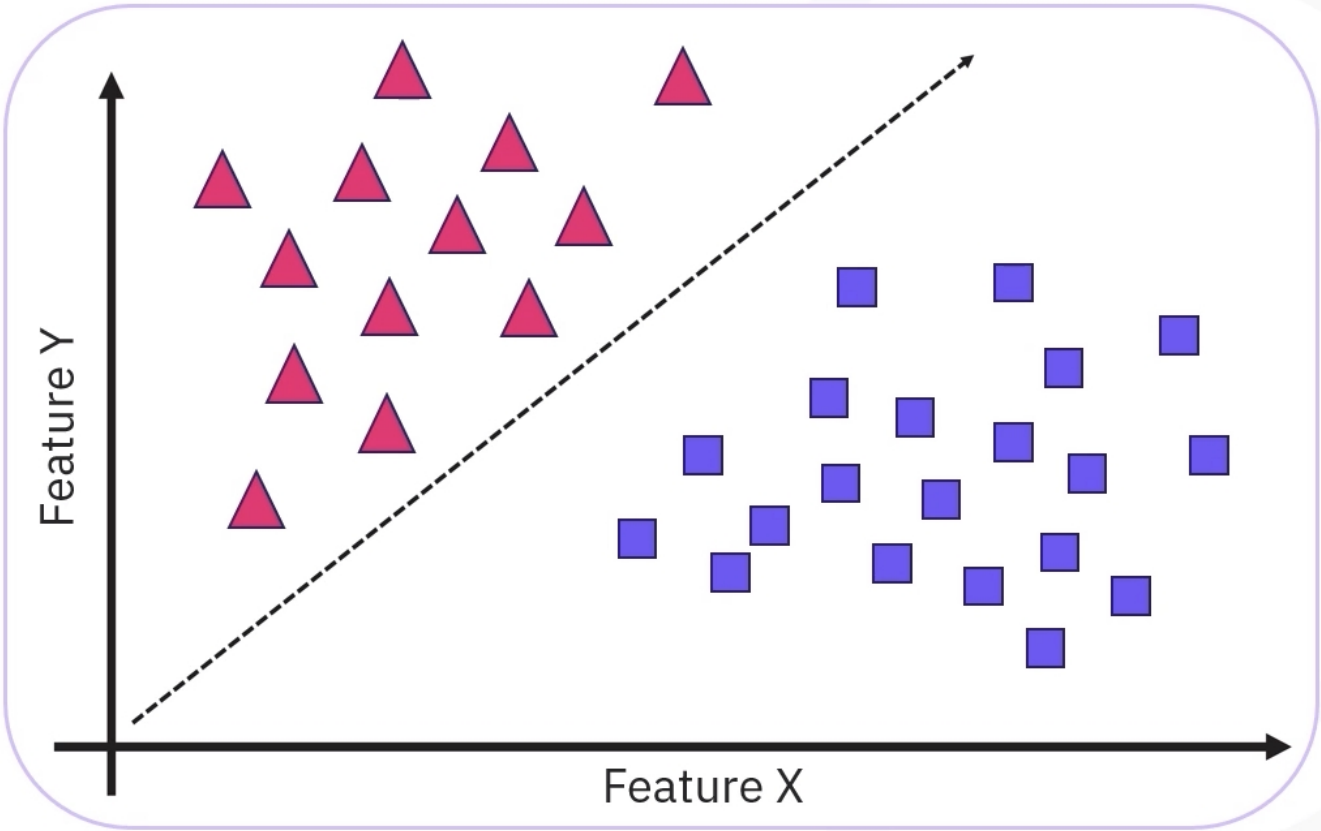
- Supervised Learing coith SVMs

  - Supervised learning technique
  - Used for building dassification and regression models
  - Classifies input by identifying hyperplane

SVM goals and objectives



Create a hyperplane that segregates data set

Incorporate soft margin

Tolerate misclassifications while maximizing the margin

Create a hyperplane that segregates data set

↓

Incorporate soft margin

↓

Tolerate misclassifications while maximizing the margin

↓

Balancing between maximizing margin controlled by Parameter C

↓

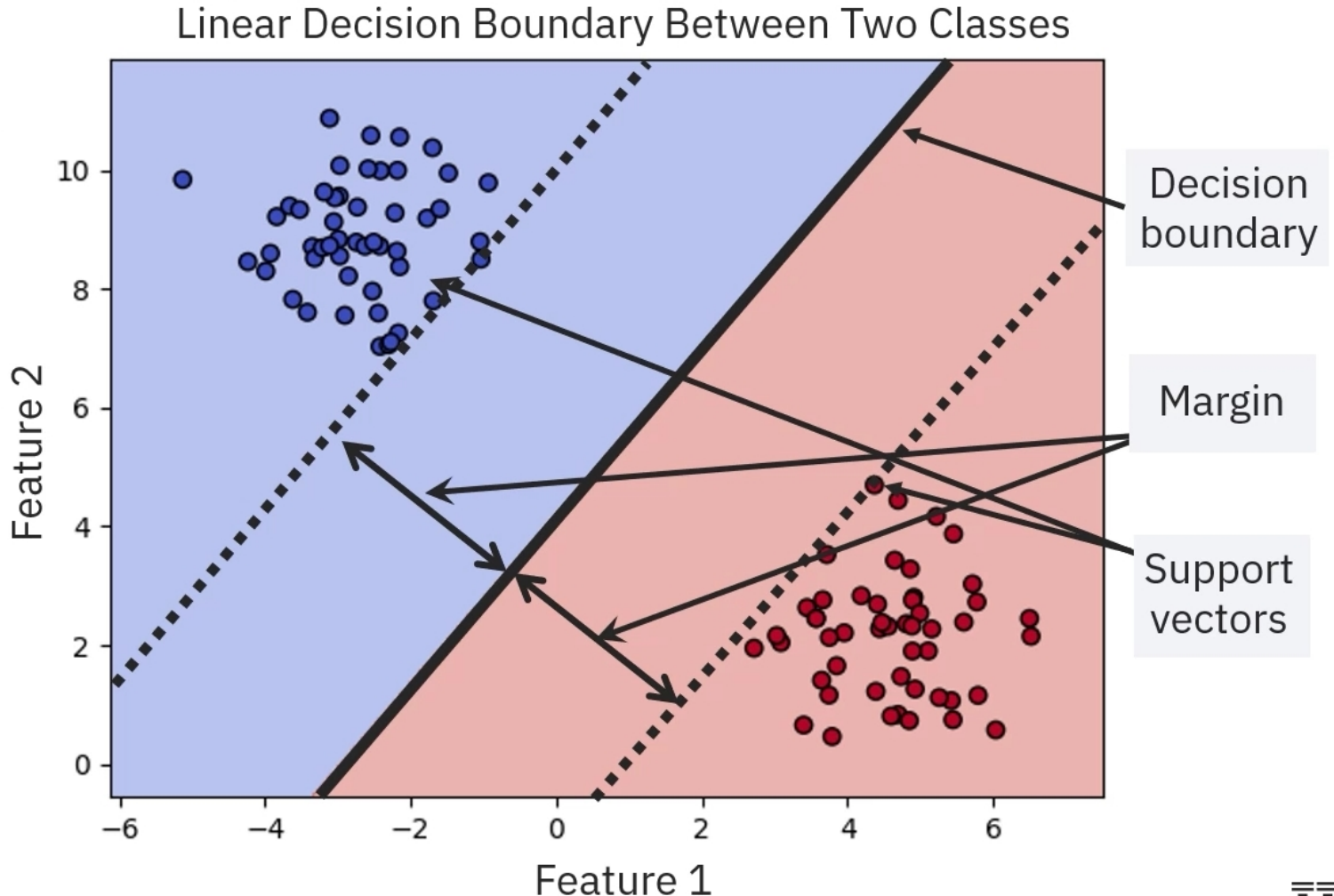
Smaller C allows more misclassificatio (softer margin)

↓

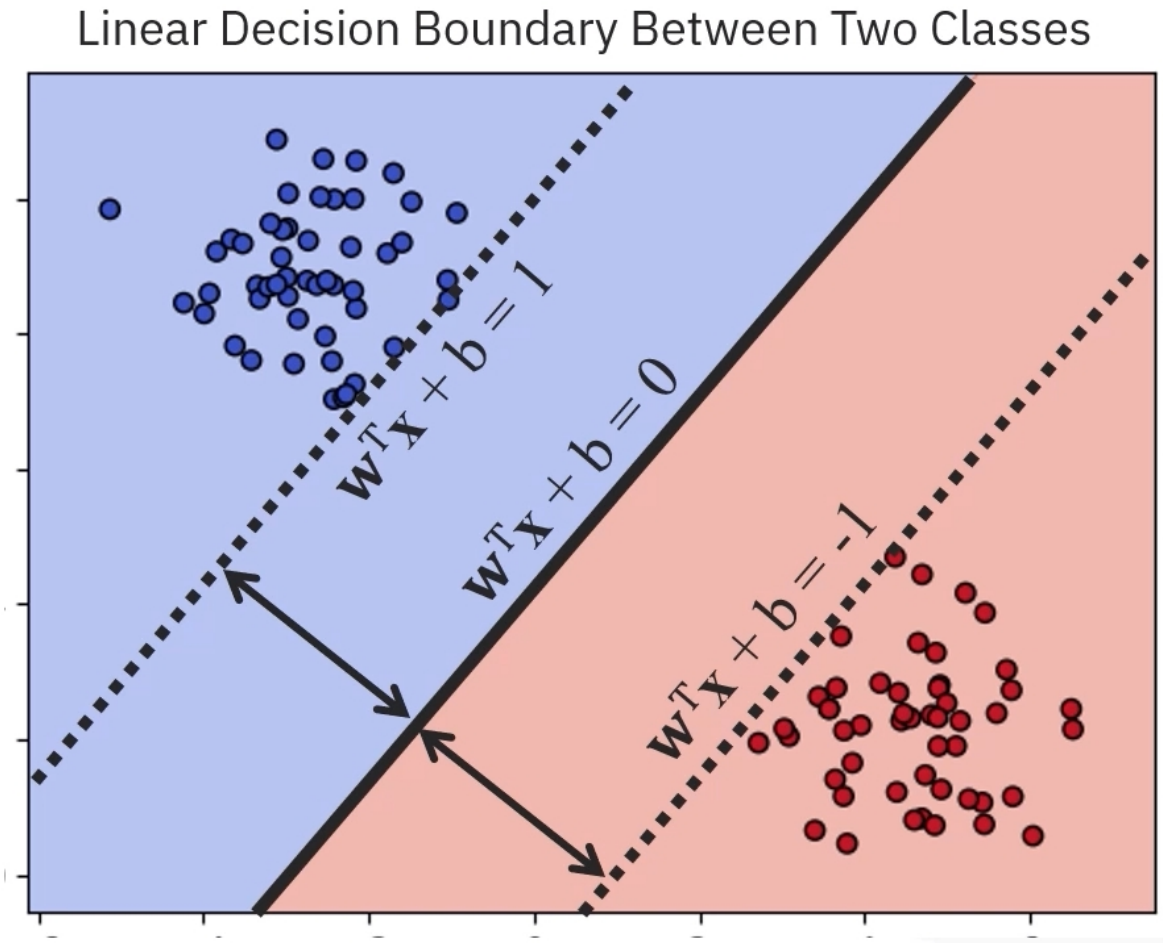
Large C forces a stricter separatio (harder margin)

How SVMs work

- Binary classification machine learning algorithms
- Divide data into two classes by finding decision boundary
- Decision boundary is hyperplane that maximizes margin



SVM training and prediction



- 2D example
- Find a weight vector and value b, called the bias term

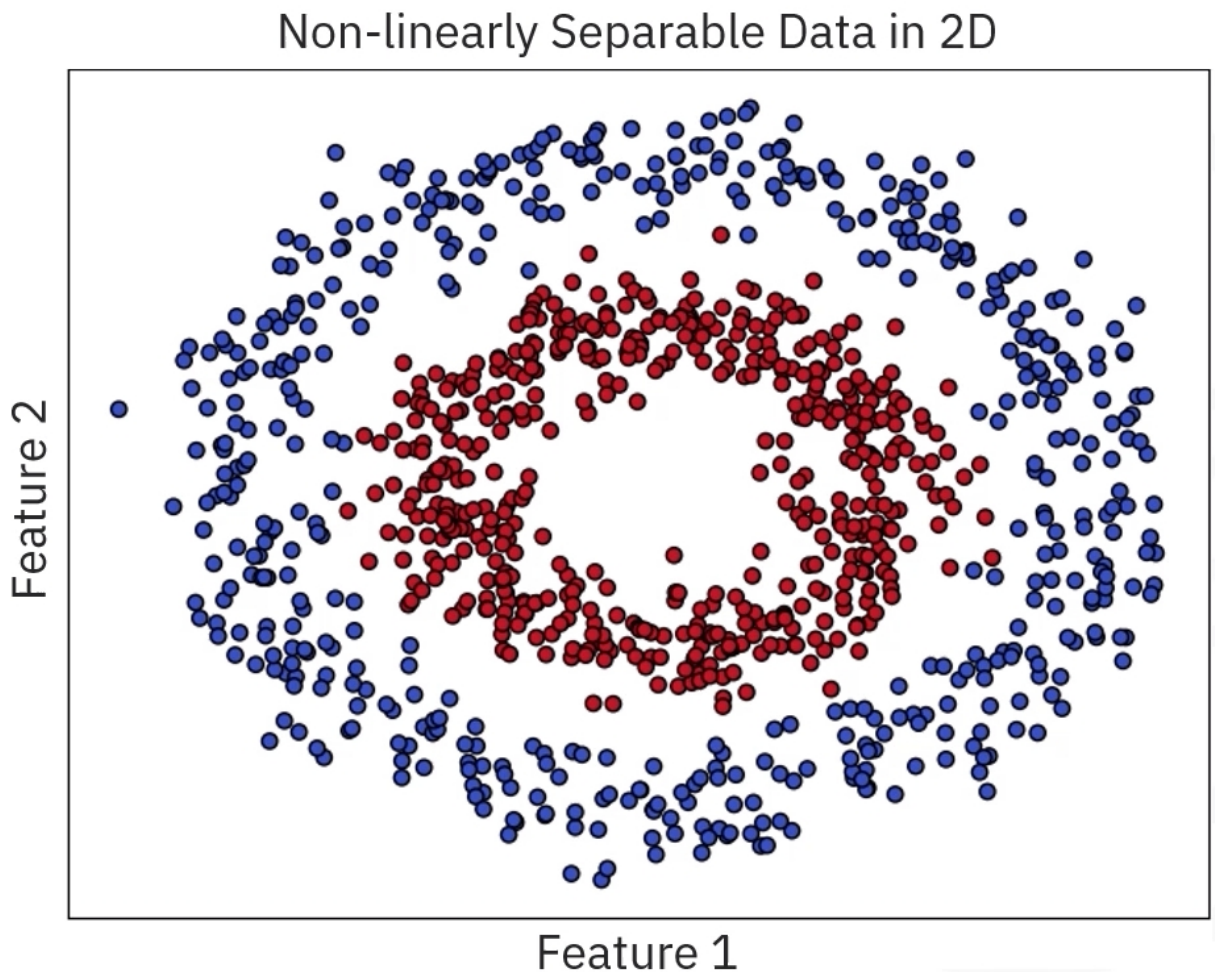
If equation > 0 :

- Point belongs to the blue class

Else:

