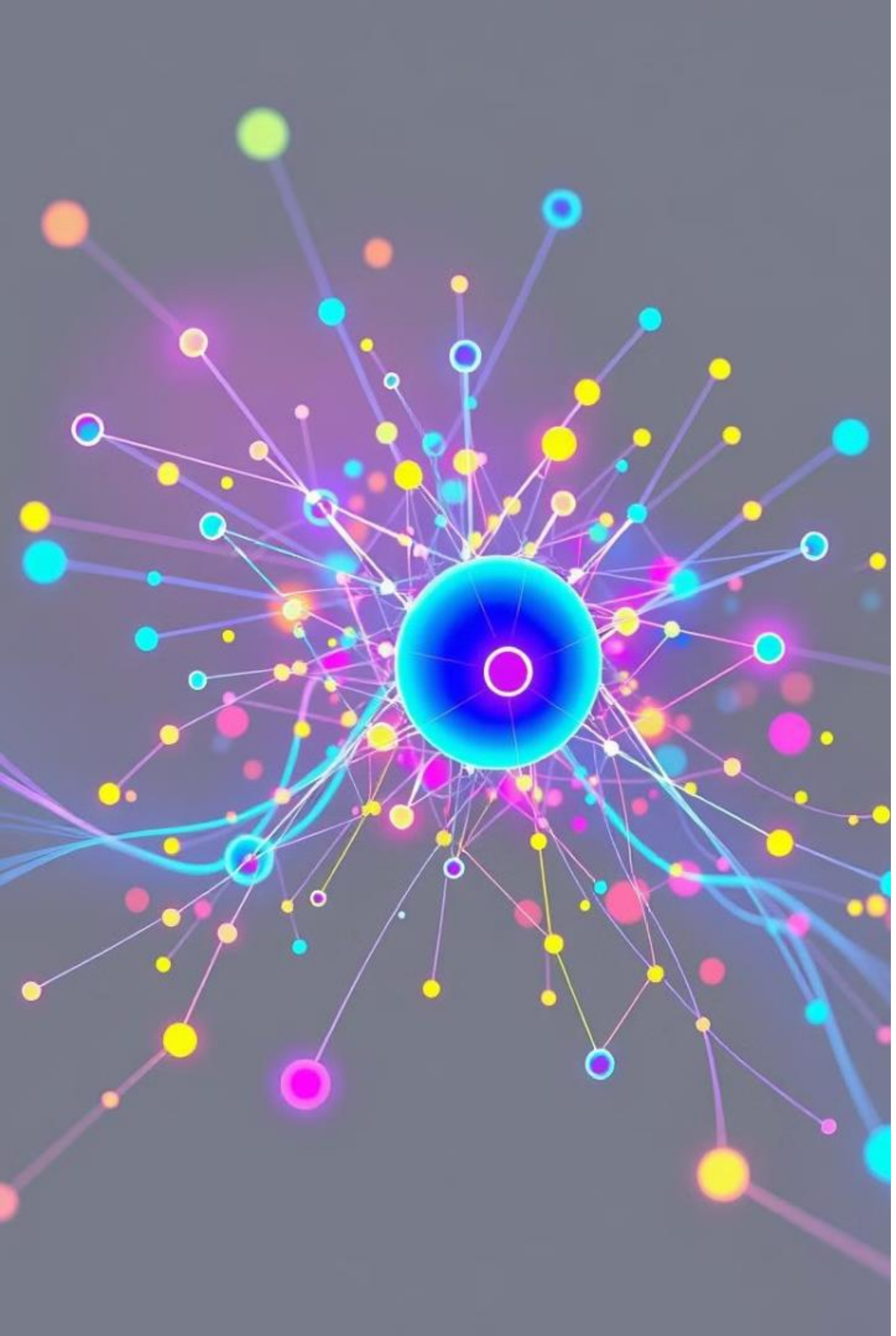
Carnegie Mellon

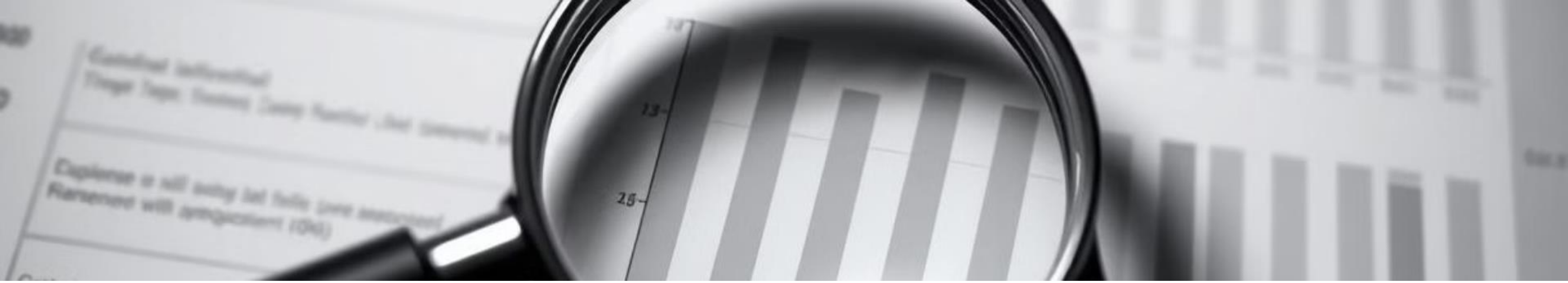
Model Selection

Performance metrics



Model Selection

- Model selection is the process of choosing the best machine learning model from a set of potential models for a given dataset and problem
- This process involves evaluating the performance of different models using specific criteria or performance metrics and selecting the model that performs best according to those metrics



Importance of Model Selection

Selecting the right model ensures optimal performance, accurate predictions, and effective insights

1. **Accuracy**

   The selected model should accurately reflect the underlying patterns in the data.

2. **Generalizability**

   It should generalize well to unseen data, avoiding overfitting to the training data.

3. **Interpretability**

   The model's insights should be readily understandable and actionable.

4. **Efficiency**

   The model should be computationally efficient and scalable for large datasets.

ML Algorithms

Decision Trees

Decision trees represent a series of decisions, modeled as a tree-like structure, where each node represents a feature, and each branch represents a possible outcome.

Root Node

Represents the starting point of the decision-making process.

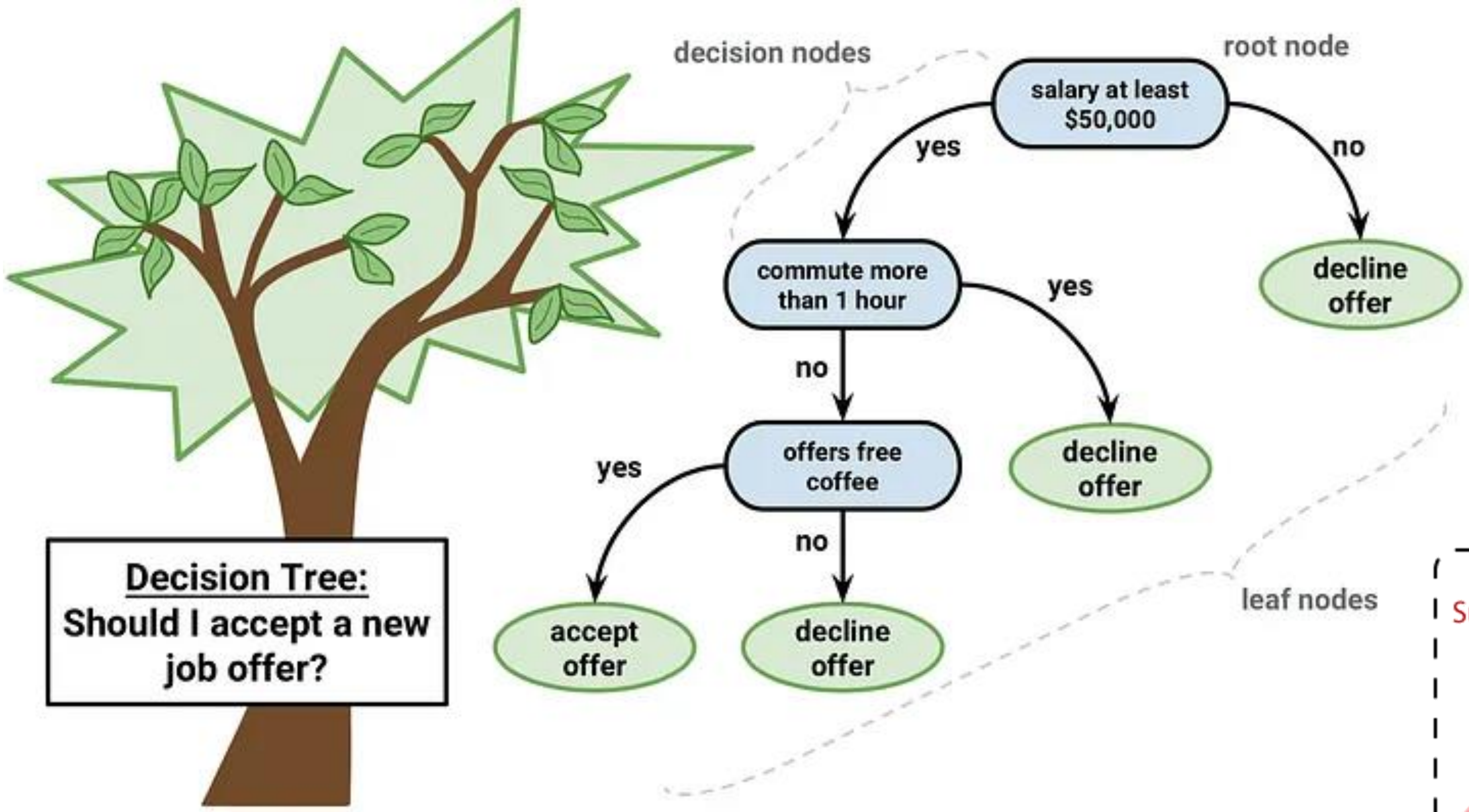
Internal Nodes

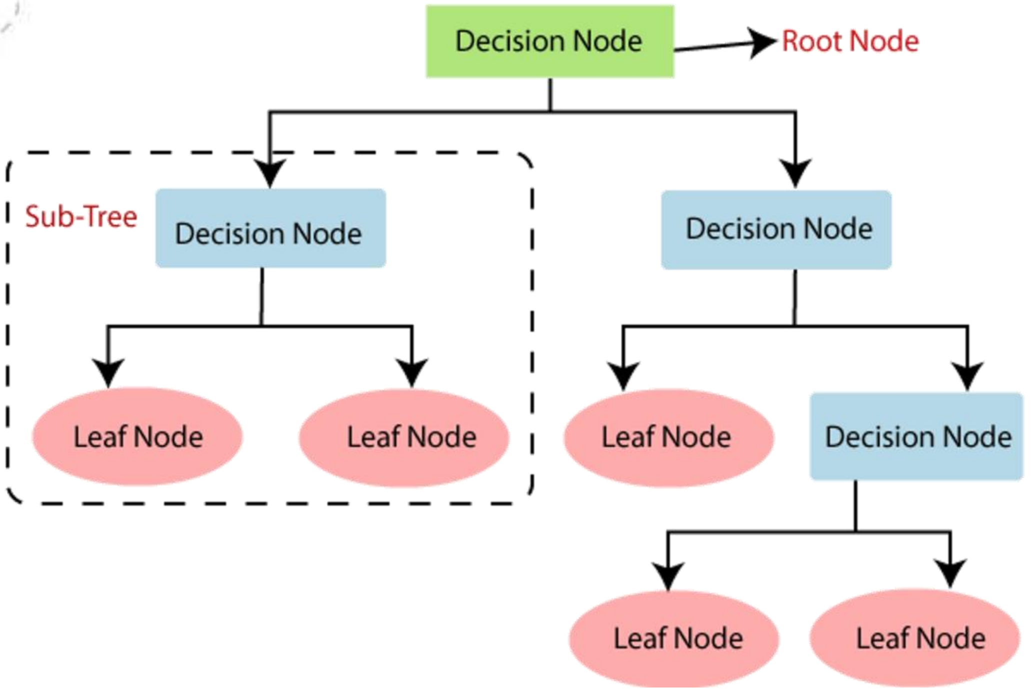
Represent the features used to split the data.