- Point belongs to the red class

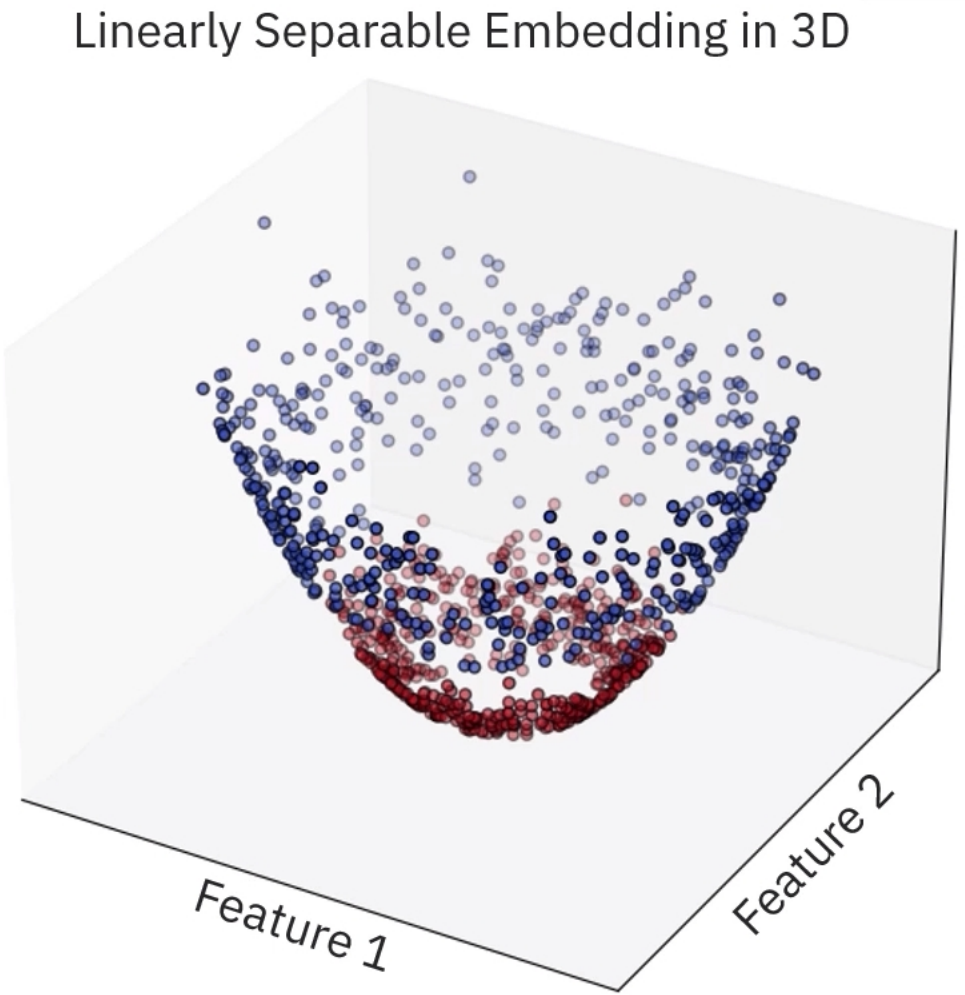
Nonlinearly separable classes



- 2D object
- Two non-overlapping classes
- Imagine these points as map contours

x_1 = Feature 1
x_2 = Feature 2

Parabolic 3D embedding



- Create 3D object to classify new cases
- Map data into a higher-dimensional space

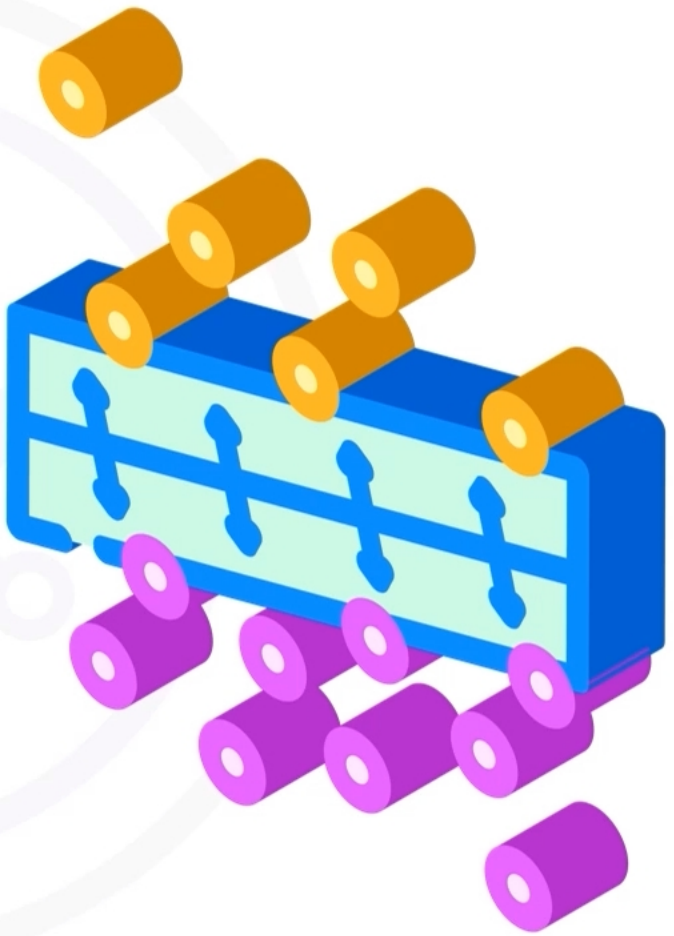
$$x_3 = {x_1}^2 + {x_2}^2$$

Kernelling

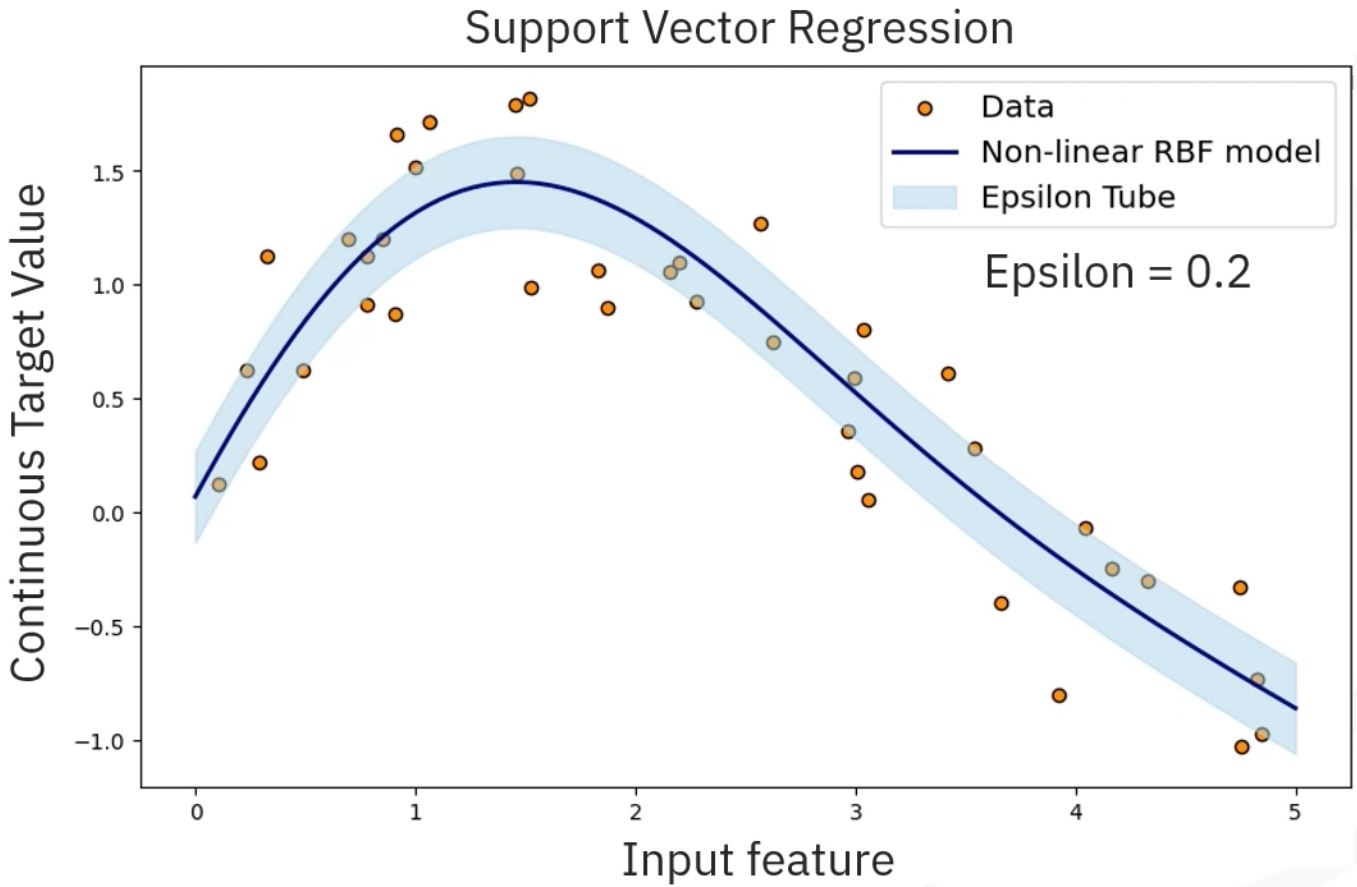
$$(x_1, x_2) \rightarrow (x_1, x_2, x_3)$$

Scikit-learn kernel functions for SVM

- Linear
- Polynomial
- RBF
- Sigmoid



Extension to regression



Orange data points represent:

- Noisy, nonlinear, continuous target variable

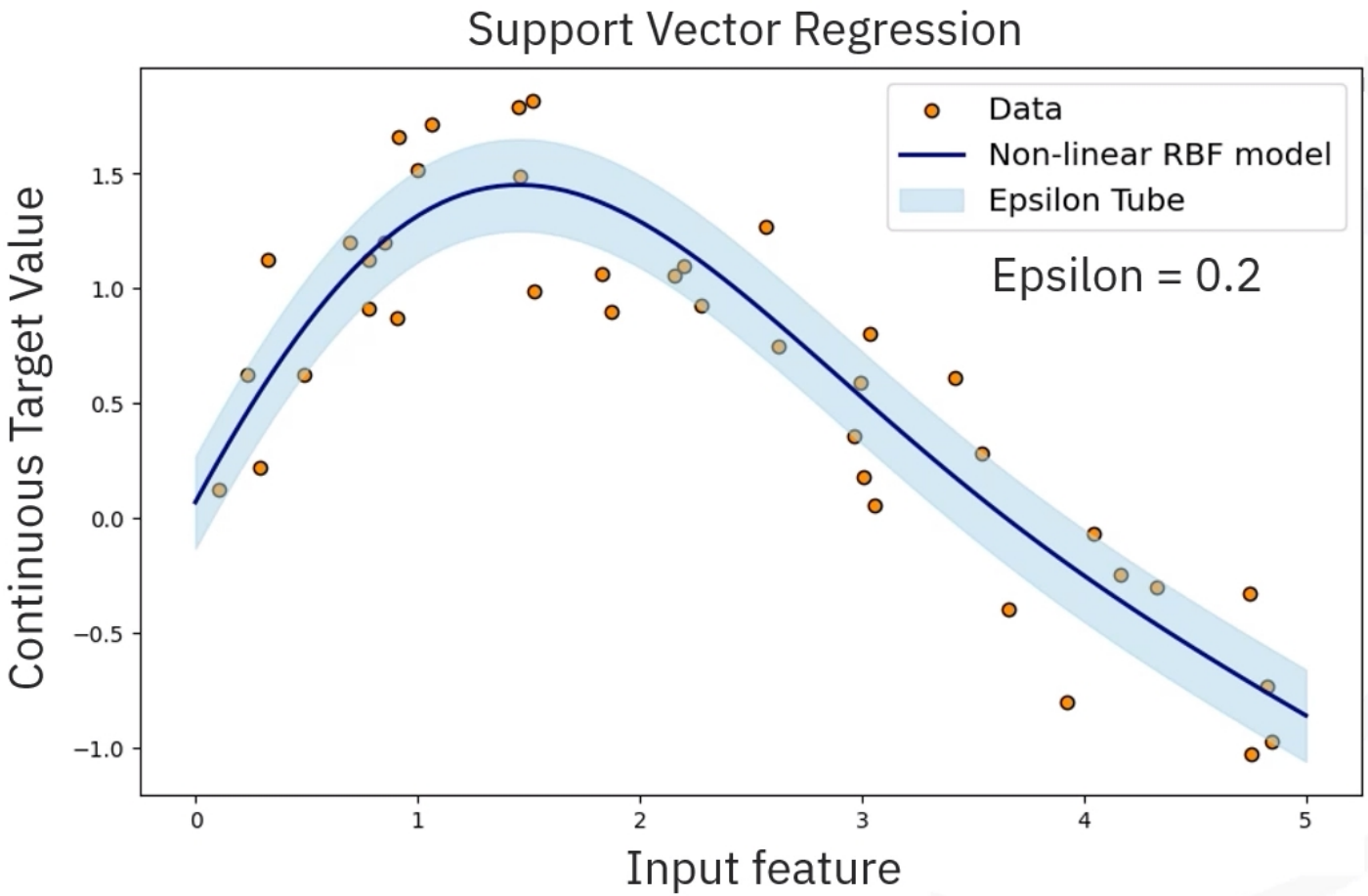
Blue curve displays:

- SVR model prediction

Shaded light blue region represents:

- Epsilon tube around prediction

Extension to regression



Epsilon is SVR parameter to define margin

Points outside the margin are noise

Points inside the margin are signal

SVM pros and cons

Advantages:

- Effective in high-dimensional spaces
- Robust to overfitting
- Excels on linear separable data
- Works with weakly separable data

Limitations:

- Slow for training on large data sets
- Sensitive to noise and overlapping classes
- Sensitive to kernel and regularization parameters

- SVM application

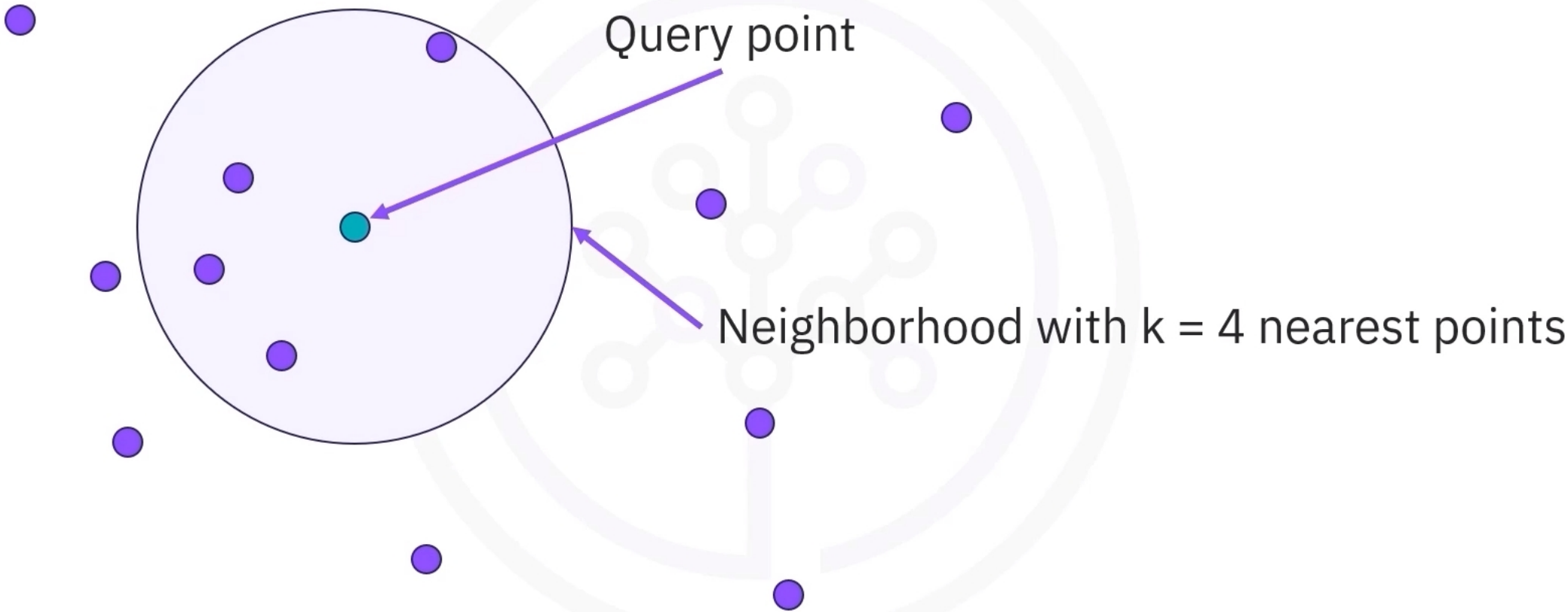
  - Image classification and handwritten digital recognition
  - Parsing, spam detection sentiment analysis
  - Speech recognition, anomaly detection, and noise filtering