Leaf Nodes

Represent the final predictions or outcomes.

Decision Tree





Advantages and Disadvantages of Decision Trees

Decision trees are relatively easy to understand and interpret, making them popular in various applications. However, they can be prone to overfitting, leading to poor generalization performance.

Advantages

Easy to understand and interpret.

Can handle both categorical and numerical data.

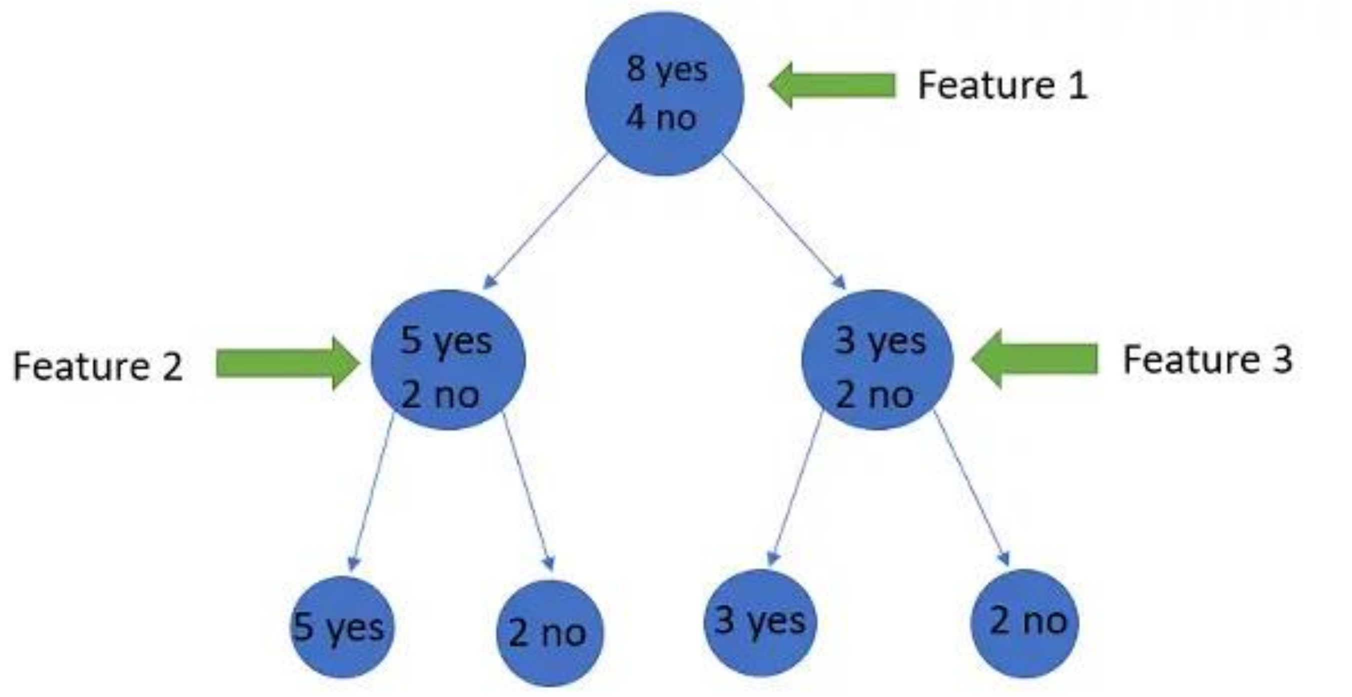
Can be used for both classification and regression tasks.

Disadvantages

Prone to overfitting, leading to poor generalization performance.

Sensitive to small changes in the data.

Can be biased towards features with more categories.



Introduction to Random Forests

Random forests are an ensemble learning method that combines multiple decision trees to improve prediction accuracy and reduce overfitting.

1. **Ensemble Learning**

   Combines multiple models to improve performance.

2. **Multiple Decision Trees**

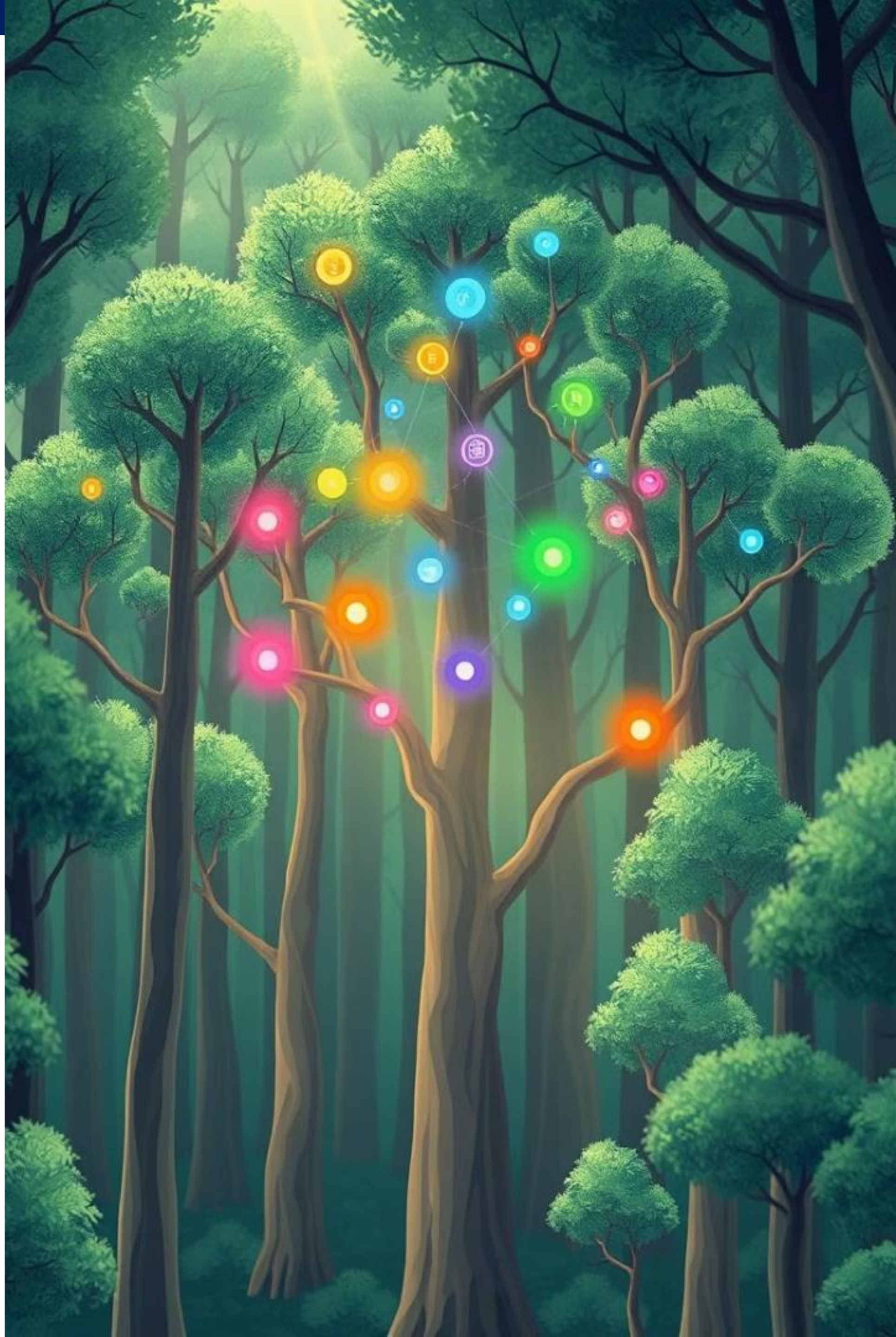
   Constructs multiple decision trees, each trained on a random subset of the data.

3. **Bagging**

   Uses bootstrap aggregation to create multiple training sets.

4. **Random Subspace**

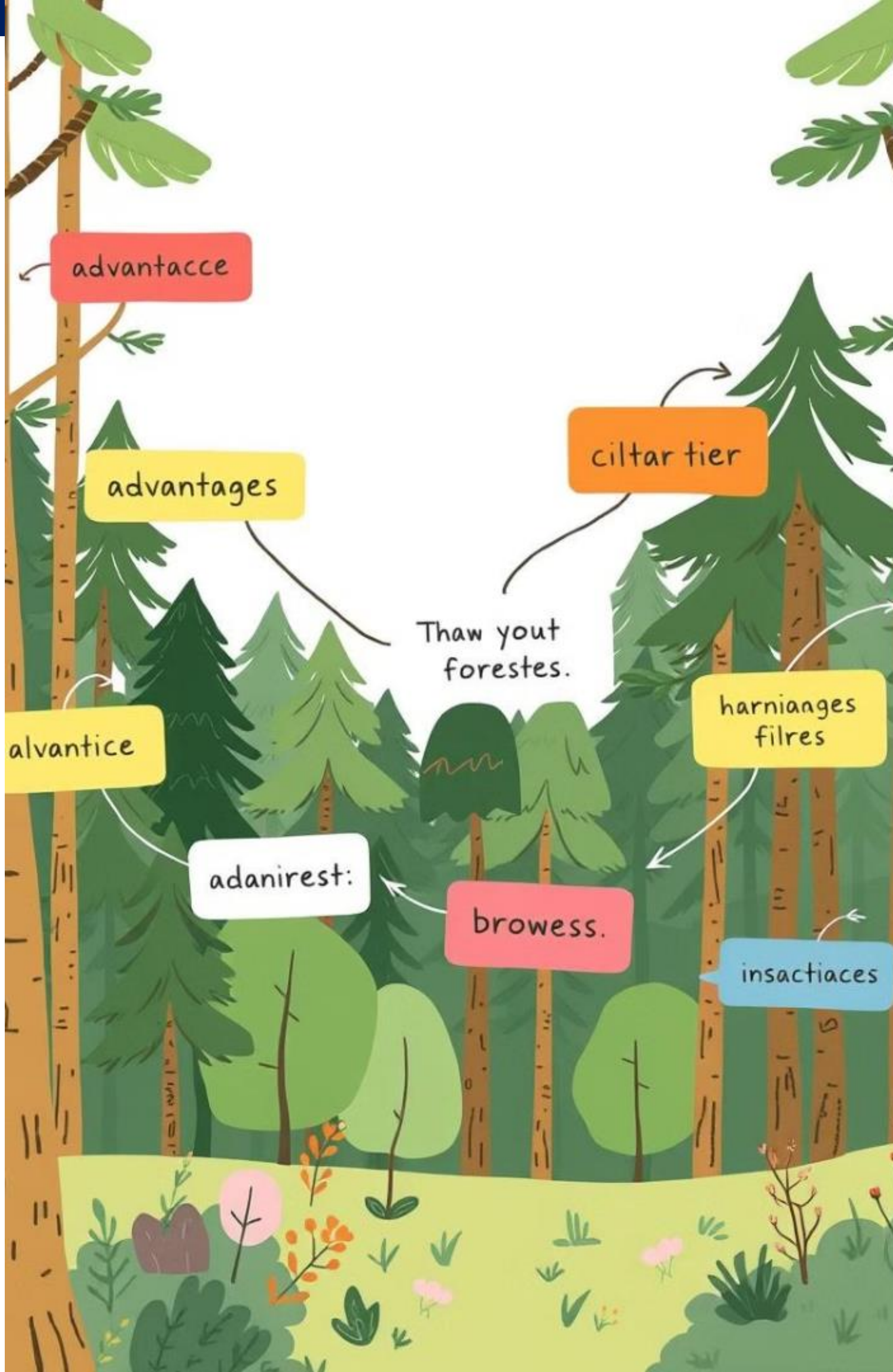
   Randomly selects a subset of features for each tree.



Advantages and Disadvantages of Random Forests

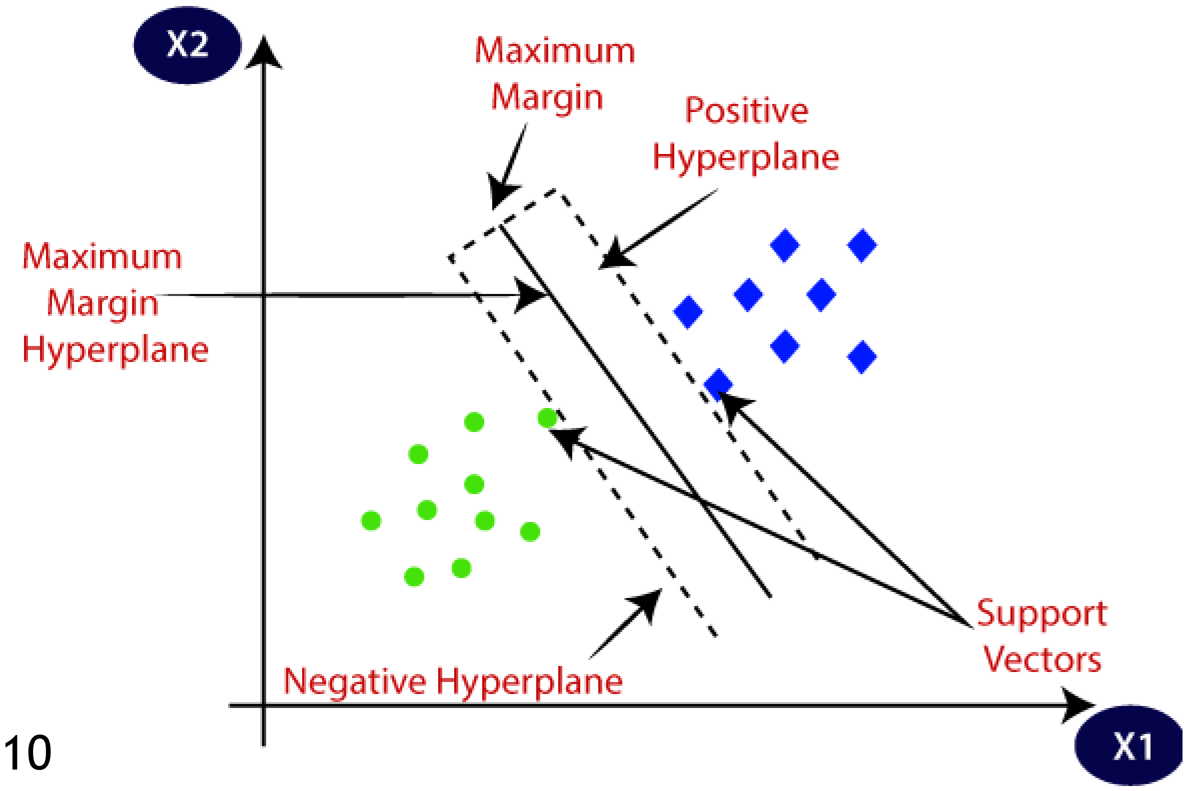
Random forests generally outperform single decision trees by reducing variance and improving prediction accuracy. However, they can be more computationally expensive to train than decision trees.

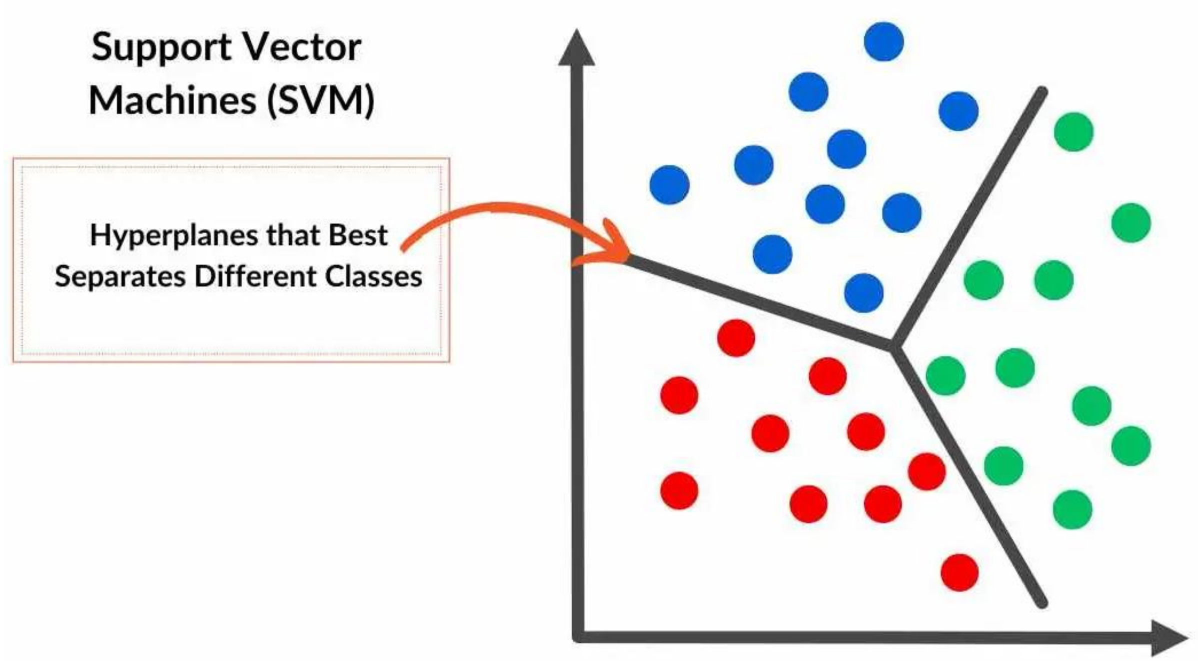
Advantages	Disadvantages
Reduced overfitting	More computationally expensive
Improved prediction accuracy	Can be difficult to interpret
Handles high dimensional data well	Not suitable for small datasets



Support Vector Machine

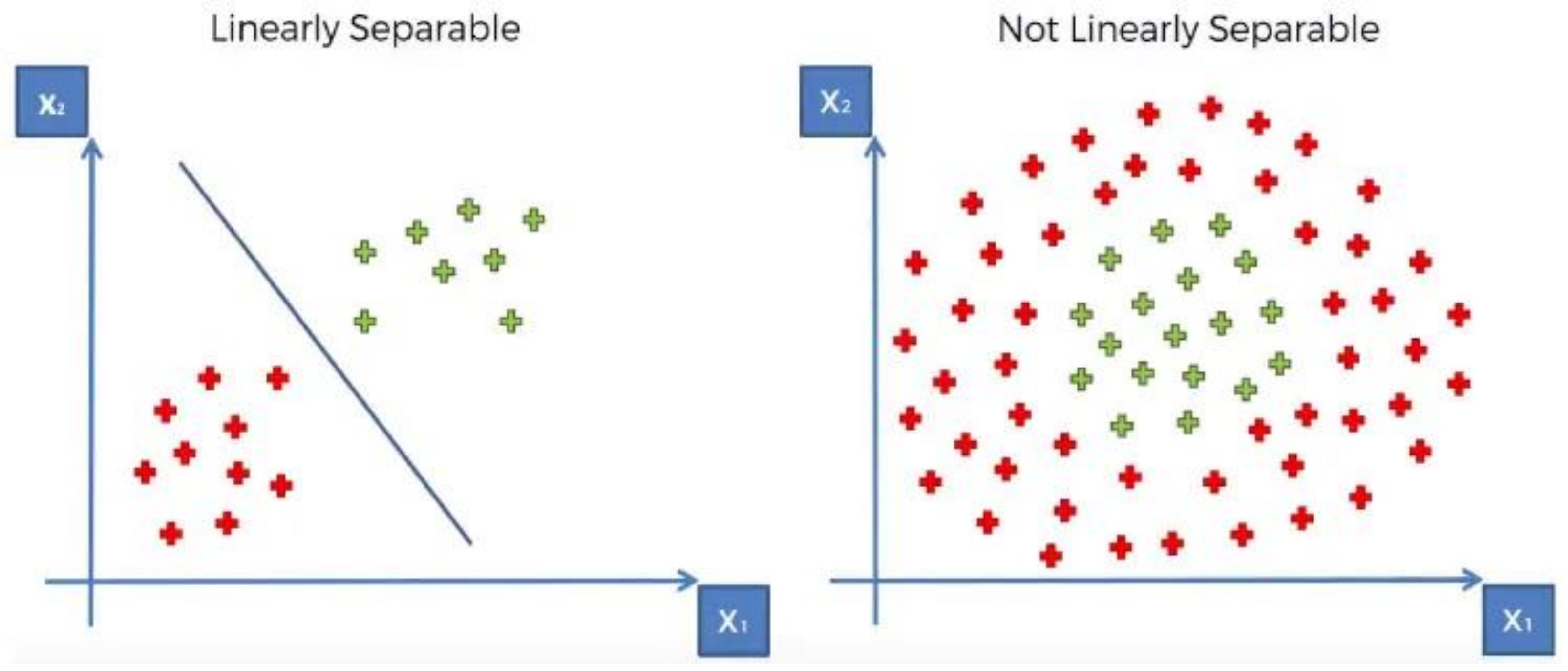
- Support Vector Machine (SVMs) is a supervised learning algorithm that aims to find the optimal hyperplane that best separates the data points of different classes in the feature space
- The hyperplane is chosen to maximize the margin — the distance between the hyperplane and the nearest data points from each class, called support vectors




Support Vector Machines (SVM)
Hyperplanes that Best Separates Different Classes

Linearly Separability

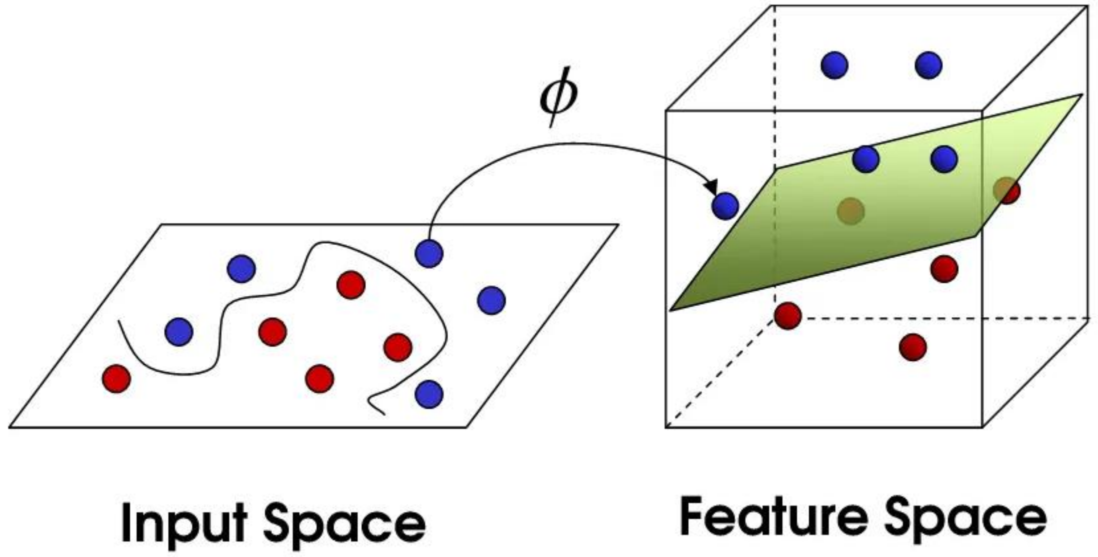
- A dataset is linearly separable if there exists a straight line (in 2D), a plane (in 3D), or a hyperplane (in higher dimensions) that can perfectly separate the data points of different classes without any overlap.