---

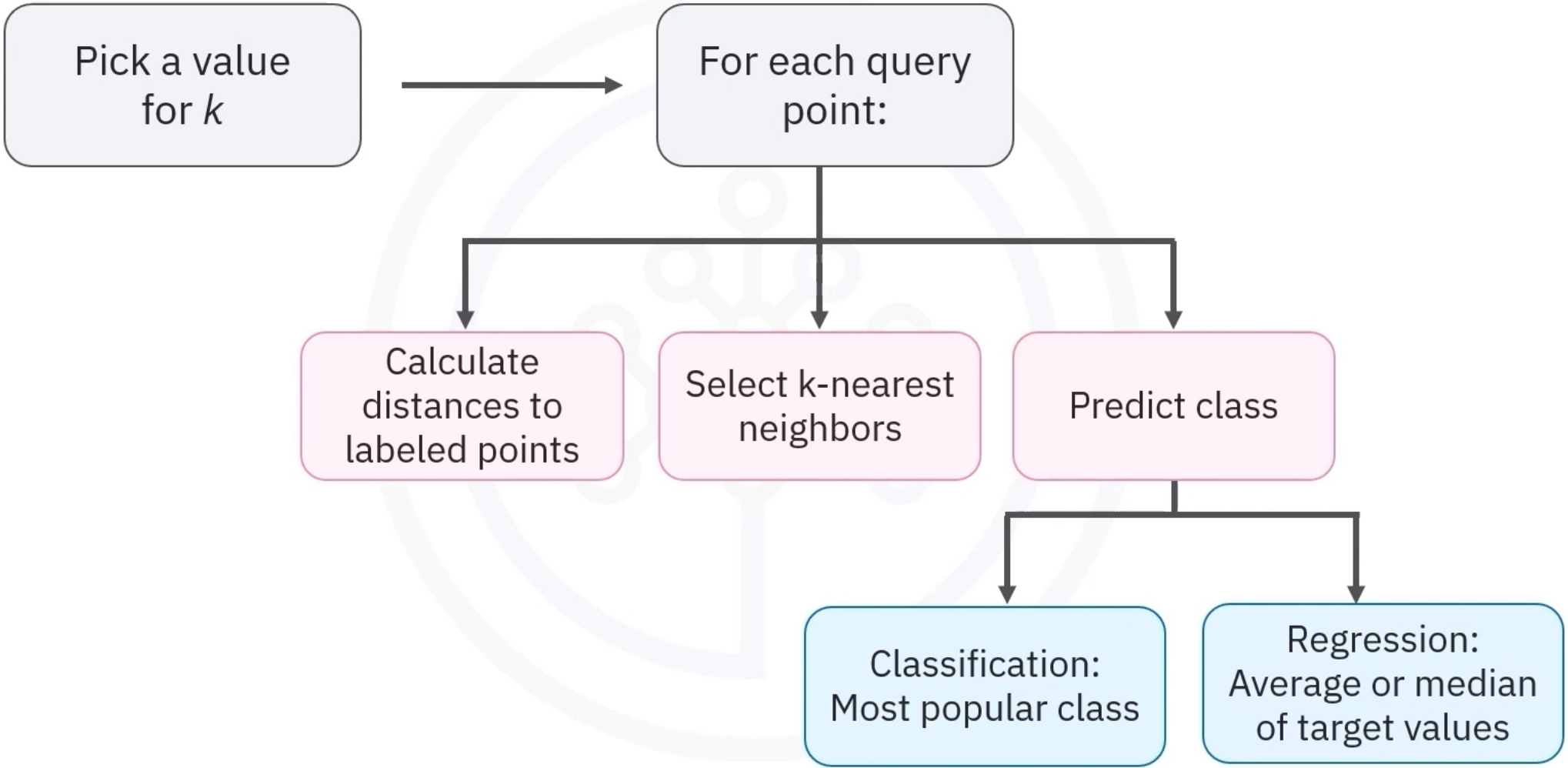
- Supervised learning with K-NN

  - Supervised ML algorithm
  - Use labeled points to learn how to label other points
  - Used for classification and regression
  - Neighbors: Data points near each other with similar features.

What is k-NN?



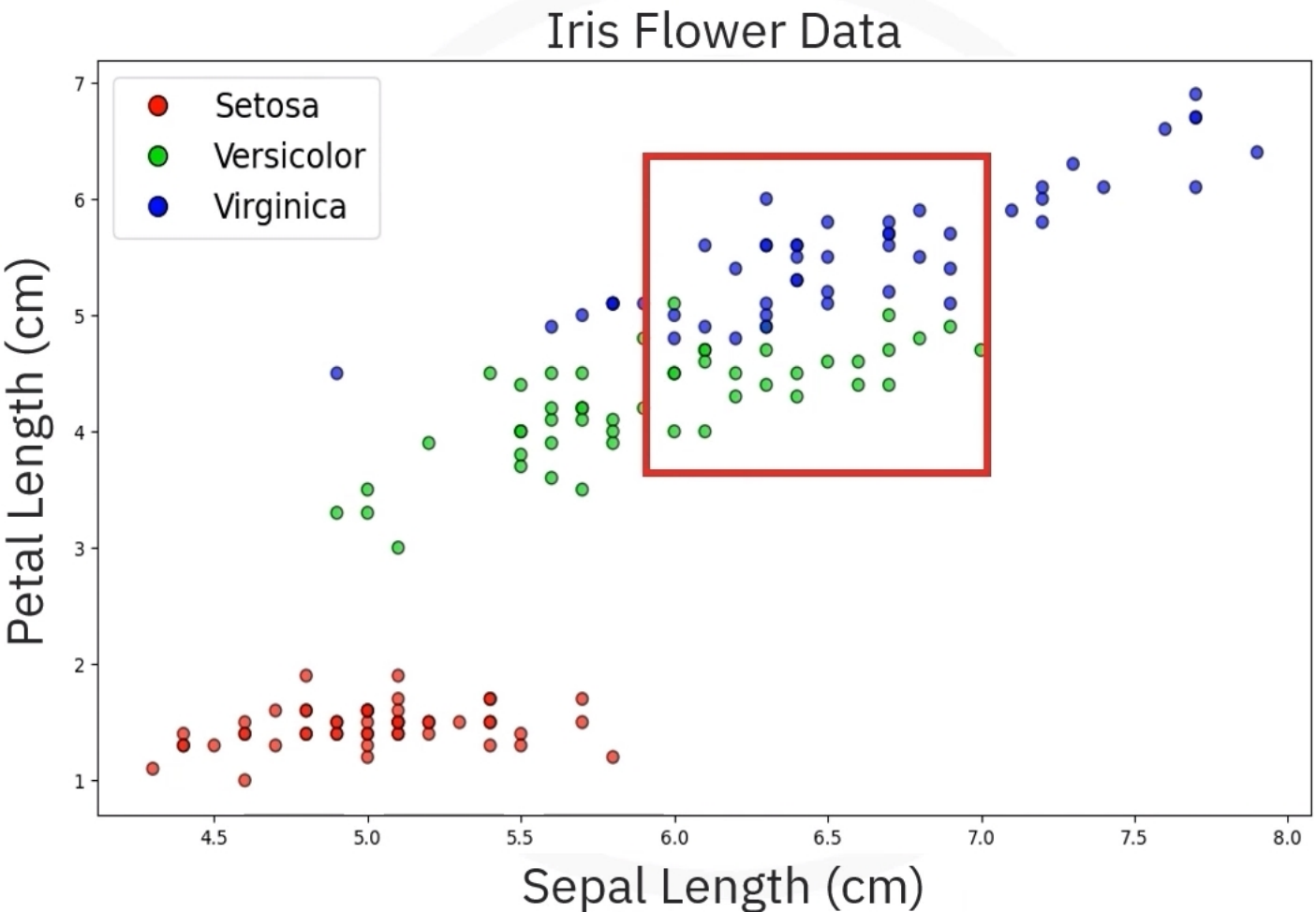
k-NN for classification or regression

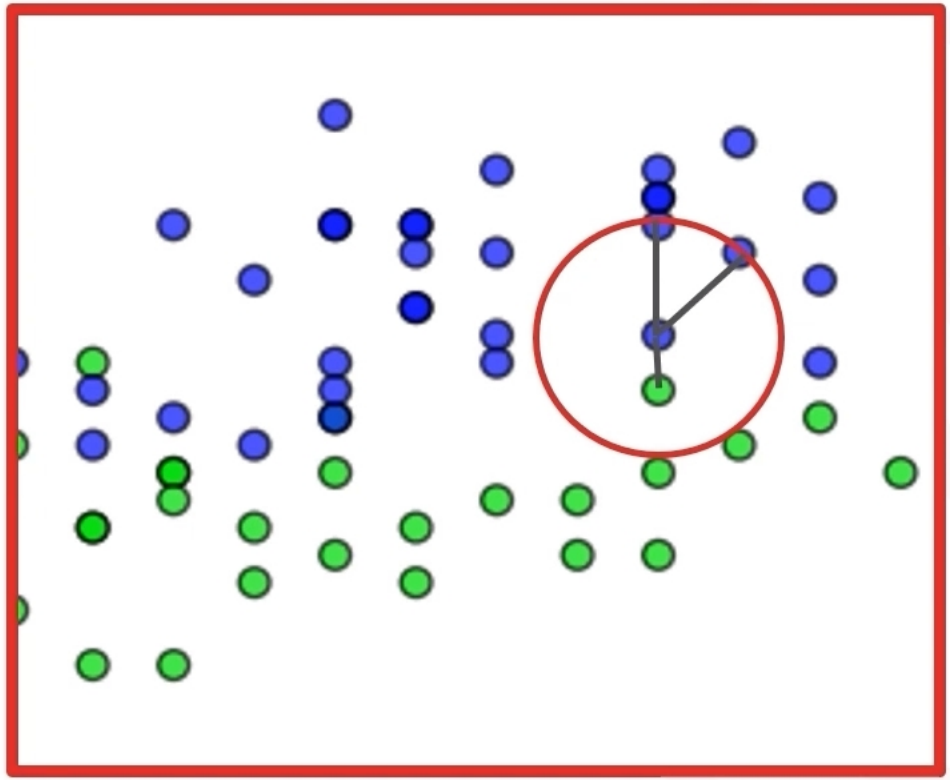


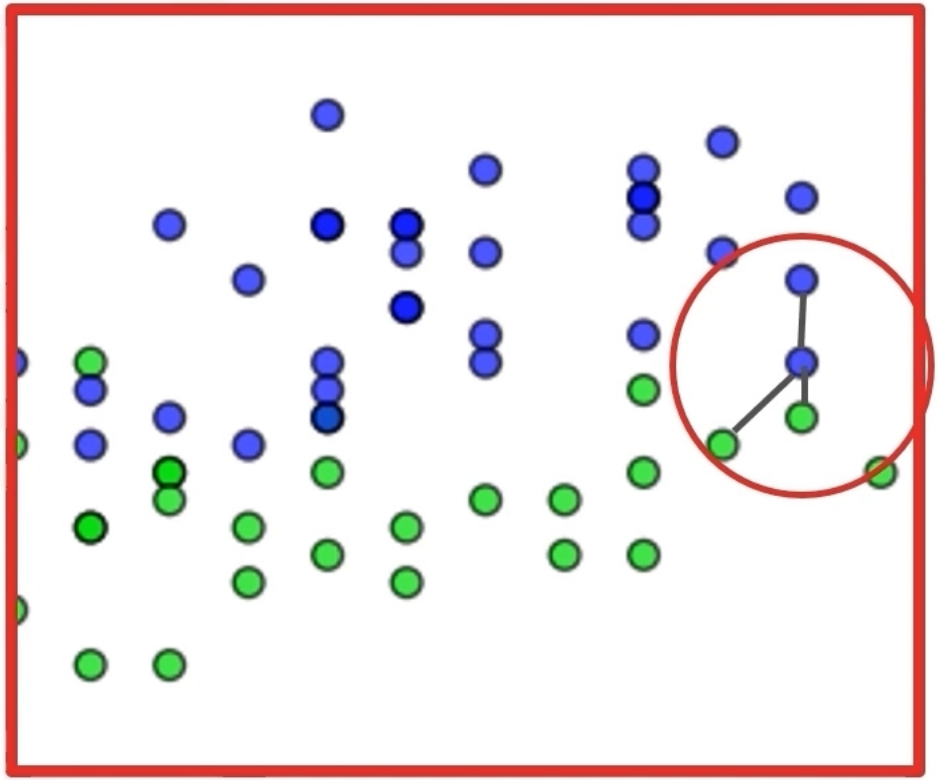
k-NN for classification

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	iris_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa
...	...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2	virginica
146	6.3	2.5	5.0	1.9	2	virginica
147	6.5	3.0	5.2	2.0	2	virginica
148	6.2	3.4	5.4	2.3	2	virginica
149	5.9	3.0	5.1	1.8	2	virginica

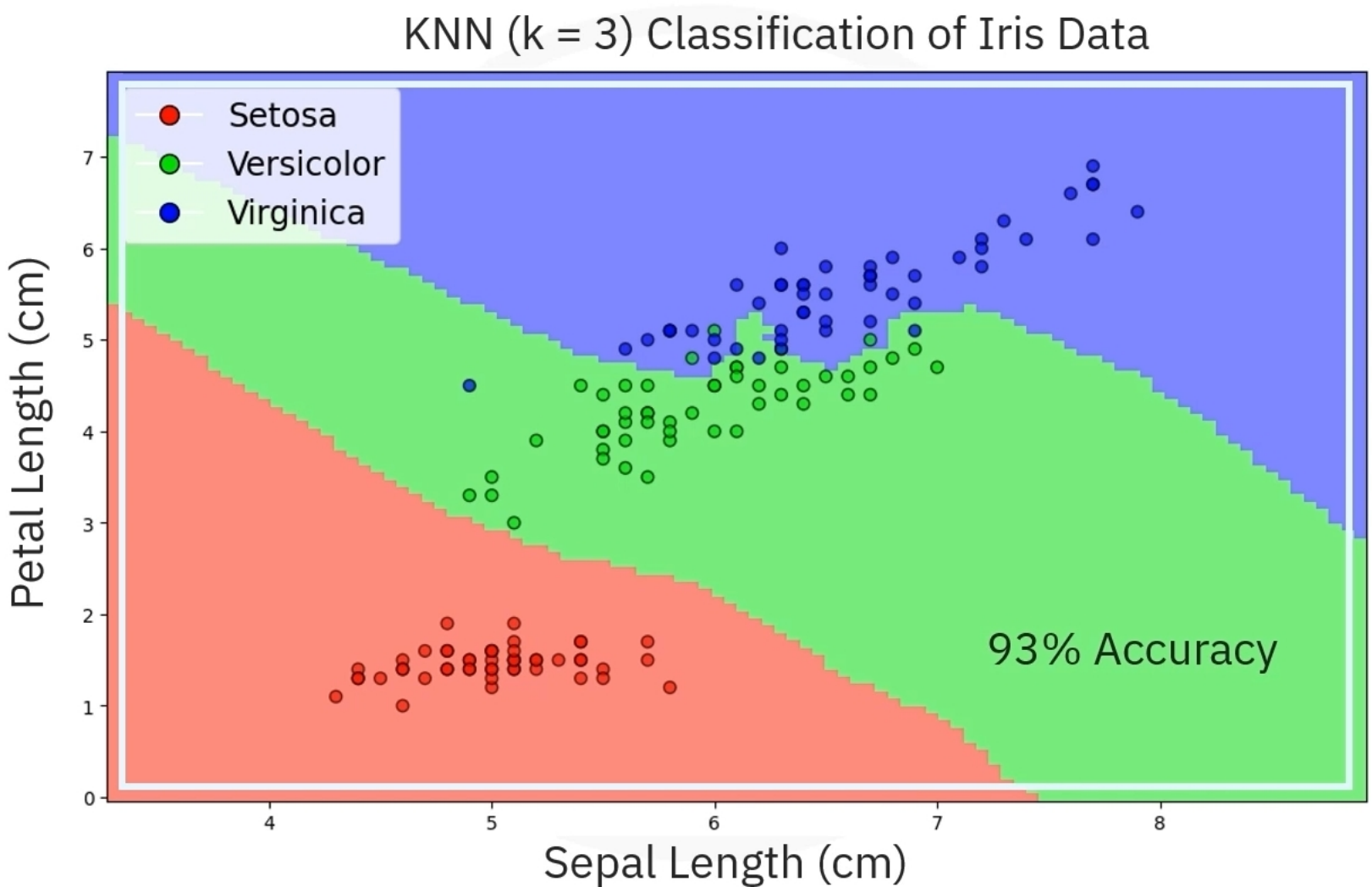
Determining classes with k = 3



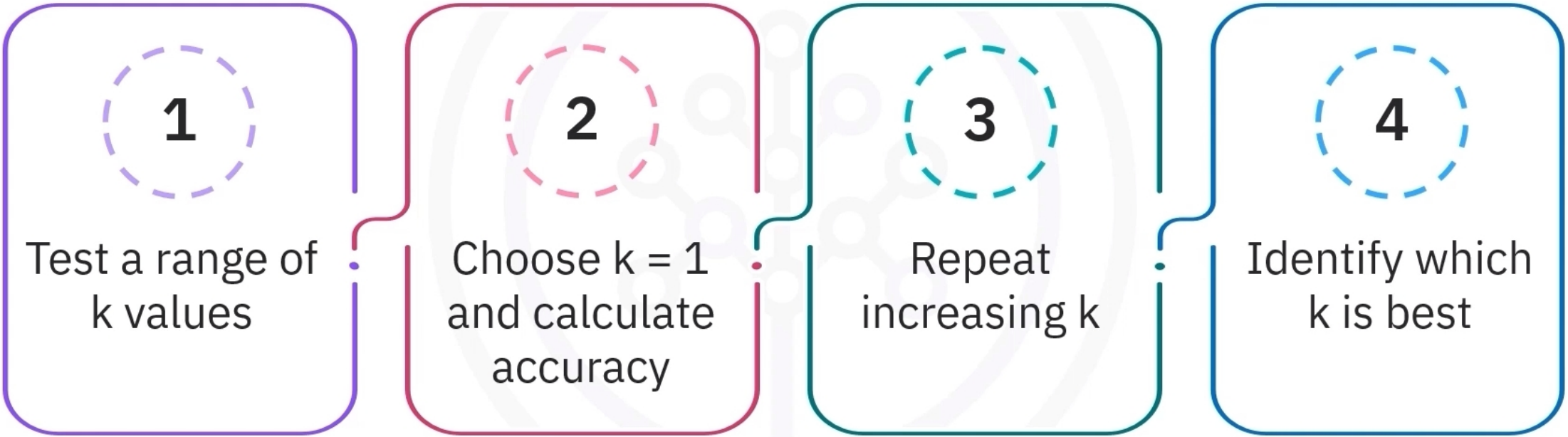


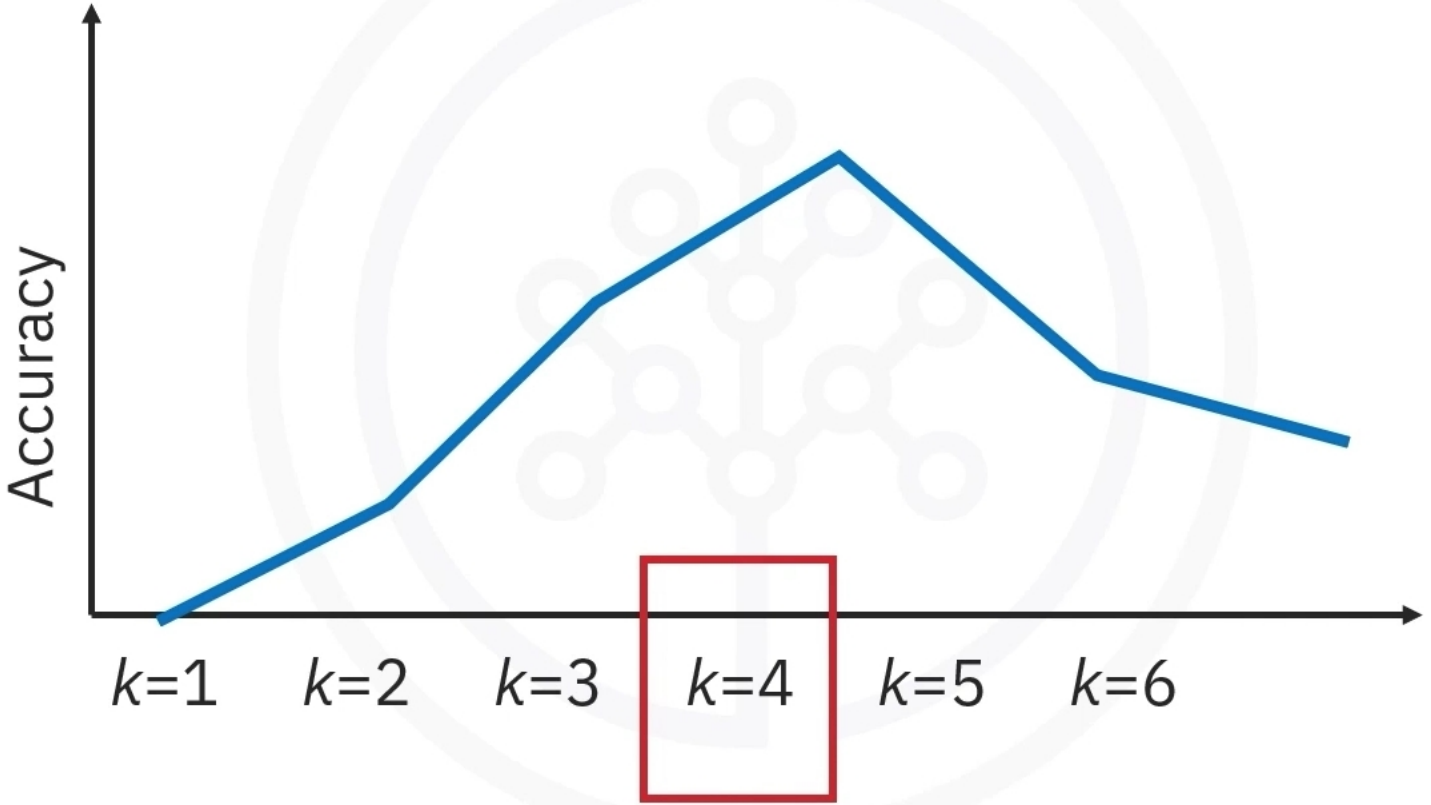


k-NN decision boundary



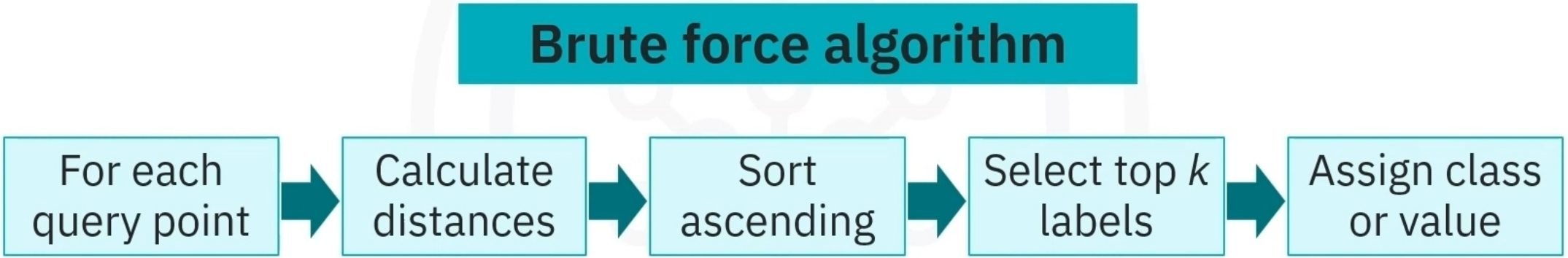
Finding the optimal k




Accuracy
k=1
k=2
k=3
k=4
k=5
k=6

K-NN is a lazy learner

- Memorizes training data
- Makes predictions based on distance to training data points



Effect of k in k-NN

K too small:

- Values fluctuate
- Overfitting

K too large:

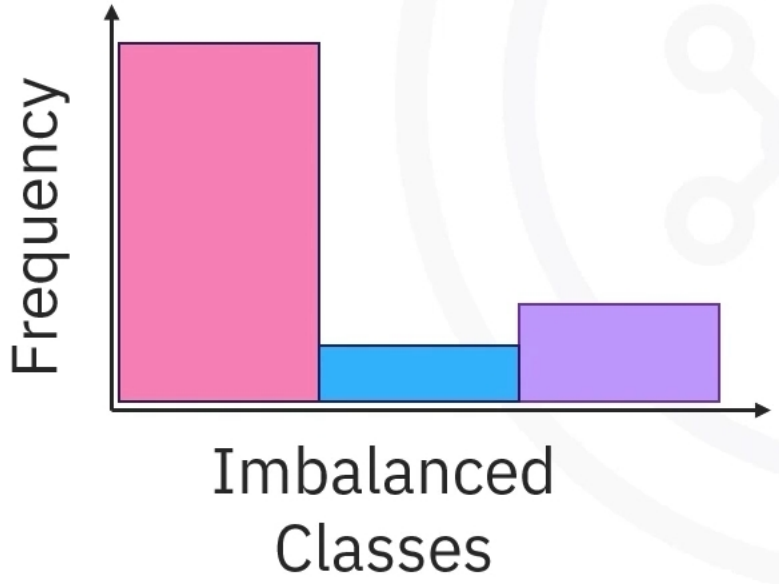
- Finer details lost
- Underfitting

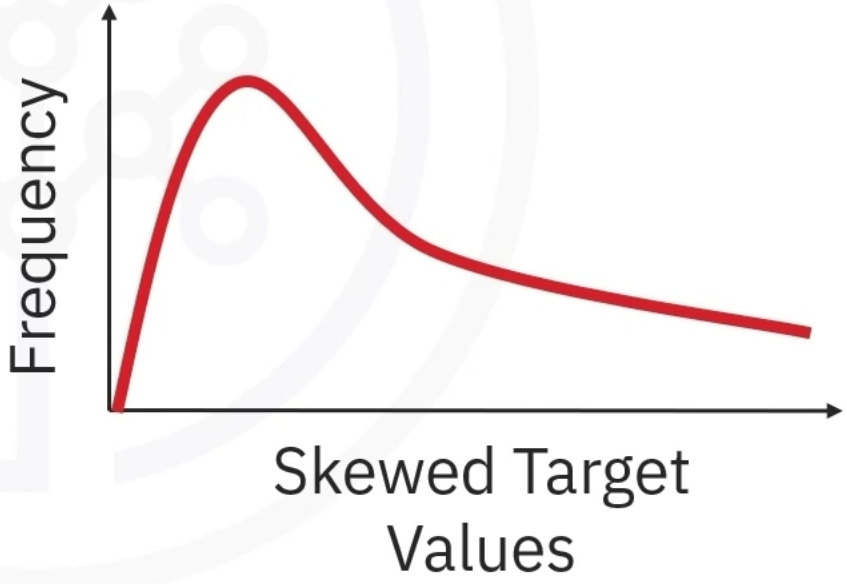
Ideal k:

- In-between

Skewed and imbalanced distributions

- Frequent classes more common in k-neighborhoods
- Dominate predictions, favoring higher frequencies
- Mitigated by penalizing distance from query point

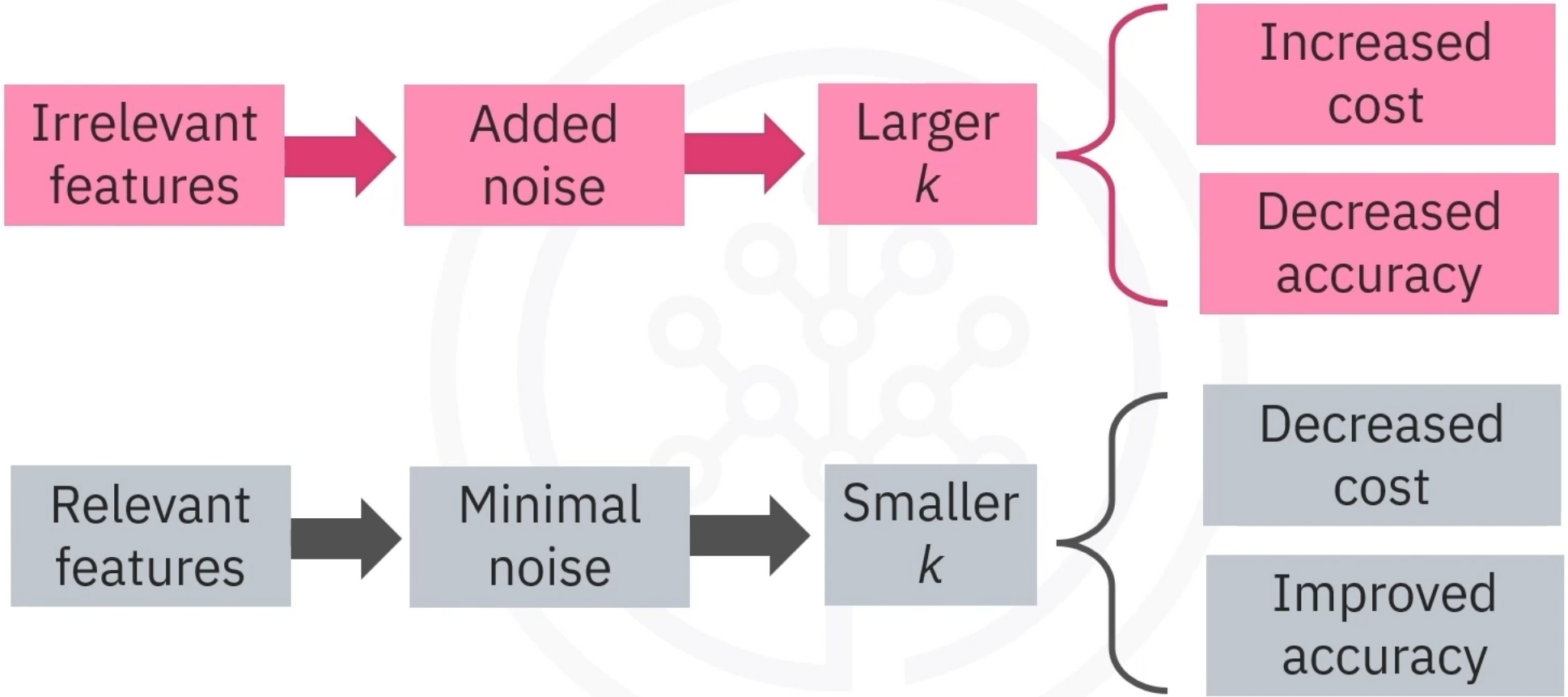




Inconsistent feature

- Large valued features dominate predictions
- Artificially more important features cause bias
- Scale features to mitigate through standardization