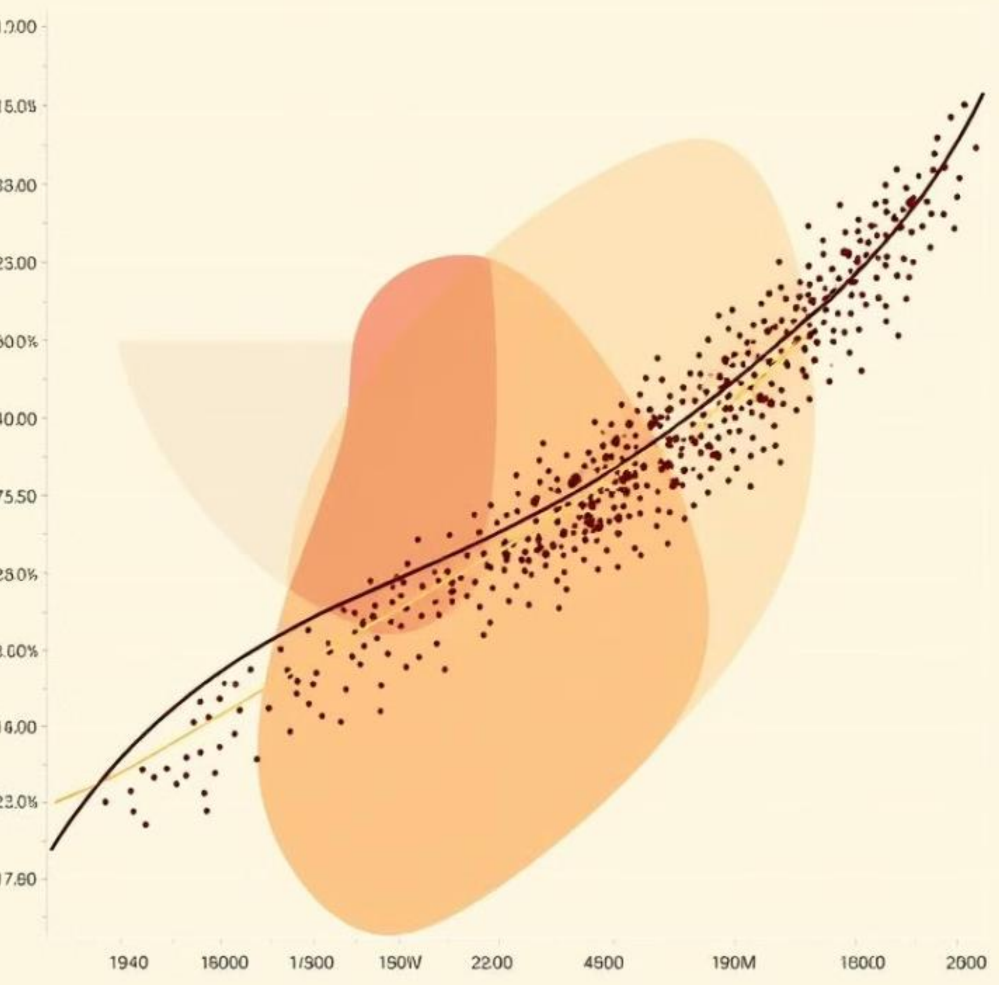


SVM Kernel

- One way to make non-linear data separable is to map it into a higher-dimensional space where a linear separator does exist
  - For example, mapping 2D points into a 3D space can transform concentric circles into linearly separable shapes

- Kernels help by implicitly mapping the original feature space into a higher-dimensional space where the data might be more easily separable





Carnegie Mellon

Advantages of SVMs

SVMs offer several advantages over other machine learning algorithms, making them a popular choice for many applications.

High Accuracy

SVMs are known for their excellent performance on both linear and non-linear data.

Robustness to Outliers

The margin maximization approach makes them less susceptible to noise and outliers.

Effective in High-Dimensional Spaces

They can handle large datasets with numerous features and complex relationships.

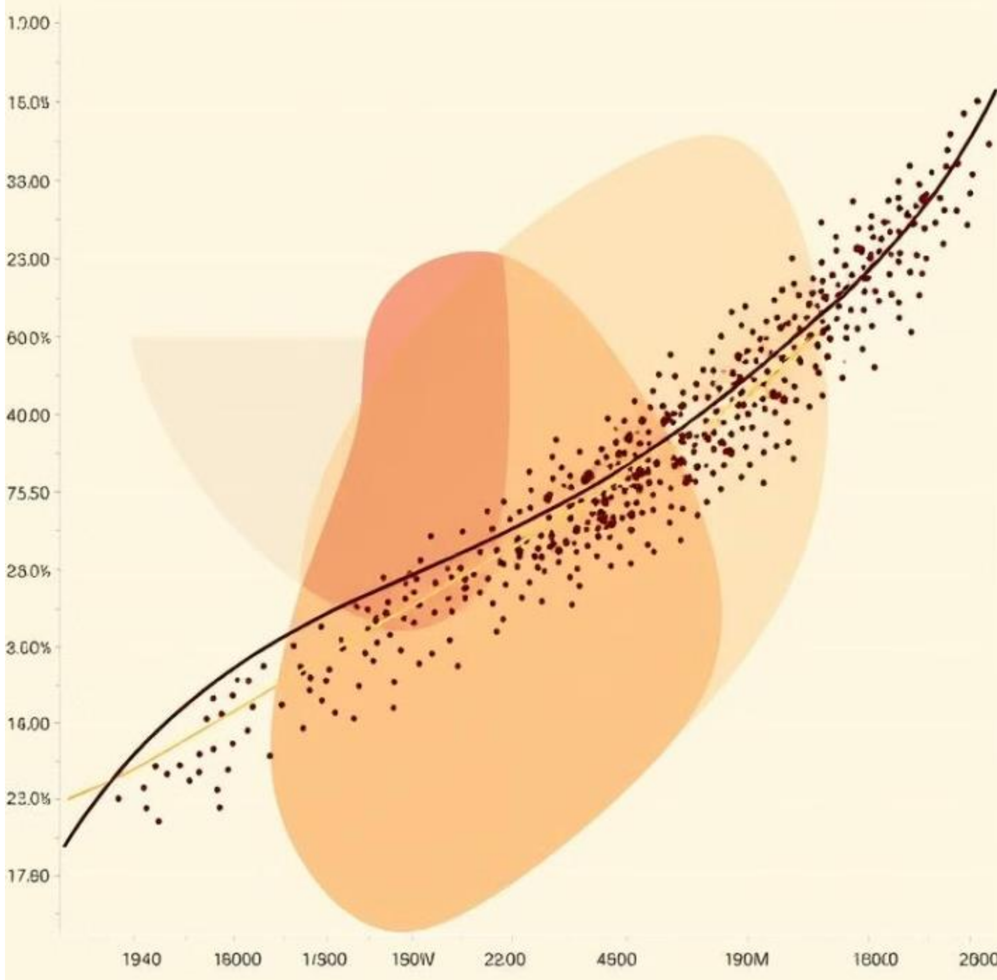
Memory Efficiency

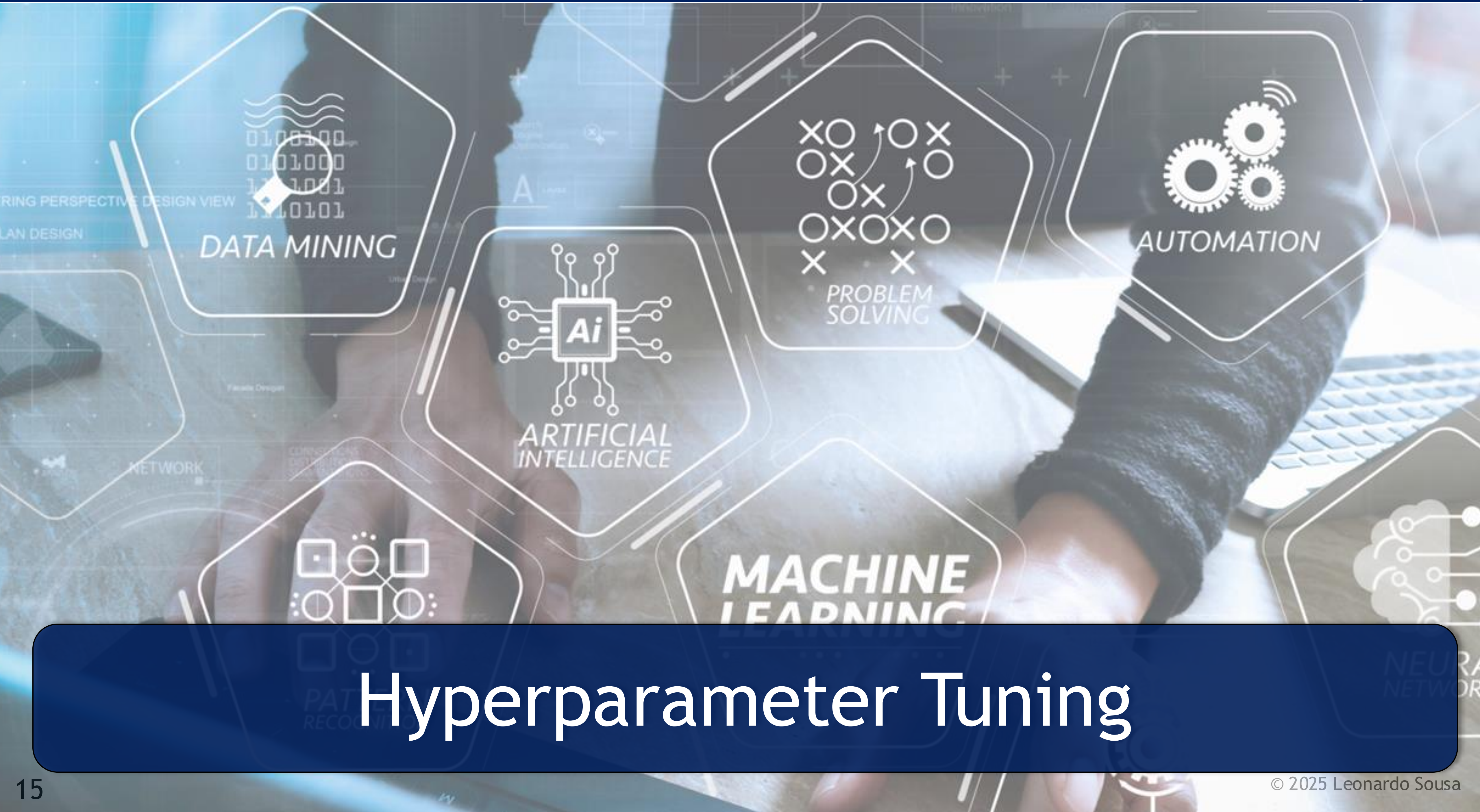
SVMs only need to store support vectors, making them memory-efficient compared to other models.

Limitations and Challenges

While SVMs have many advantages, they also have limitations that need to be considered when applying them to specific problems.

High Computational Cost	Training can be time-consuming for large datasets, especially for non-linear kernel functions.
Parameter Tuning	Finding the optimal values for kernel parameters and regularization terms requires careful tuning.
Sensitivity to Data Scaling	Performance can be affected by the scale of features, necessitating pre-processing techniques.