Feature relevancy considerations



Feature selection

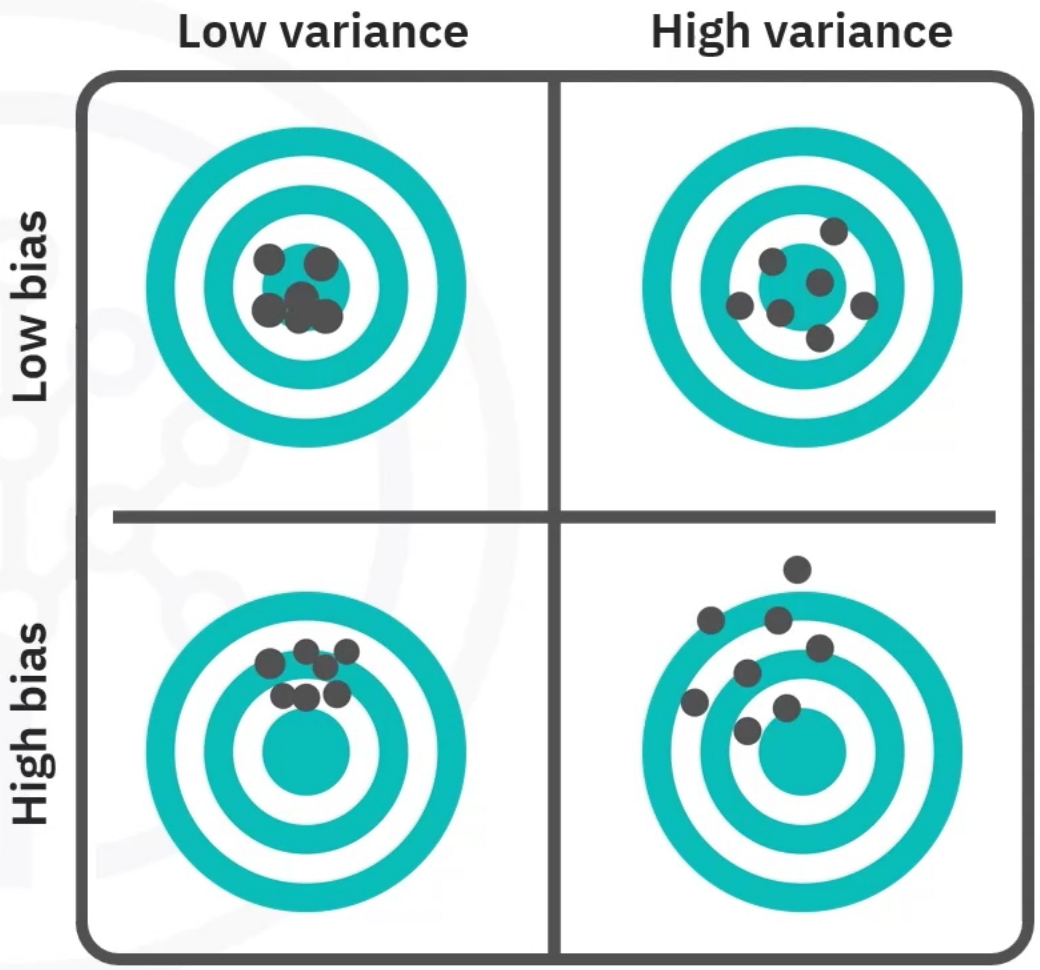
Seek relevant features

Apply domain knowledge

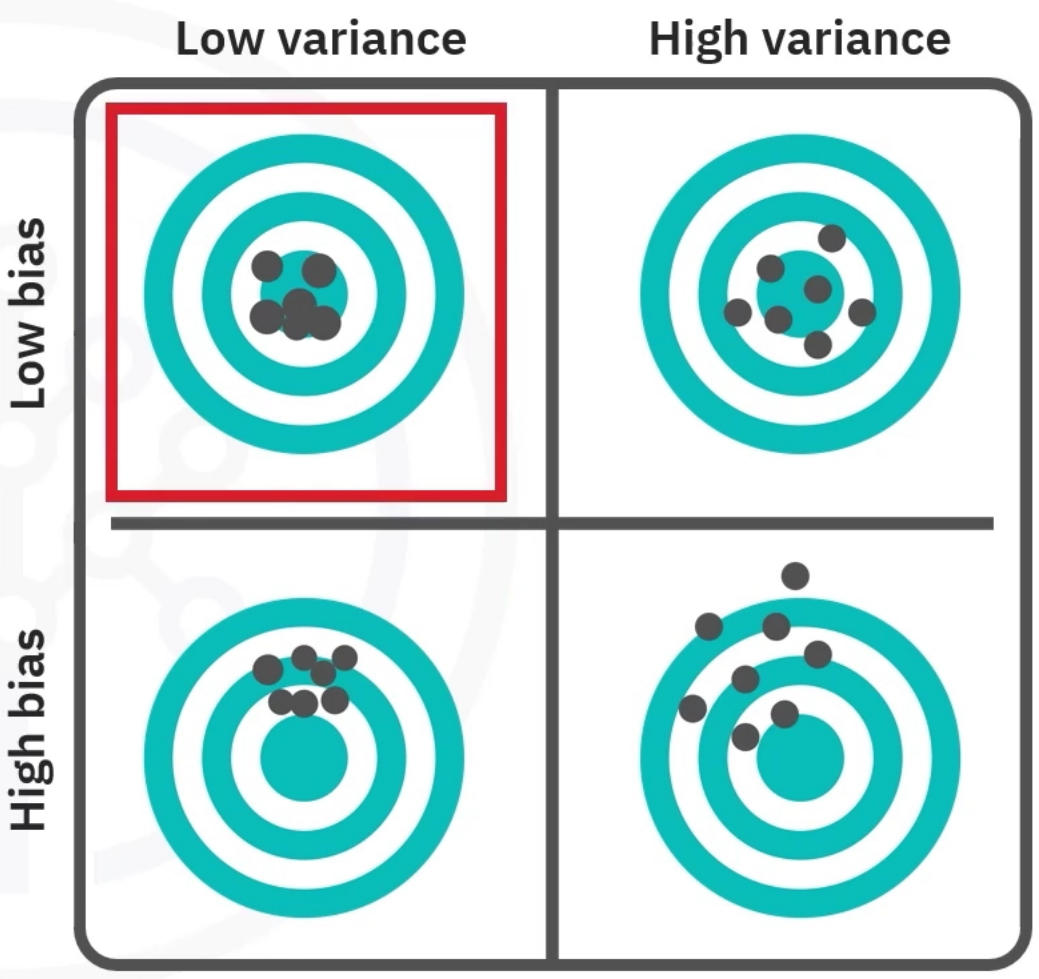
Iteratively test by tuning k

Bias and variance

- Darts near the center show:
  - High accuracy
  - Low bias
- Top boards demonstrate low bias, higher accuracy
- Bottom boards reflect high bias, lower accuracy
- Bias indicates how “on target” predictions are



- Variance measures how spread out darts are
- Higher variance means darts are spread out
- Lower variance means darts are grouped closely
- High scores need low bias and variance



Prediction bias

Measures the accuracy of predictions

$$\text{Bias} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) = \frac{1}{N}\sum_{i=1}^{N}\hat{y}_i - \frac{1}{N}\sum_{i=1}^{N} y_i$$

Reflects differences from target values

Average prediction – average of actuals

Is zero for perfect predictors

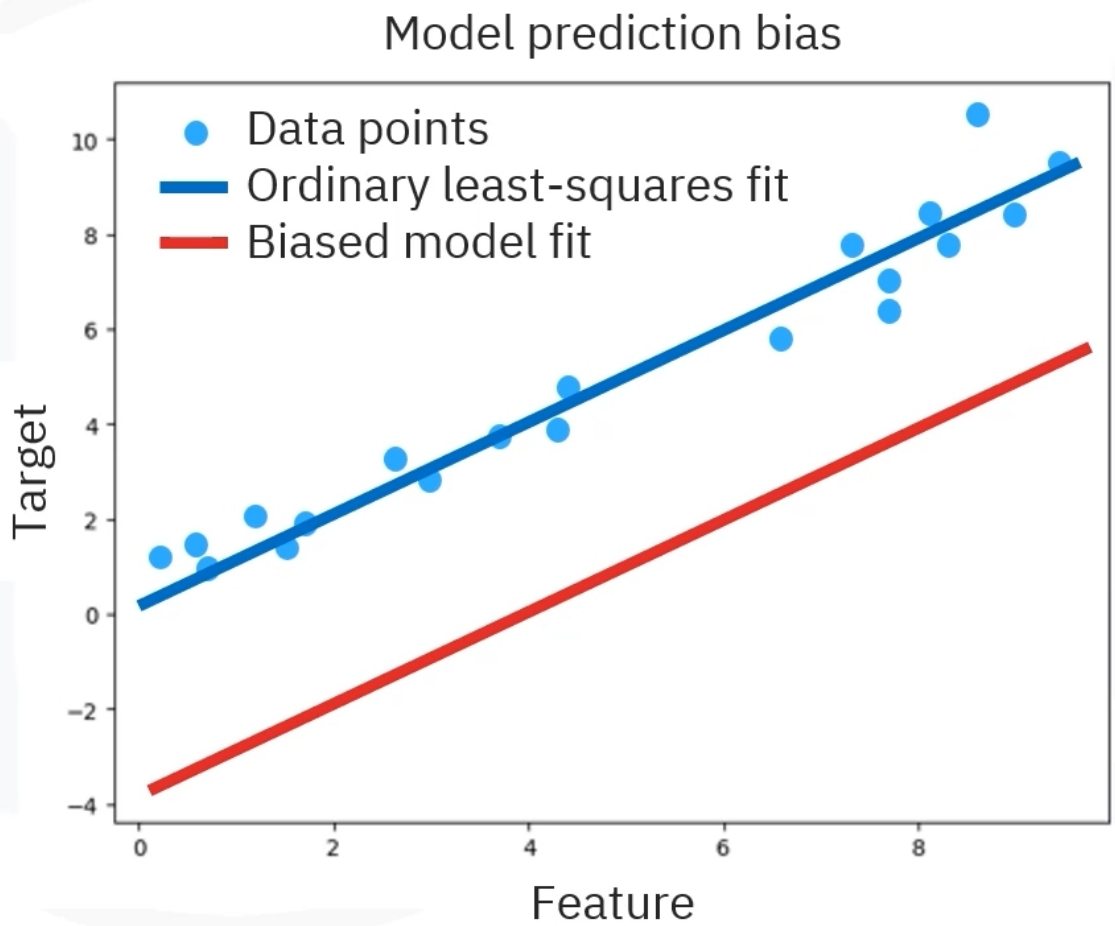
$\hat{y}_i = y_i$ for all $i \implies$ Bias = 0

Blue line:

- Shows least-squares fit
- Bias is 0.22

Red line:

- Shifts model down by 4
- Bias is 4.22



Bias = 0.22

Bias = 4.22

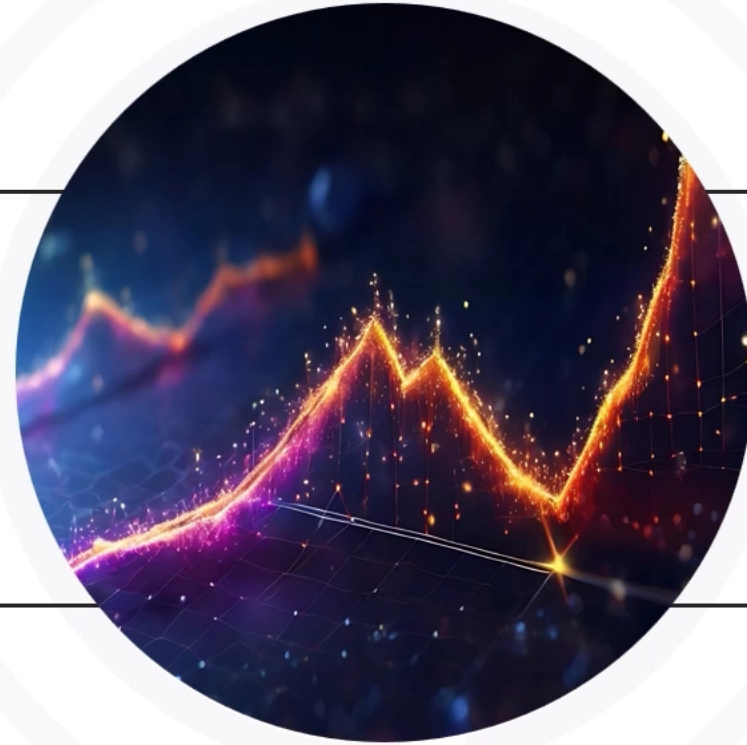
Prediction variance

Prediction variance measures prediction fluctuations

High variance shows sensitivity to training data

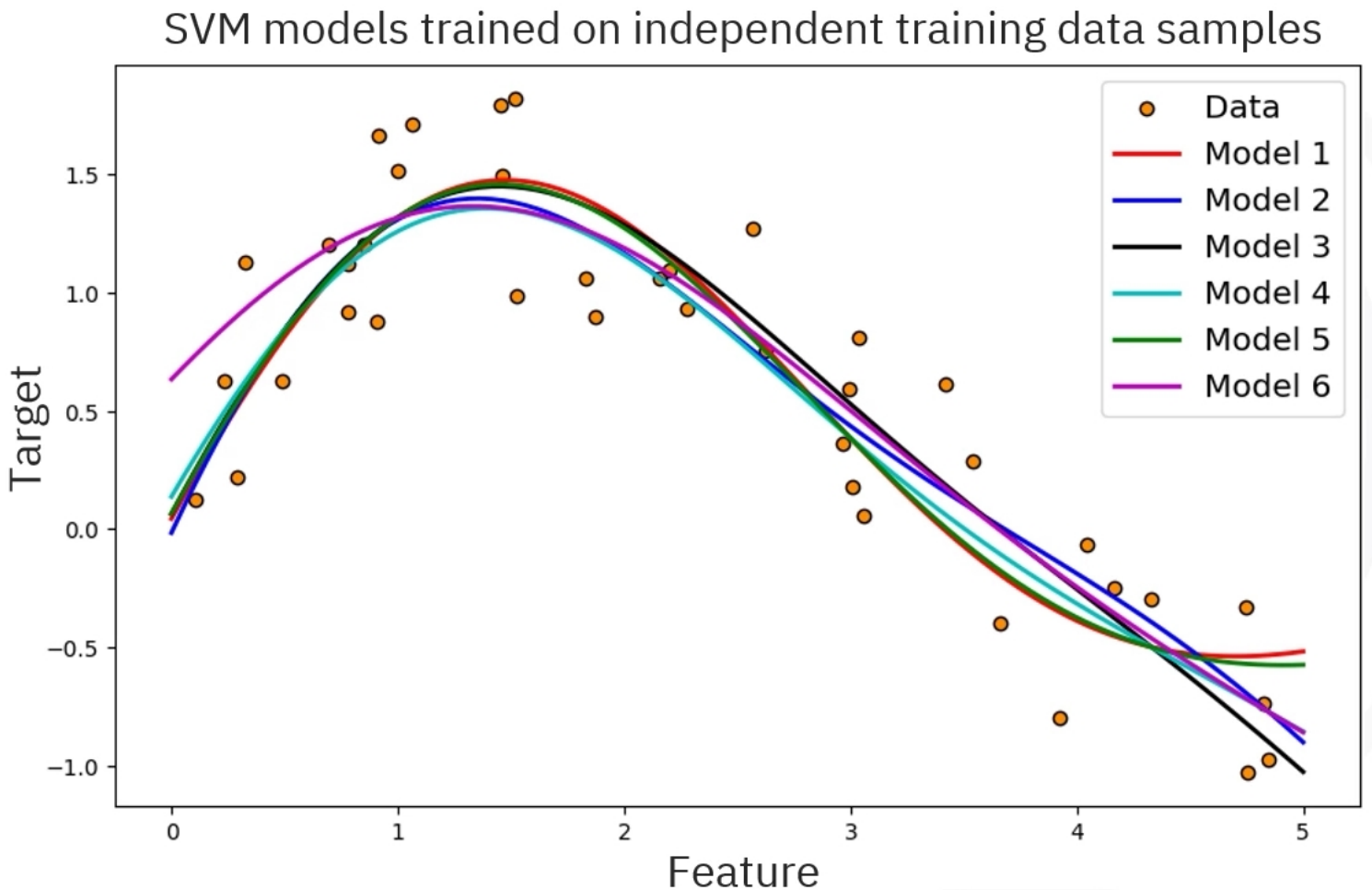
High variance:

- Leads to overfitting training data
- Tracks noise in training data



Low variance:

- Generalizes well to unseen data
- Means less sensitivity to noise

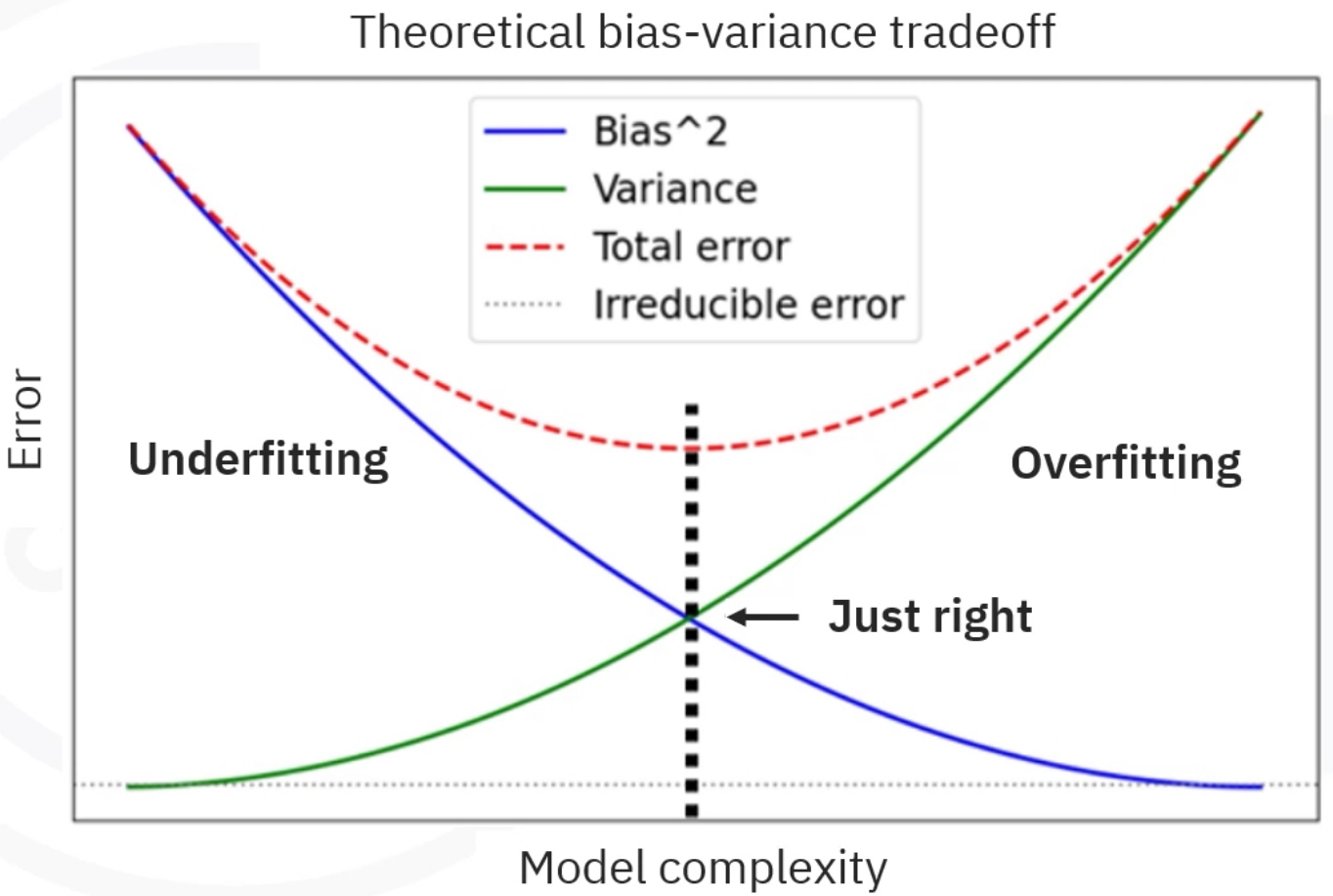


- Chart shows orange points with nonlinear pattern
- Models use randomly sampled training data set

- Curves align if variance is near zero
- Differences appear at data's start and end
- Variation indicates prediction variance exists
- Prediction variance shows model instability

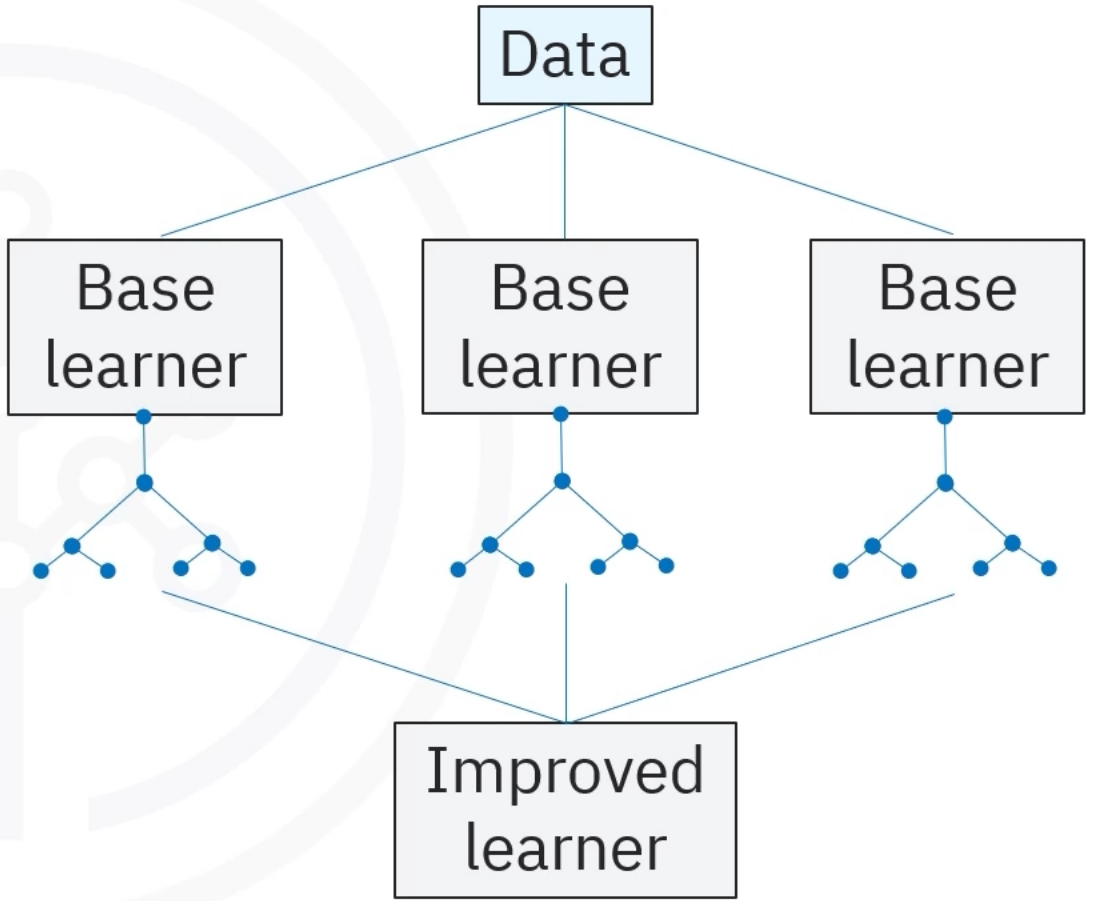
Bias-variance tradeoff

- Plot shows changes in bias and variance

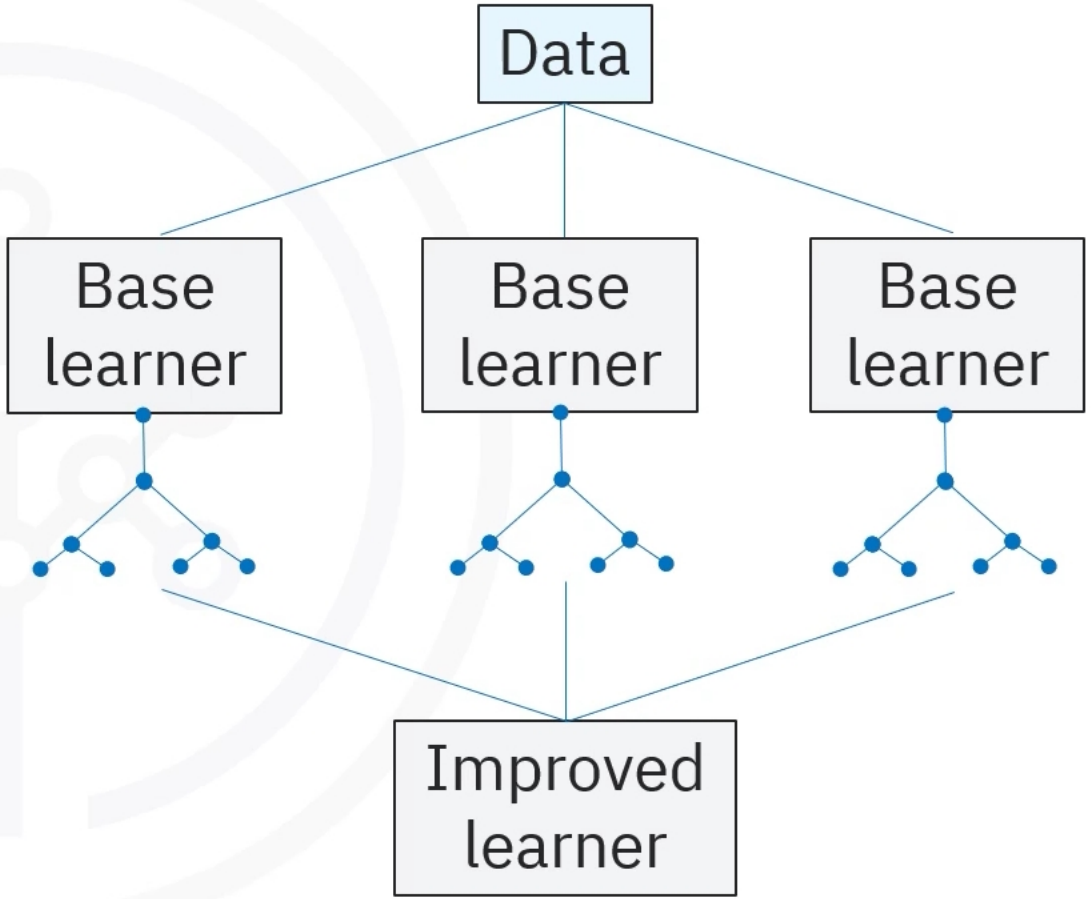


Mitigating bias and variance

- Weak learners perform slightly better than random guessing
- Weak learners have high bias, low variance
- High bias often leads to underfitting

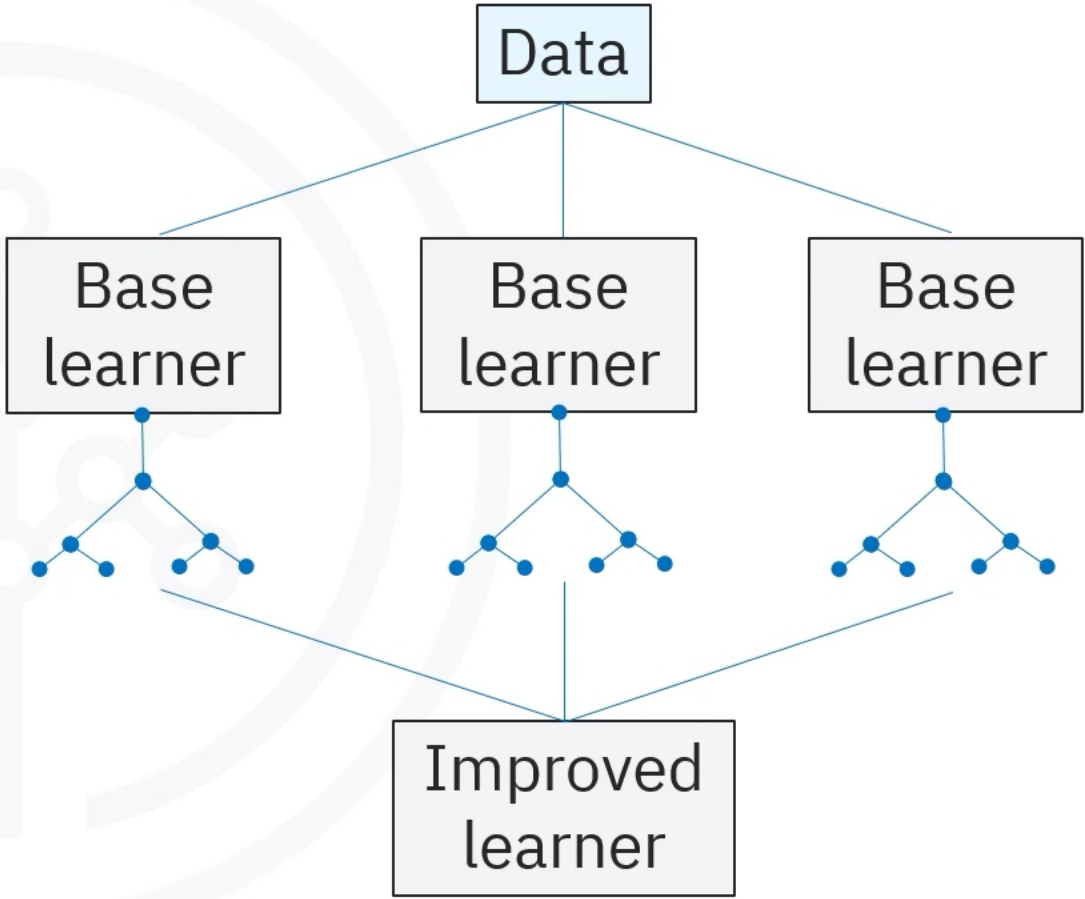


- Strong learners have low bias, high variance
- High variance often causes overfitting

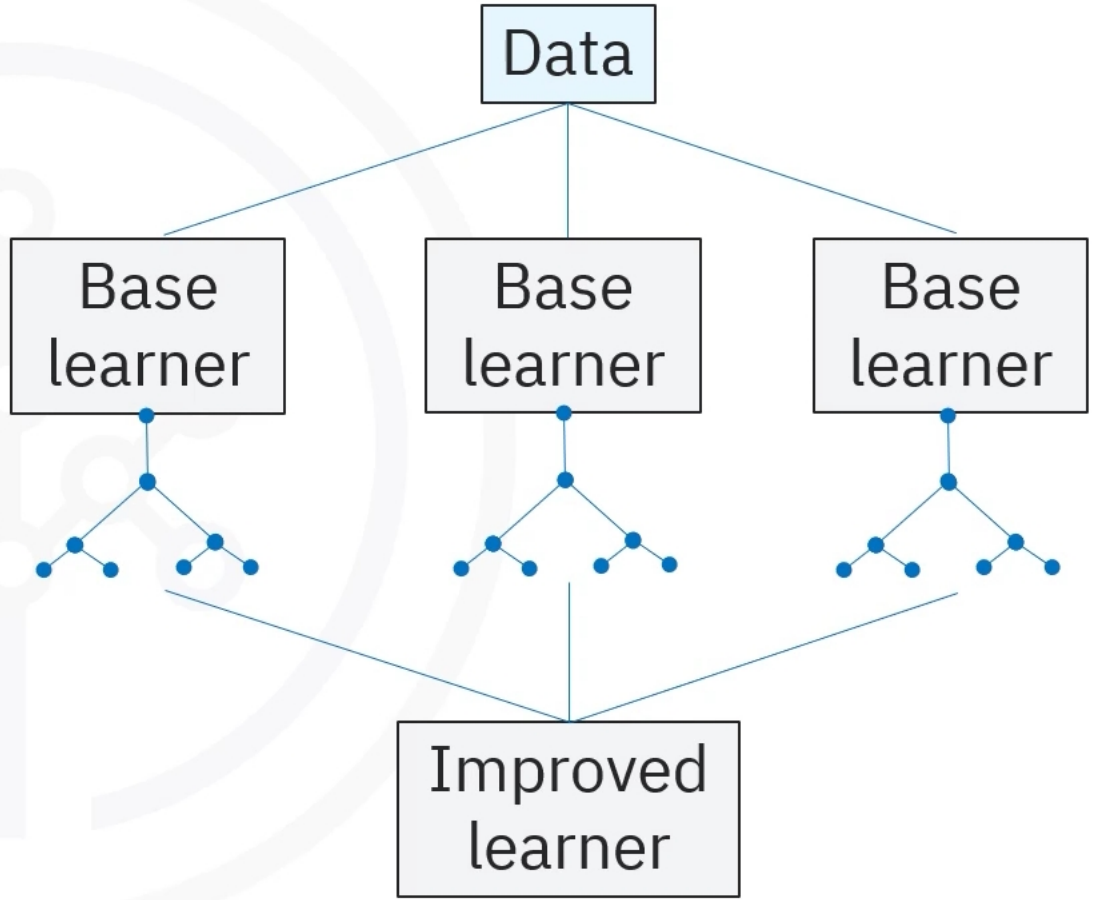


Bagging and boosting:

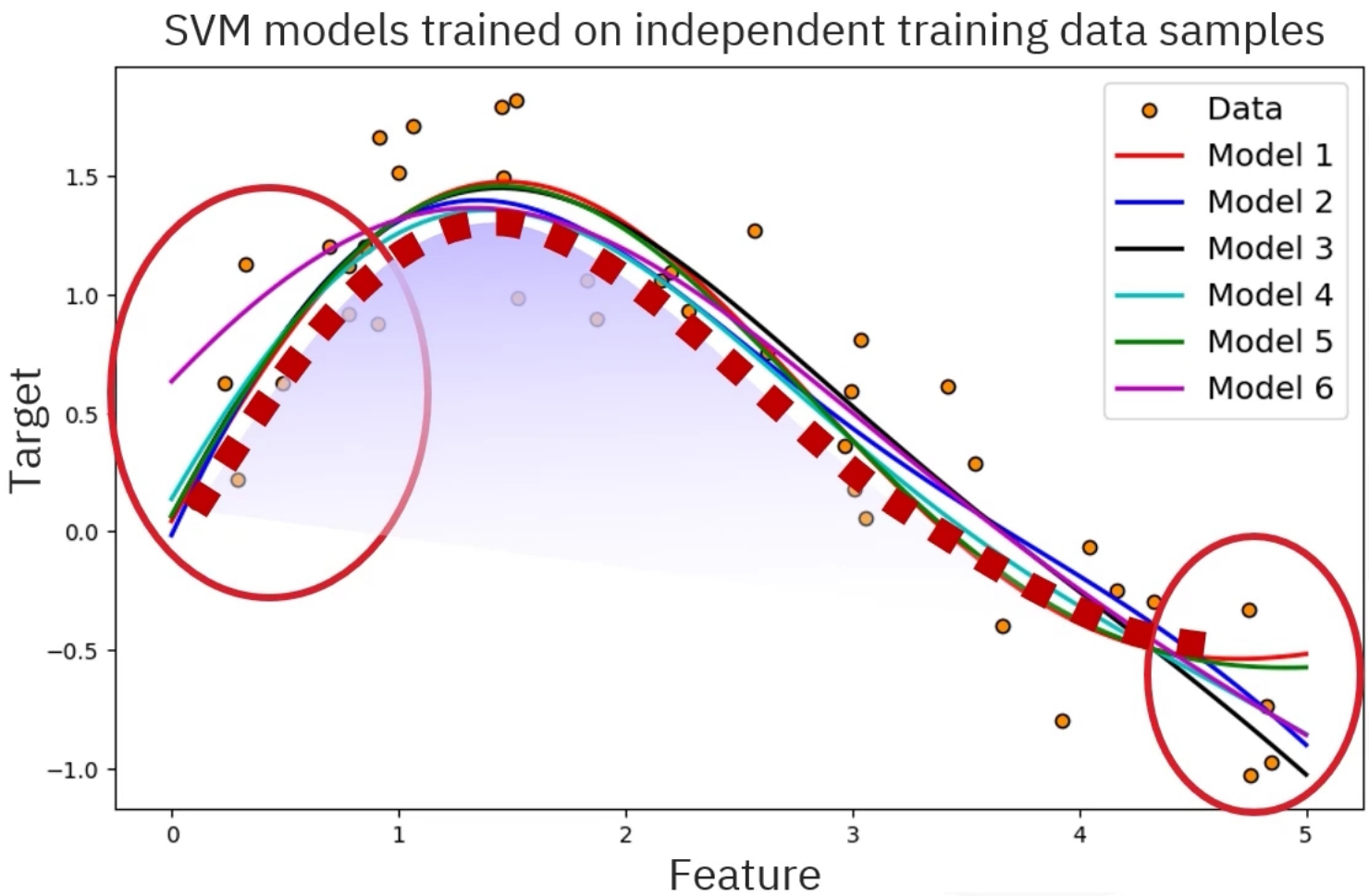
- Popular methods for ensemble learning
- Effective at balancing bias and variance



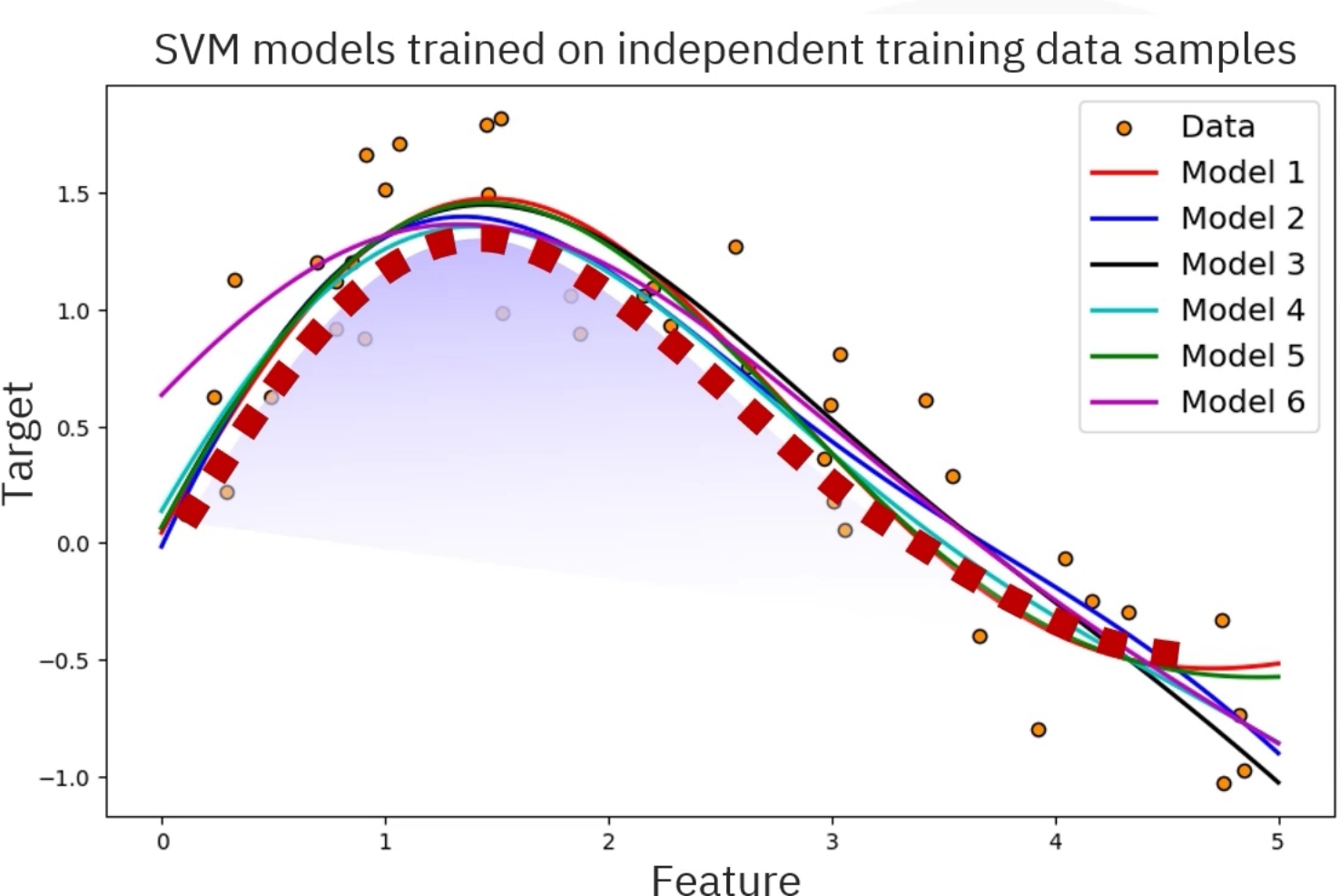
- Decision trees serve as base learners
- Tree depth adjusts bias and variance



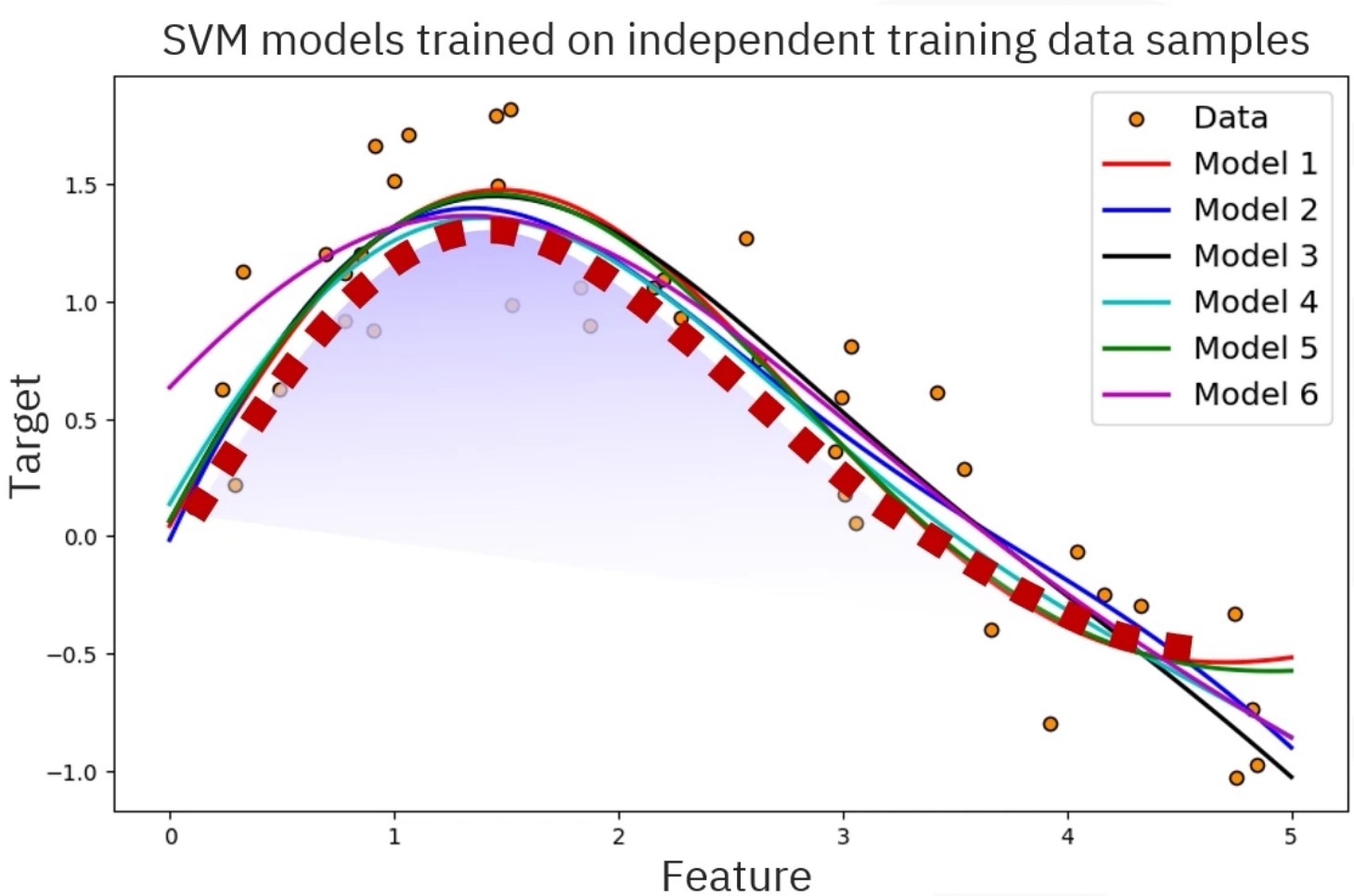
Bootstrap aggregating or bagging



- Model predictions use the same algorithm
- Algorithm trains on bootstrapped data subsets
- Variance appears at curve ends



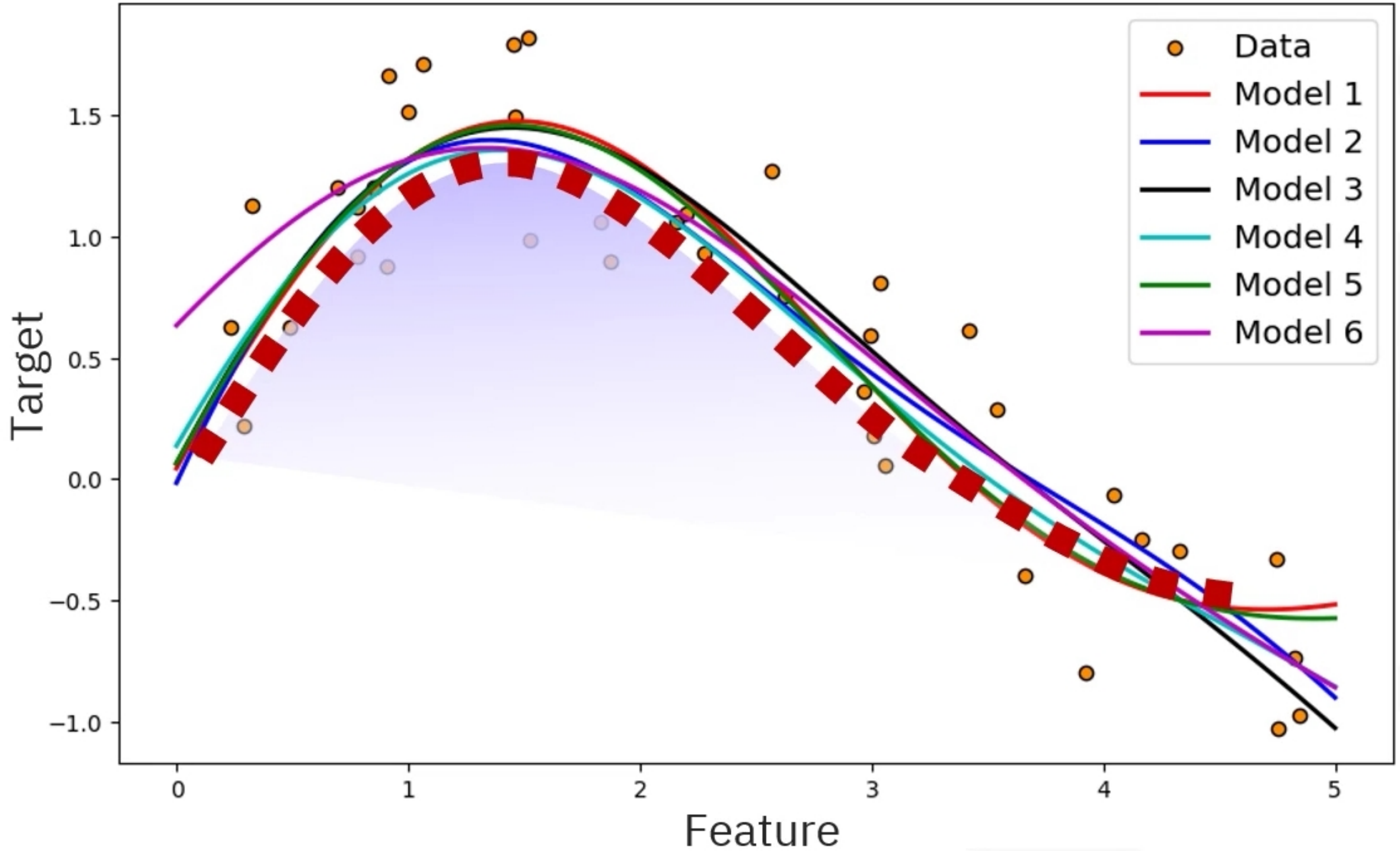
- Perform the process multiple times
- Average predictions from multiple iterations



This technique is known as:

- Bagging
- Bootstrap aggregating

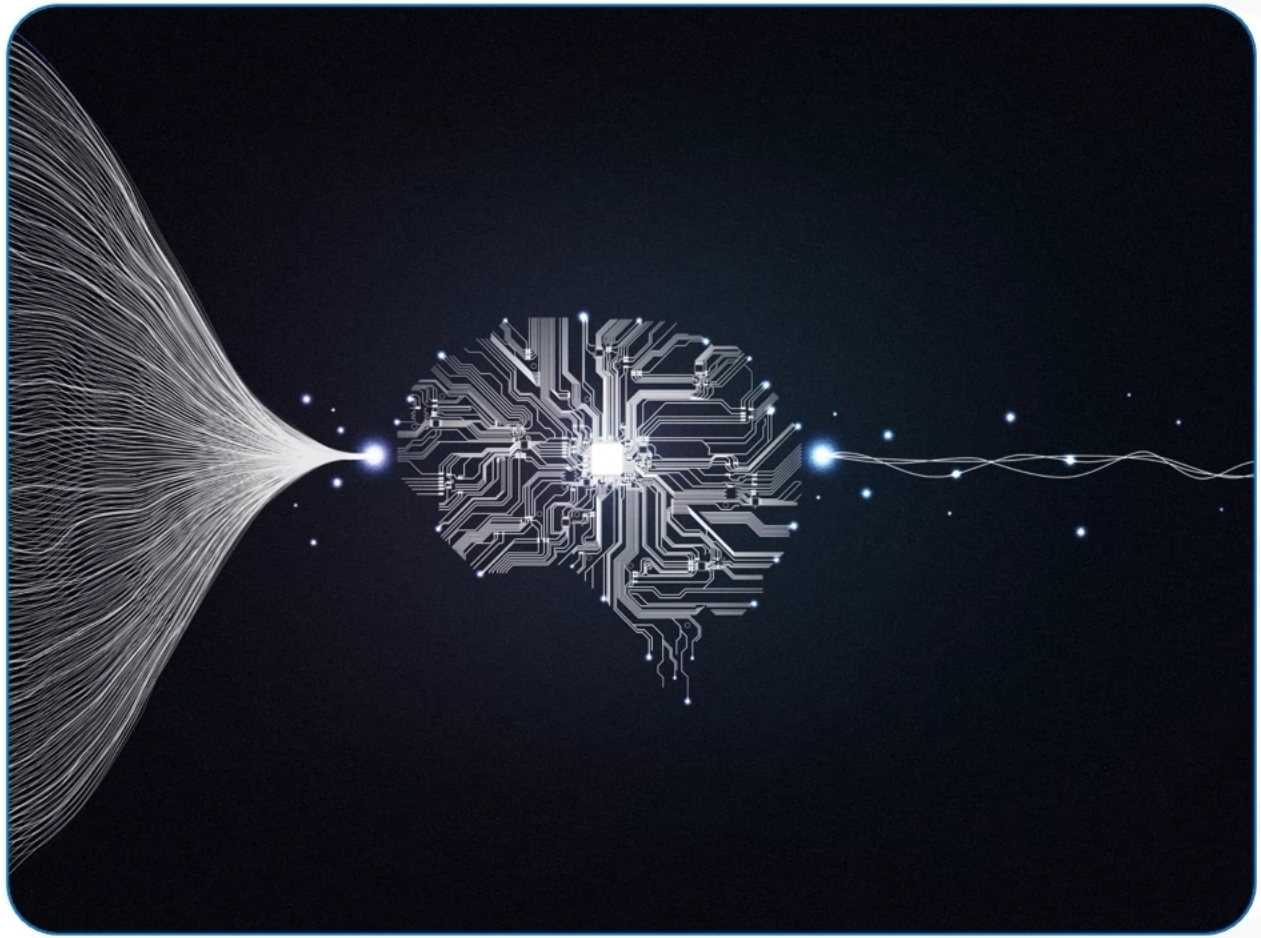
SVM models trained on independent training data samples



Averaging the models:

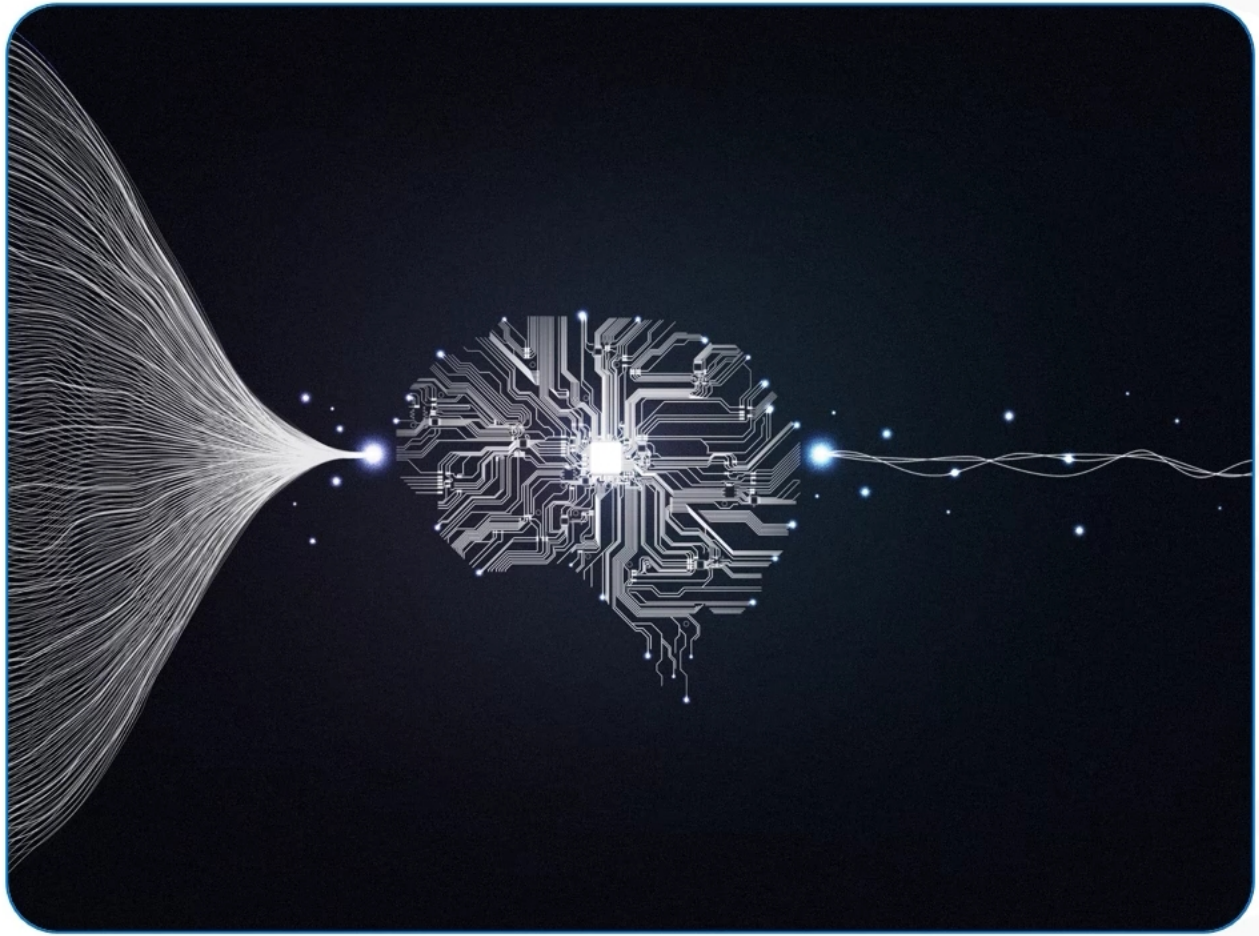
- Reduces prediction variance
- Lowers the risk of overfitting

Random forests



Random forests

- Use bagging for training
- Train decision trees on bootstrapped data sets
- Focus on minimizing prediction bias



- Shallow trees have high prediction variance
- Focus on minimizing prediction bias

Boosting

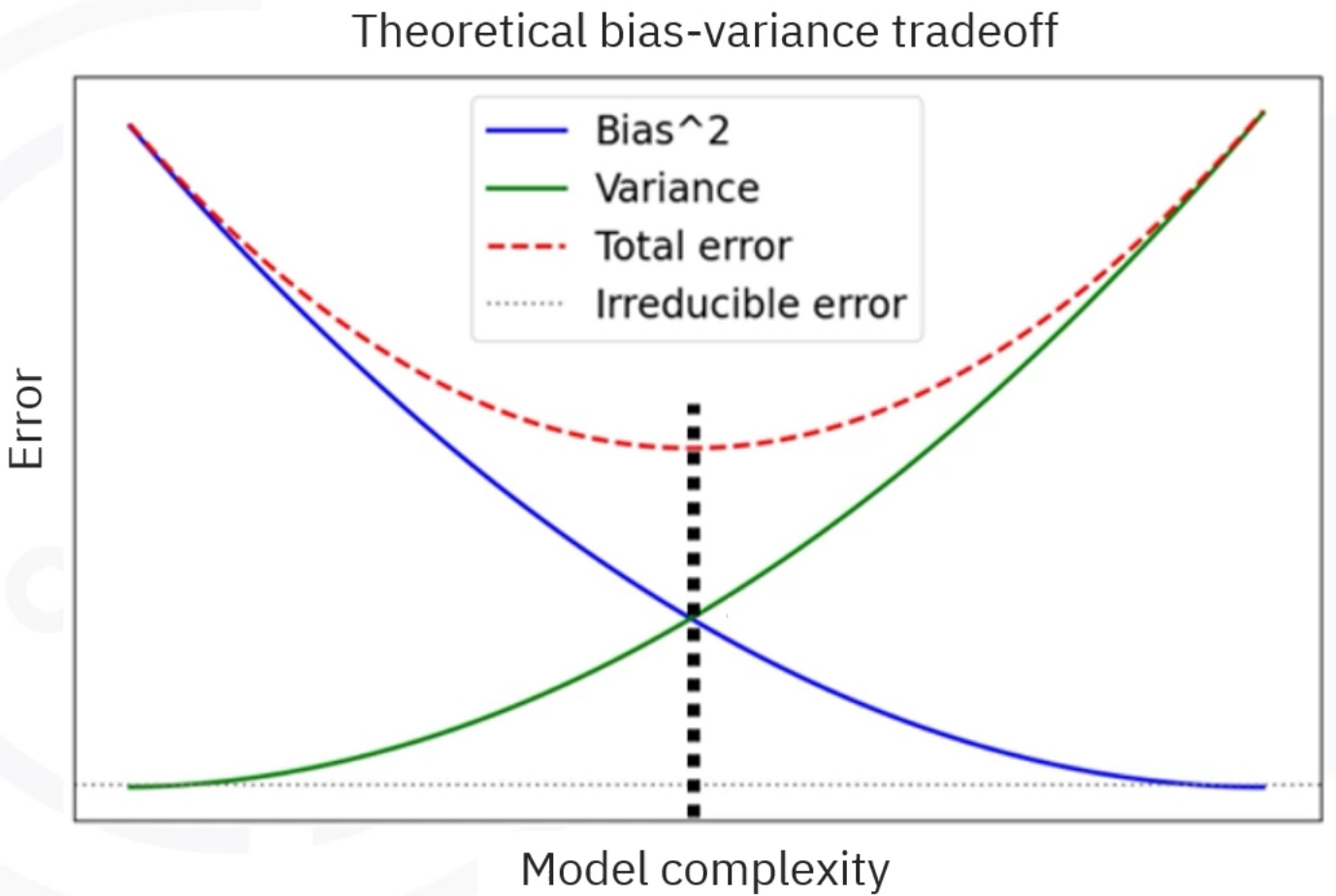
- Boosting builds a series of weak learners
- Each learner corrects the previous learner’s errors
- Boosting systematically reduces prediction error
- The final model is a weighted sum

- Increase weights for misclassified data
- Decrease weights for correctly classified data
- Reweighting focuses on correcting mistakes
- Update model weights based on performance

- Gradient Boosting
- XGBoost
- AdaBoost

Bias-variance tradeoff

- Mitigates the bias-variance tradeoff
- Adjusts model complexity

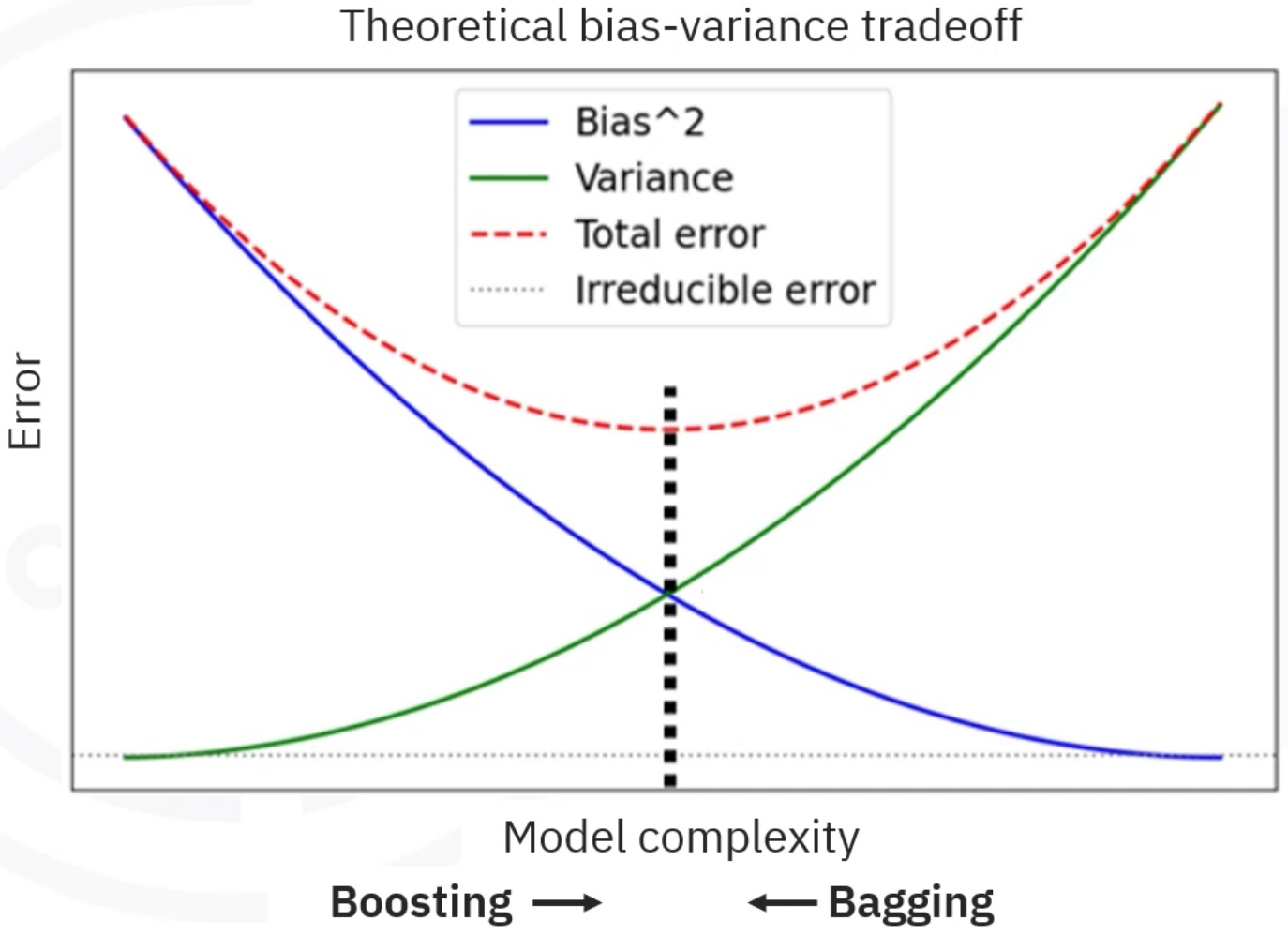


Boosting:

- Increases model complexity
- Decreases bias

Bagging:

- Increases variance



Bagging and boosting

Ensemble	Objective	Base Learners	Training	Outcome
Bagging	Mitigate overfitting	High variance Low bias	Parallel on bootstrapped data	Reduced variance
Boosting	Mitigate underfitting	Low variance, High bias	Builds on previous result	Reduced bias

Recap

- Analyze bias and variance for accuracy and precision
- Explain prediction bias to measure prediction accuracy
- Analyze prediction variance to measure prediction fluctuations
- Explain the bias-variance tradeoff
- Explain mitigating bias and variance
- Analyze bagging or bootstrap aggregating to observe variance
- Explain random forests to train multiple decision trees
- Analyze bagging and boosting outcomes