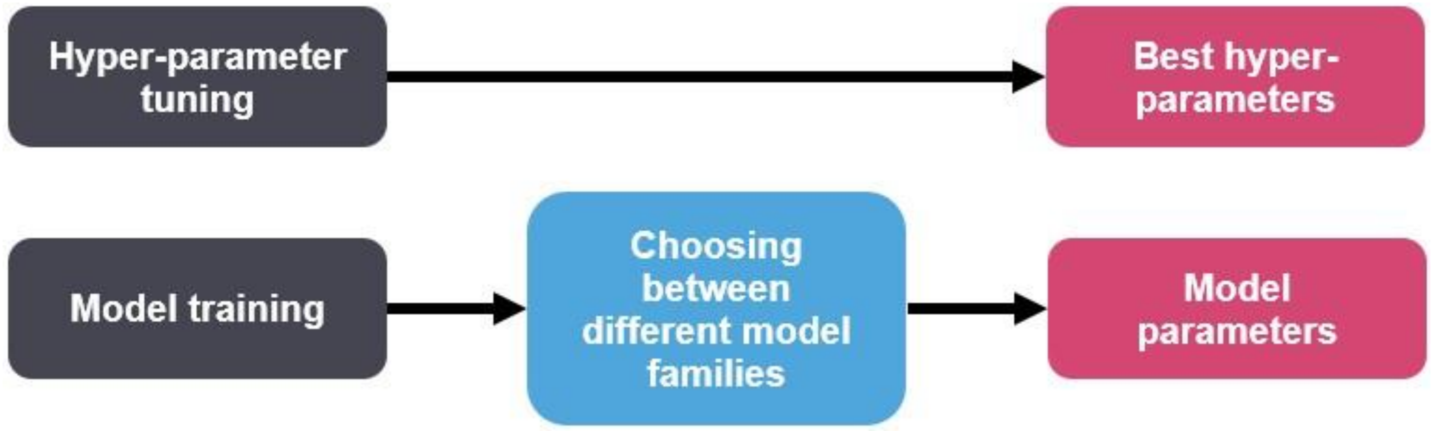




Hyperparameter Tuning

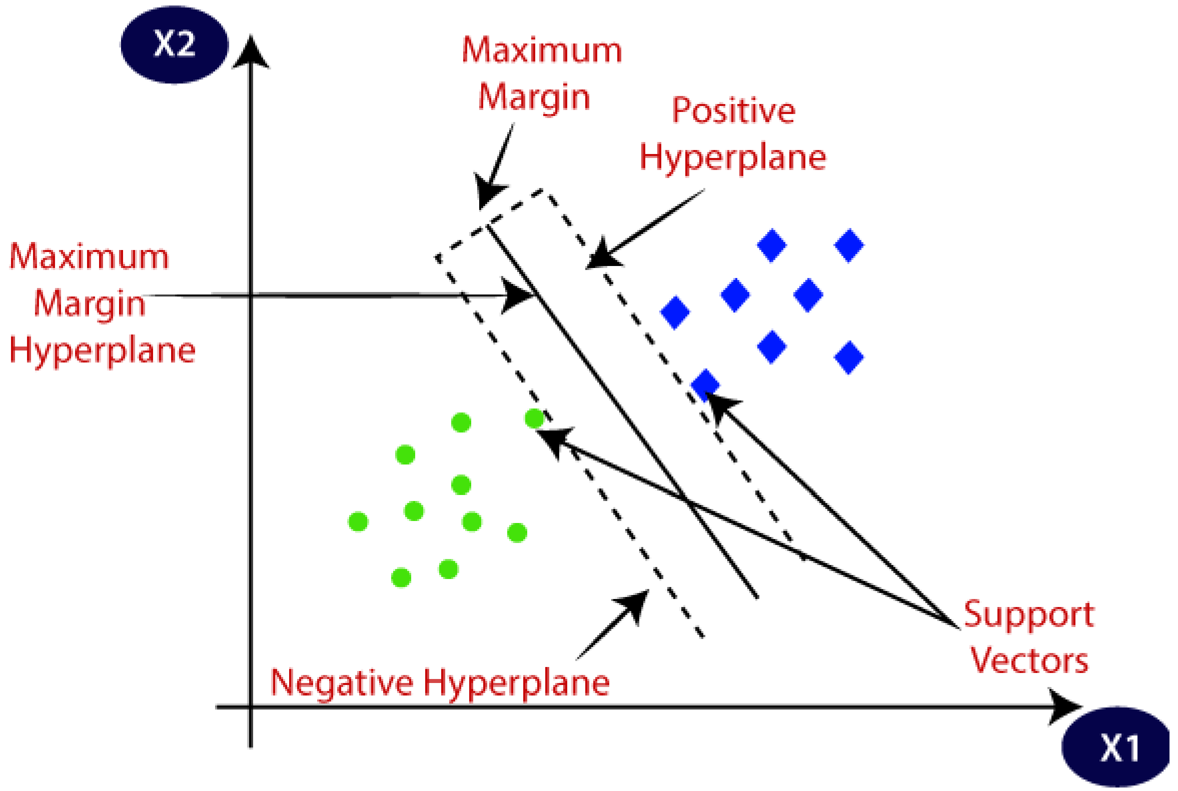
Hyperparameter Tuning

- Hyperparameter tuning is a critical step in the machine learning workflow
- Optimizing the parameters of a model that are not learned from data
- These hyperparameters control the learning process and the structure of the machine learning model itself, affecting its performance on given tasks
- Unlike model parameters, which are learned during training, hyperparameters are set prior to the training process and remain constant during it



Parameter vs Hyperparameters

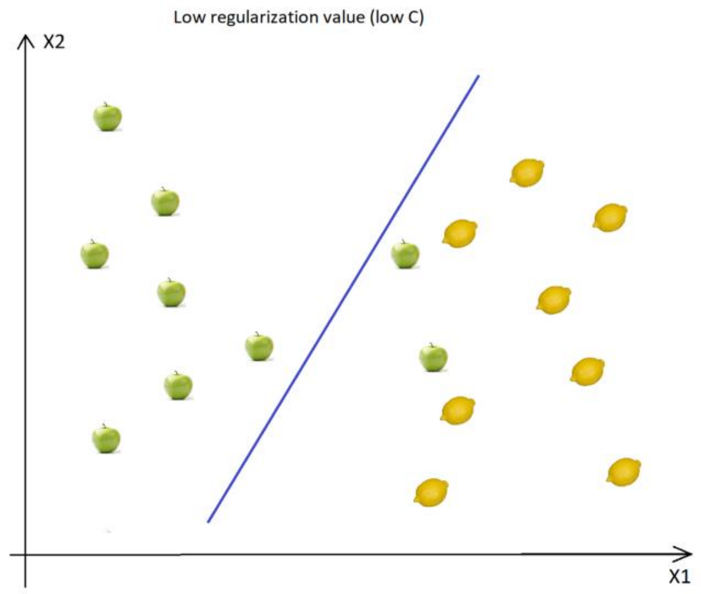
- Model **parameter** is a configuration variable that is internal to the model
  - Its value can be estimated from data
  - E.g., *weights* in a neural network or *support vectors* in SVM

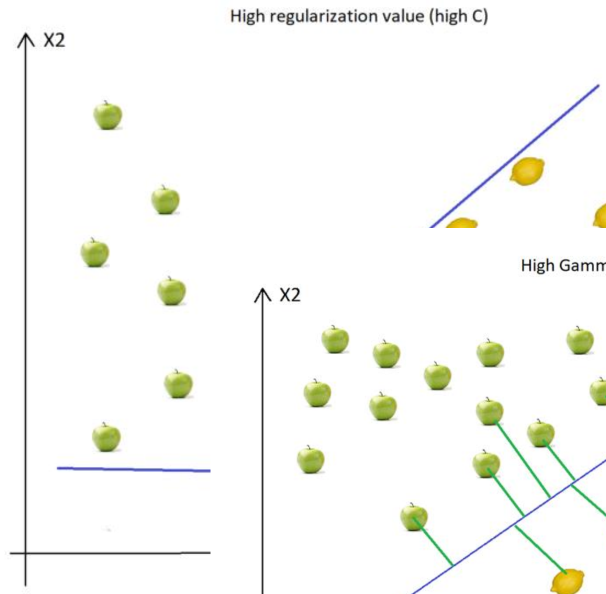


SVM tries to find the best hyperplane to separate the different classes by maximizing the distance between sample points and the hyperplane

Parameter vs Hyperparameters

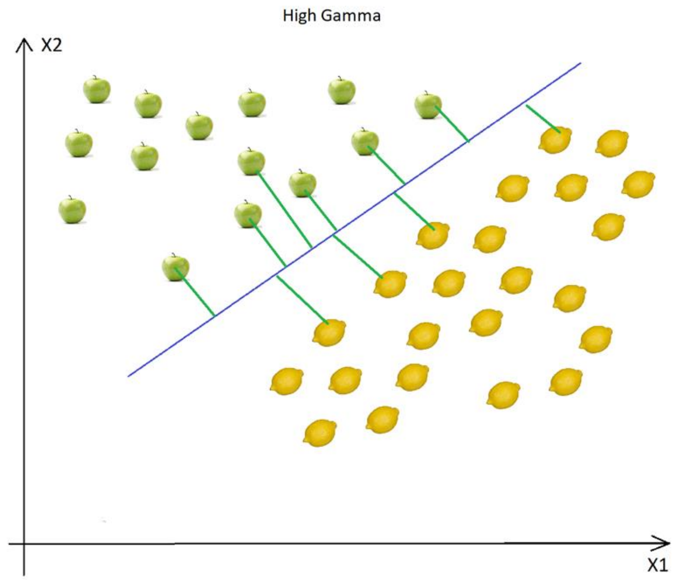
- Model **hyperparameter** is a configuration that is external to the model
  - Its value cannot be estimated from data
  - E.g., *splitting criterion* in Decision Tree, *learning rate* in a neural network, *C* and *sigma* in SVM

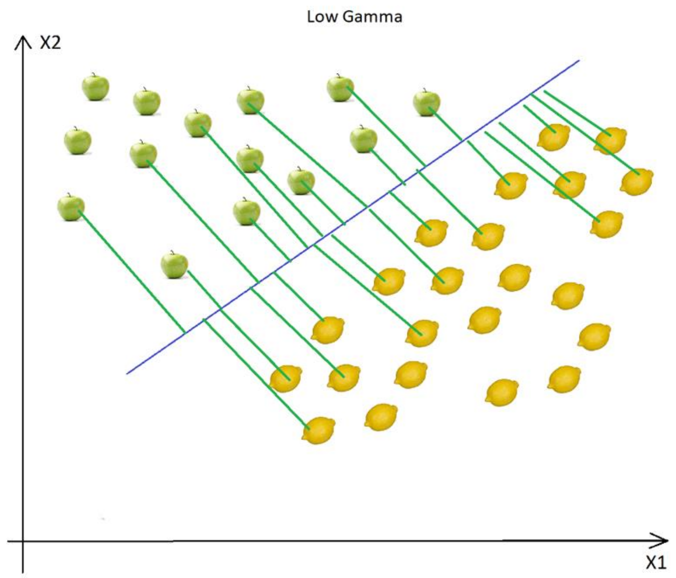




Degree to avoid misclassifications

How far the influence of instances go

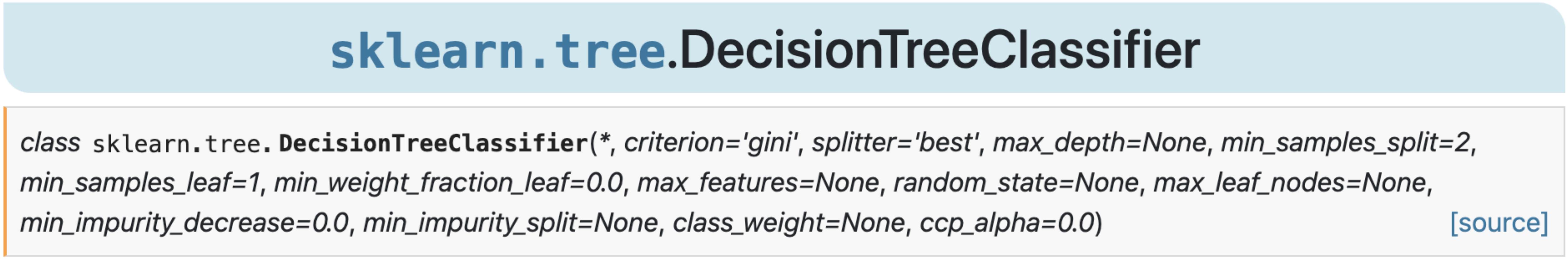




Carnegie Mellon

Tuning a Decision Tree

- Hyperparameter tuning
  - The process of searching the hyperparameter space for a set of values that will optimize your model architecture



sklearn.tree.DecisionTreeClassifier

class sklearn.tree.**DecisionTreeClassifier**(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0*) [source]

Decision Tree: Hyperparameter Example

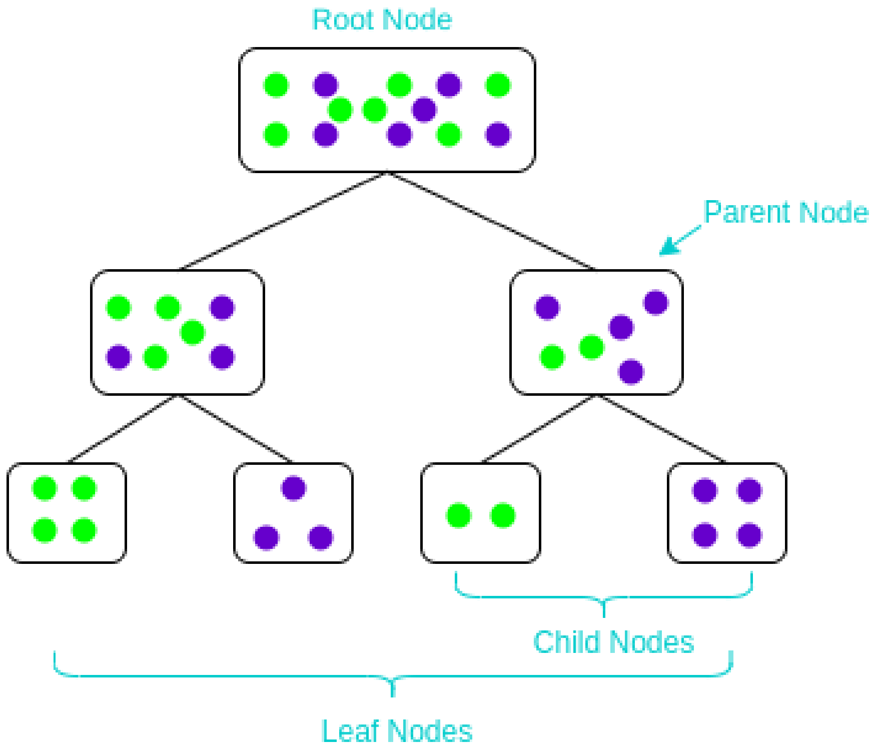
- Splitting criterion is called the best when after splitting, each partition will be **pure**
  - **Information Gain** measures the reduction in entropy (or uncertainty) before and after a dataset is split on an attribute
    - The attribute with the highest Information Gain is chosen for the split

Entropy

The **entropy** is calculated using the following formula:

$$Entropy = -\sum_{j} p_j \cdot log_2 \cdot p_j$$

Where, as before, p_j is the probability of class j.



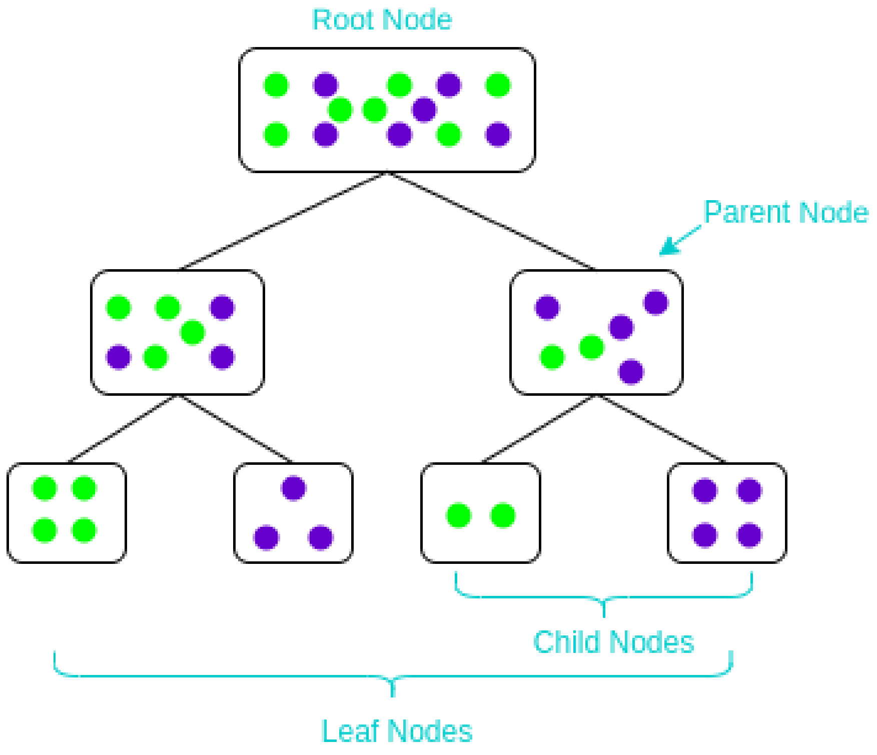
Decision Tree: Hyperparameter Example

- Splitting criterion is called the best when after splitting, each partition will be **pure**
  - **Gini Index** measures the impurity or diversity of an attribute in the dataset.
    - An index of 0 indicates that all the cases in the node fall into a single category
    - An index close to 1 indicates a random distribution of classes

The **gini impurity** is calculated using the following formula:

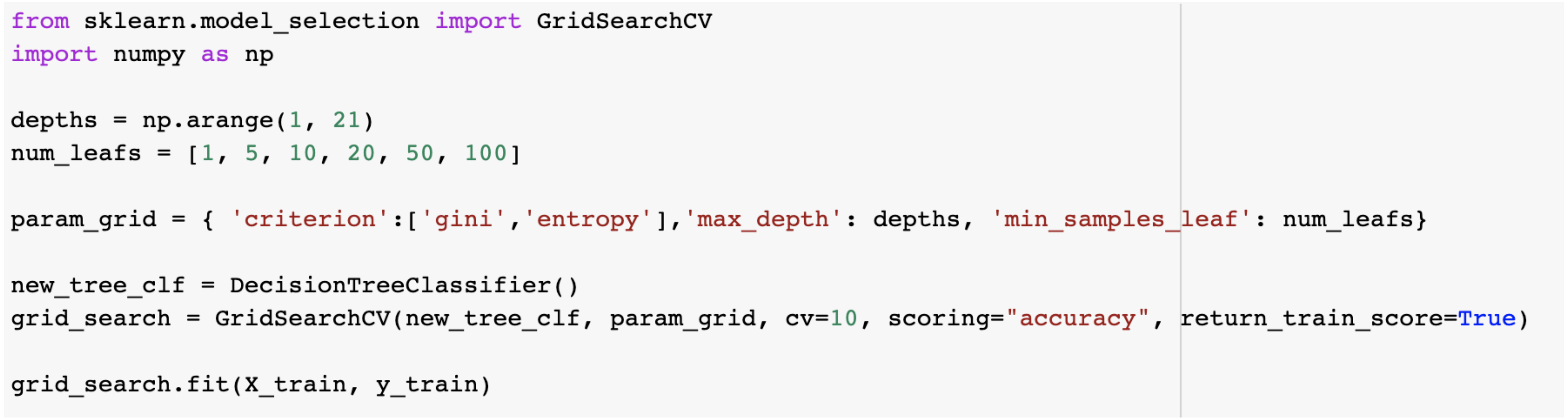
$$GiniIndex = 1 - \sum_j p_j^2$$

Where p_j is the probability of class j.



Tuning a Decision Tree: Grid Search

- Grid search is the process of performing hyper parameter tuning to determine the optimal values for a given model
  - the set of trials is formed by assembling every possible combination of values (combination of hyperparameters)



```python
from sklearn.model_selection import GridSearchCV
import numpy as np

depths = np.arange(1, 21)
num_leafs = [1, 5, 10, 20, 50, 100]

param_grid = { 'criterion':['gini','entropy'],'max_depth': depths, 'min_samples_leaf': num_leafs}

new_tree_clf = DecisionTreeClassifier()
grid_search = GridSearchCV(new_tree_clf, param_grid, cv=10, scoring="accuracy", return_train_score=True)

grid_search.fit(X_train, y_train)
```

Model Evaluation

Train-Test Split

Process

The data is randomly divided into two sets: a training set (usually 70-80% of the data) and a test set (20-30% of the data).

Training

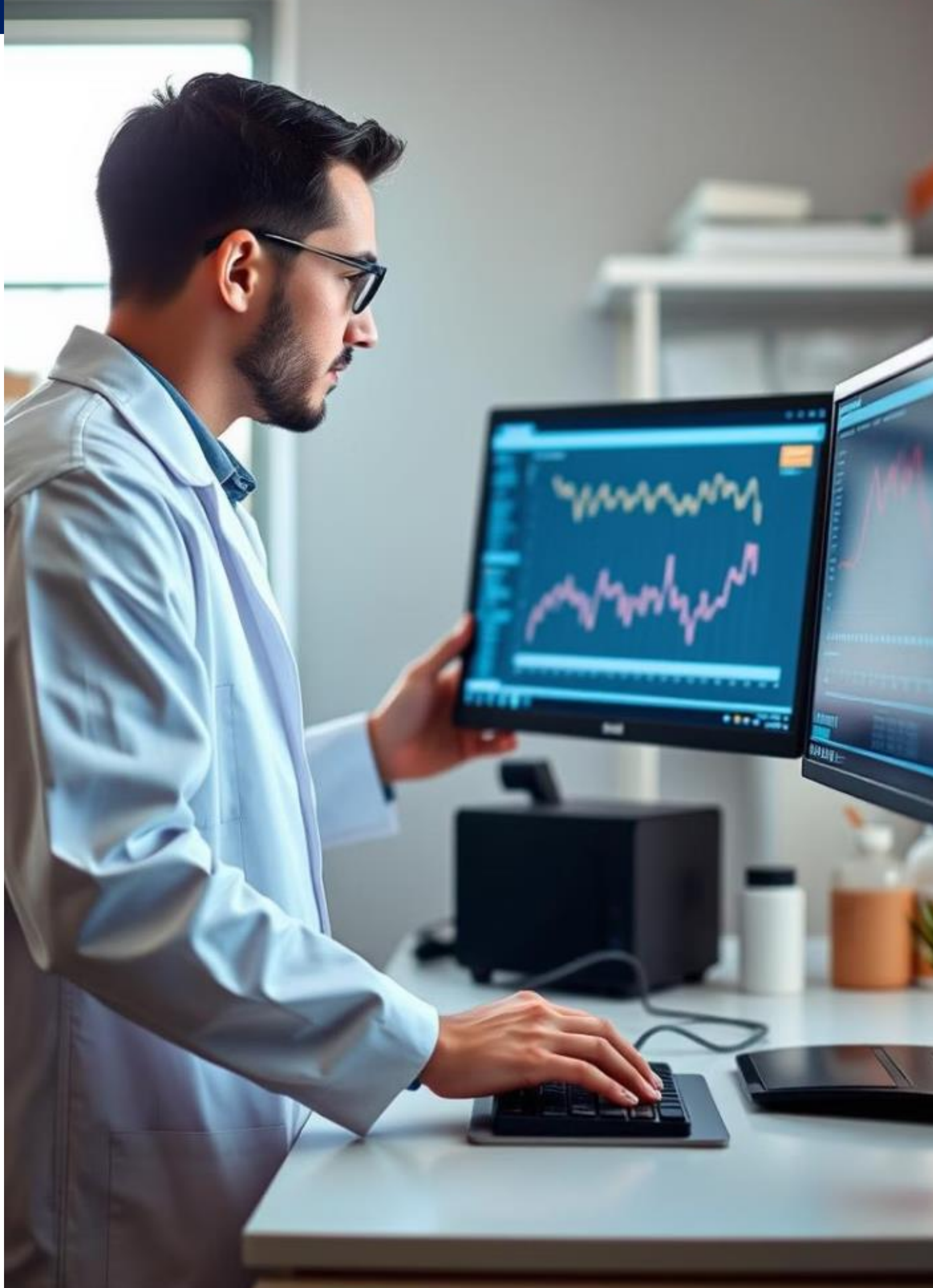
The model is trained on the training set, learning the underlying patterns and relationships in the data.

Evaluation

The trained model is then evaluated on the test set, which it has never seen before, to assess its performance and generalization ability.

Evaluation Metrics in Machine Learning

- Evaluation or performance metrics are essential for measuring the performance of machine learning models
- They provide insights into how well a model is performing and help in choosing the best model for a given task
- The choice of metrics depends on the specific type of problem (classification, regression, clustering, etc.) and the goals of the model



Terminology

- What is Positive and Negative?
  - They represent the classes
    - E.g., buggy and non-buggy

- What is **True** and **False**?
  - They represent the model’s prediction

  - “**True**” represents the instances that the model identified its class
  - “**False**” represents the instances that the model was not able to identify

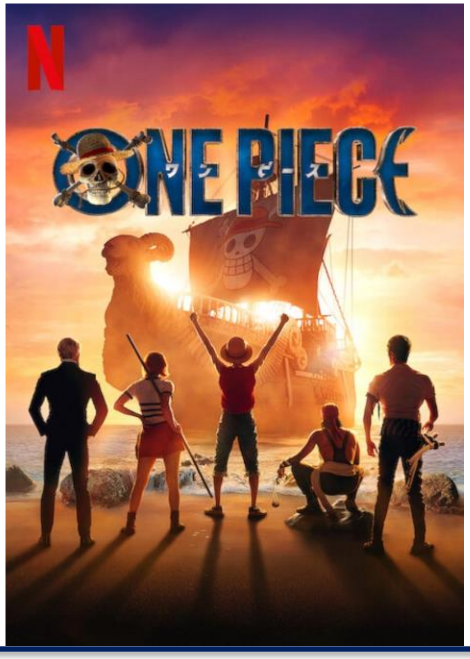
HBO Example

- Train a model to classify shows according to their streaming service
  - E.g., NETFLIX and HBO

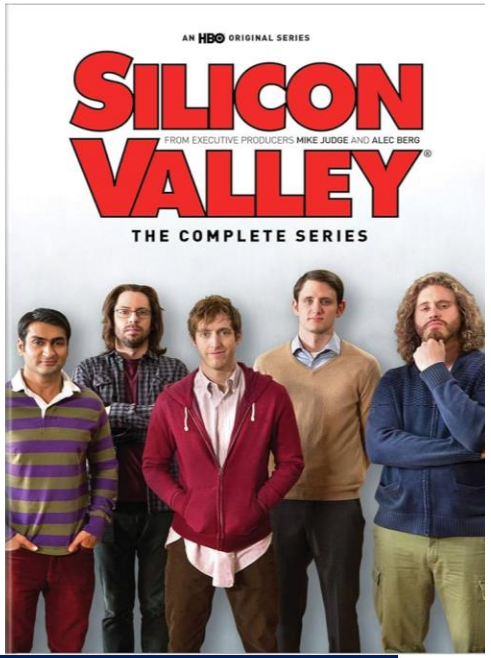
NETFLIX

HBO

THE WAIT HAS BEEN TORTURE
FROM VISIONARY TIM BURTON
WEDNESDAY
SEASON 2
ONLY ON NETFLIX | TEASER TOMORROW



FROM THE DIRECTOR OF SUPERMAN
DC STUDIOS
PEACEMAKER
IN WHAT WORLD ARE THEY THE HEROES?
NEW SEASON AUG 21
HBO max



Model

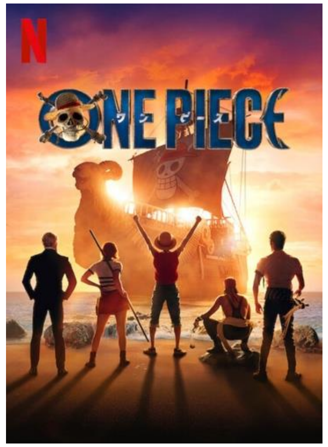
- Classes:
  - HBO (positive) = 1 ⟸ *reference point*
  - NETFLIX (negative)= 0 ⟸ *NOT HBO*

- Classify Peacemaker as HBO and is HBO show → True positive
- Classify One Piece as NOT HBO show and is not an HBO show → True negative

- Classifies One Piece as HBO but is not an HBO show → False positive
- Classifies Peacemaker as NOT HBO but is an HBO show → False negative

FROM THE DIRECTOR OF SUPERMAN
DC STUDIOS
PEACEMAKER
IN WHAT WORLD ARE THEY THE HEROES?
NEW SEASON AUG 21
HBO max

HBO (positive)



NETFLIX (negative)

Confusion Matrix

A confusion matrix is a visual representation of the performance of a classification model.

True Positive (TP)

Correctly predicted positive instances.

True Negative (TN)

Correctly predicted negative instances.

False Positive (FP)

Incorrectly predicted positive instances.

False Negative (FN)

Incorrectly predicted negative instances.

		Predicted HBO	
		YES	NO
Actual HBO	YES	TP: 25	FN: 10
	NO	FP: 5	TN: 65

Accuracy

- Accuracy is the most basic evaluation metric.
- It measures the proportion of correctly classified instances out of the total instances

1 **Simple and Widely Used**

Accuracy is a commonly used metric due to its simplicity.

2 **Limited in Imbalanced Datasets**

Accuracy can be misleading in imbalanced datasets where one class dominates the other

Accuracy

- Accuracy: the measure of all the correctly identified cases

$$ACC = \frac{TP\ +\ TN}{TP + FP + TN + FN}$$

$$ACC = \frac{25\ + 65}{25 + 5 + 65 + 10}$$

$$ACC = \frac{90}{105} = 0.8571\ (86\%)$$

		Predicted HBO	
		YES	NO
Actual HBO	YES	TP: 25	FN: 10
	NO	FP: 5	TN: 65

Precision

Precision measures the proportion of correctly classified positive instances out of all instances predicted as positive.

Focus on Positive Predictions

Precision prioritizes the accuracy of positive predictions, minimizing false positives.

Example: Spam Detection

In spam detection, high precision means fewer legitimate emails are mistakenly labeled as spam.

Recall

Recall measures the proportion of correctly classified positive instances out of all actual positive instances.

1

Focus on Actual Positives

Recall ensures that the model correctly identifies most of the positive instances.

2

Example: Medical Diagnosis

In medical diagnosis, high recall ensures that the model correctly identifies most patients with a specific disease.

Precision and Recall

- Precision: correctly identified positive cases from all predicted positive cases

$$precision = \frac{TP}{TP + FP} = \frac{25}{25 + 5} = \frac{25}{30} = 0.83\ (83\%)$$

- Recall: correctly identified positive cases from all the actual positive cases

$$recall = \frac{TP}{TP + FN} = \frac{25}{25 + 10} = \frac{25}{35} = 0.71\ (71\%)$$

		Predicted HBO	
		YES	NO
Actual HBO	YES	TP: 25	FN: 10
	NO	FP: 5	TN: 65

F1-Score

The F1-score combines precision and recall into a single metric, providing a balanced measure of model performance.

High Precision	Low Recall	F1-Score is low
Low Precision	High Recall	F1-Score is low
High Precision	High Recall	F1-Score is high

F1-Score

- Harmonic mean of Precision and Recall
  - it penalizes the extreme values
  - It gives a better measure of the incorrectly classified cases than the Accuracy Metric

$$F1_{score} = \left(\frac{recall^{-1} + precision^{-1}}{2}\right)^{-1} = 2 * \frac{precision * recall}{precision + recall} = 2 * \frac{(0.83 * 0.71)}{(0.83 + 0.71)}$$
$$= \ 0.76\ (76\ \%)$$

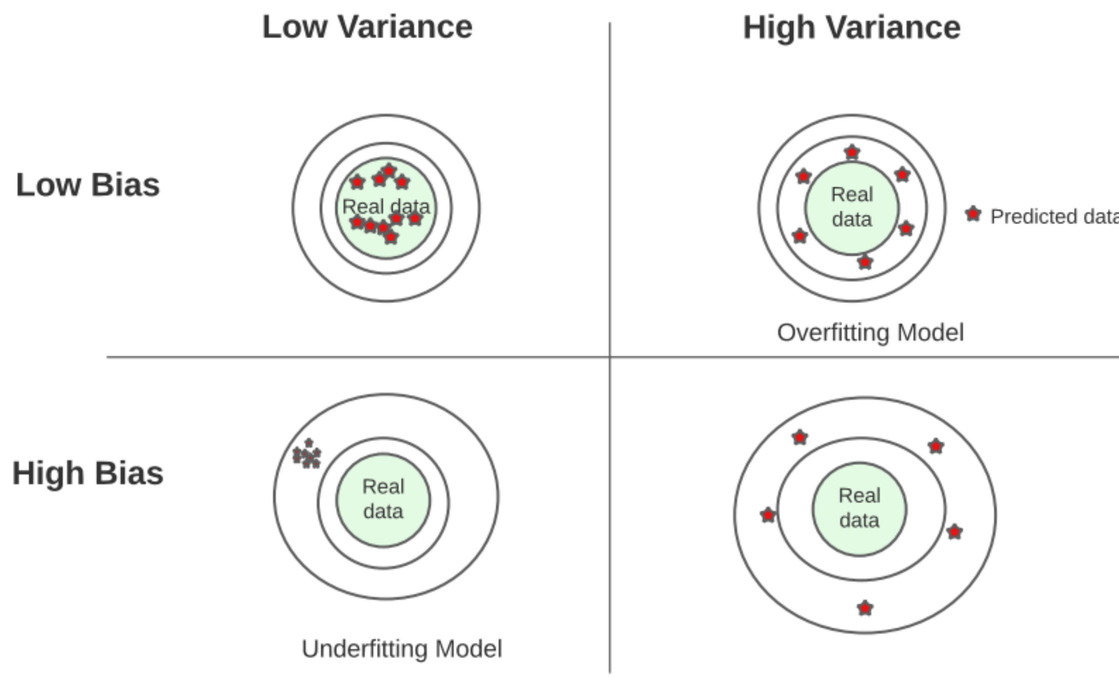
		Predicted Comedy	
		YES	NO
Actual Comedy	YES	TP: 25	FN: 10
	NO	FP: 5	TN: 65

Training a ML model

Bias and Variance

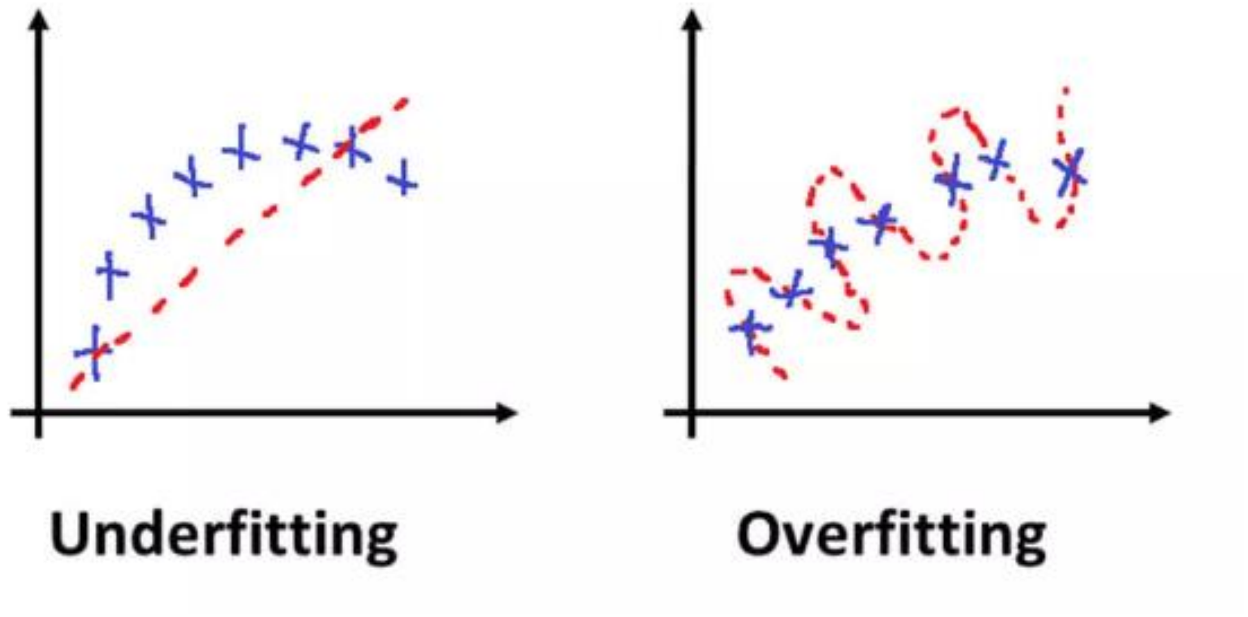
- **Bias** is the difference between the average prediction and the correct value which we are trying to predict
  - Model with high bias pays very little attention to the training data and oversimplifies the model
  - It always leads to high error on training and test data.

- **Variance** refers to the model's sensitivity to fluctuations in the training data
  - It indicates how much the model's predictions would change if it were trained on a different dataset drawn from the same distribution.



Overfitting and Underfitting

- Overfitting occurs when the model learns the training data too well, resulting in poor performance on unseen data
- Underfitting occurs when the model is too simple and fails to capture the underlying patterns



Overfitting and Underfitting

Overfitting

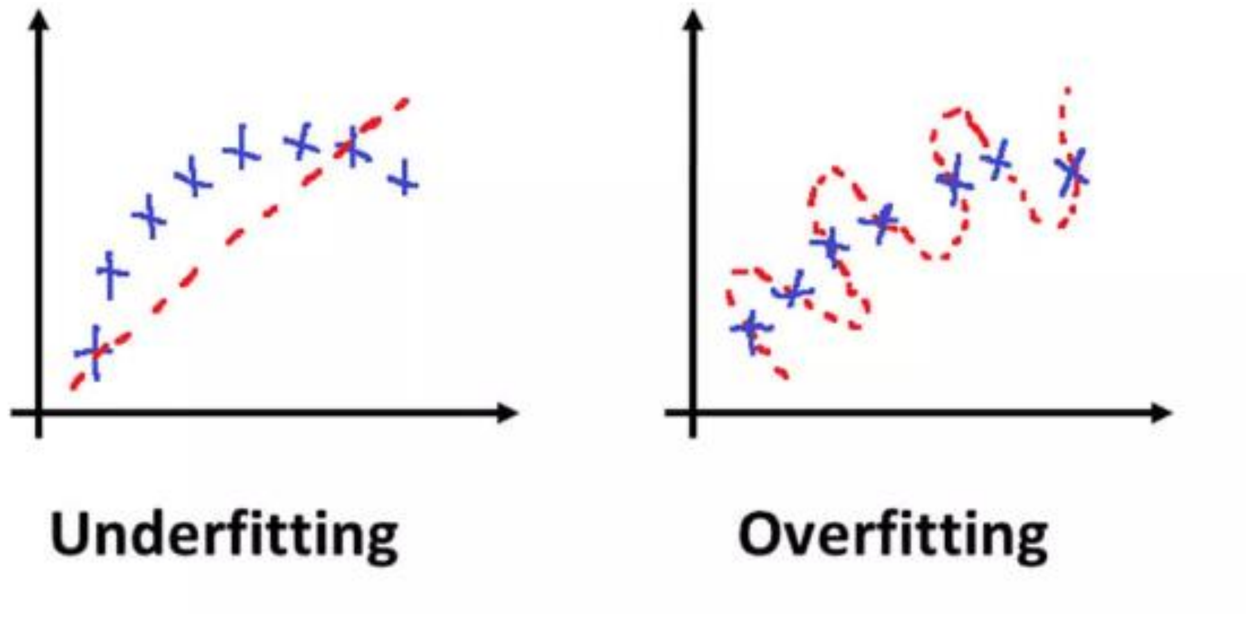
High variance, low bias

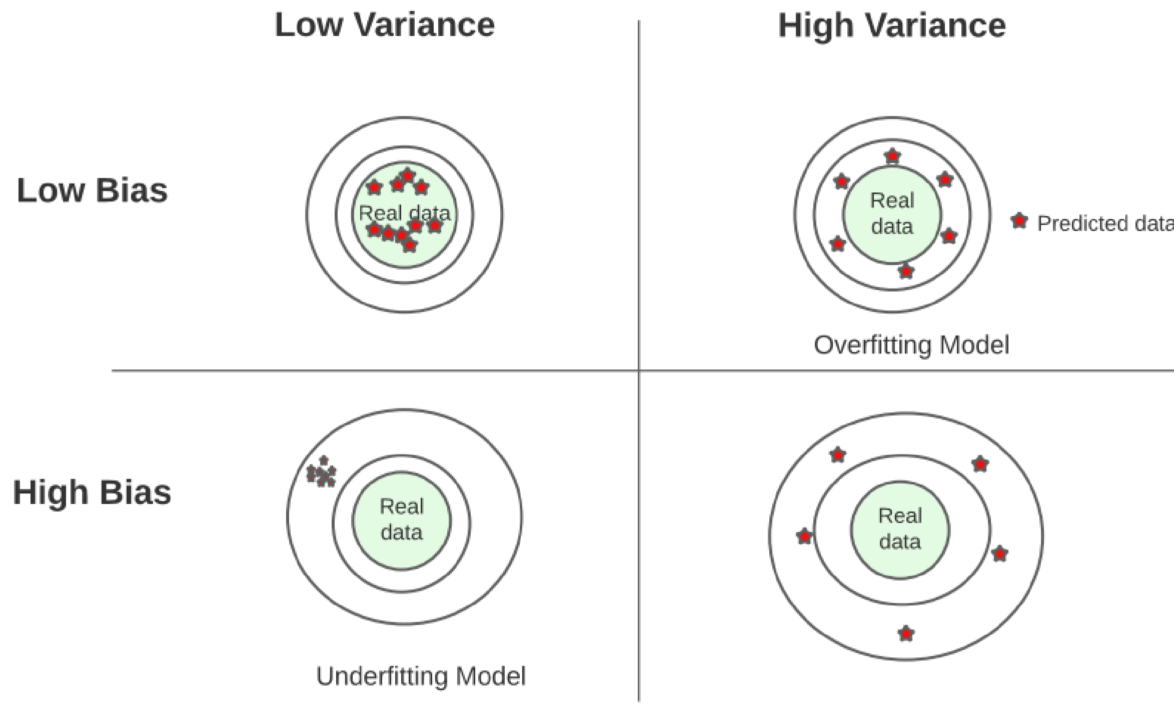
- Complex model
- Poor generalization
- Sensitive to noise

Underfitting

Low variance, high bias

- Simple model
- Poor accuracy
- Fails to capture patterns





Bias-Variance Tradeoff

- The bias-variance tradeoff is a fundamental concept in model selection
- It represents the balance between model complexity and its ability to generalize

1. High Bias

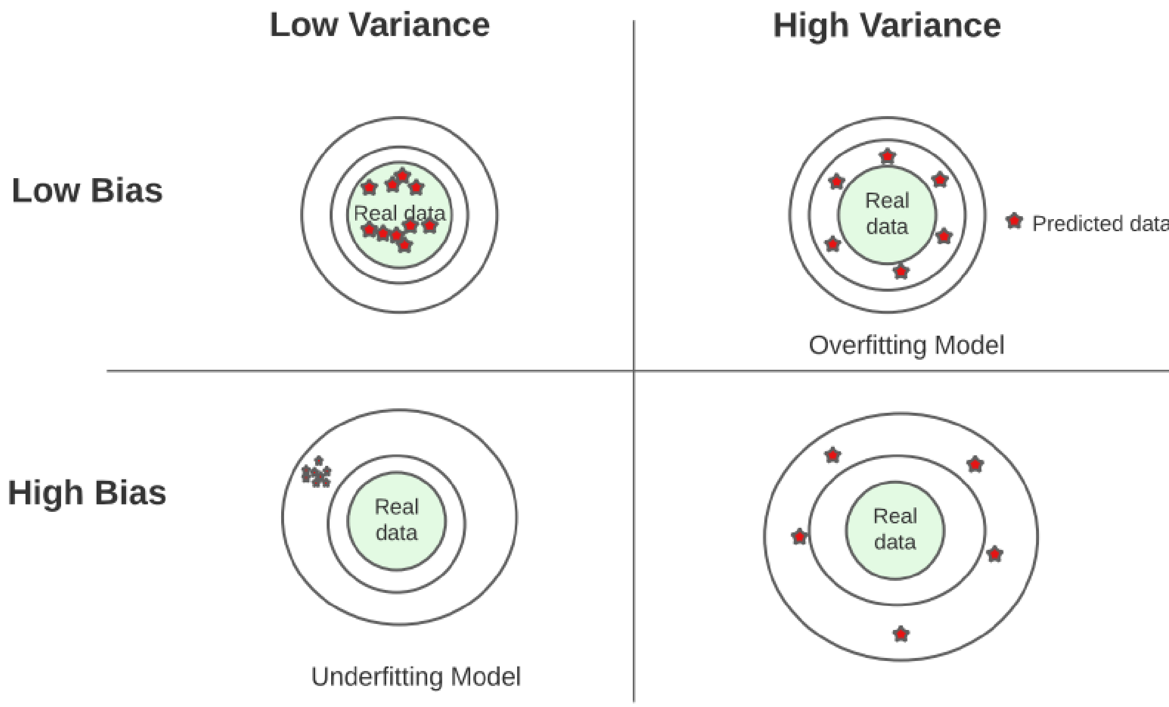
   Simple model, underfitting

2. Low Bias

   Complex model, overfitting

3. Optimal Bias-Variance

   Balanced model, good generalization



Common Model Selection Techniques

Various techniques help choose the best model. Each technique has its strengths and weaknesses, making it important to consider the specific problem and dataset.

Cross-Validation

Evaluating model performance on unseen data using multiple folds of the dataset.

Regularization

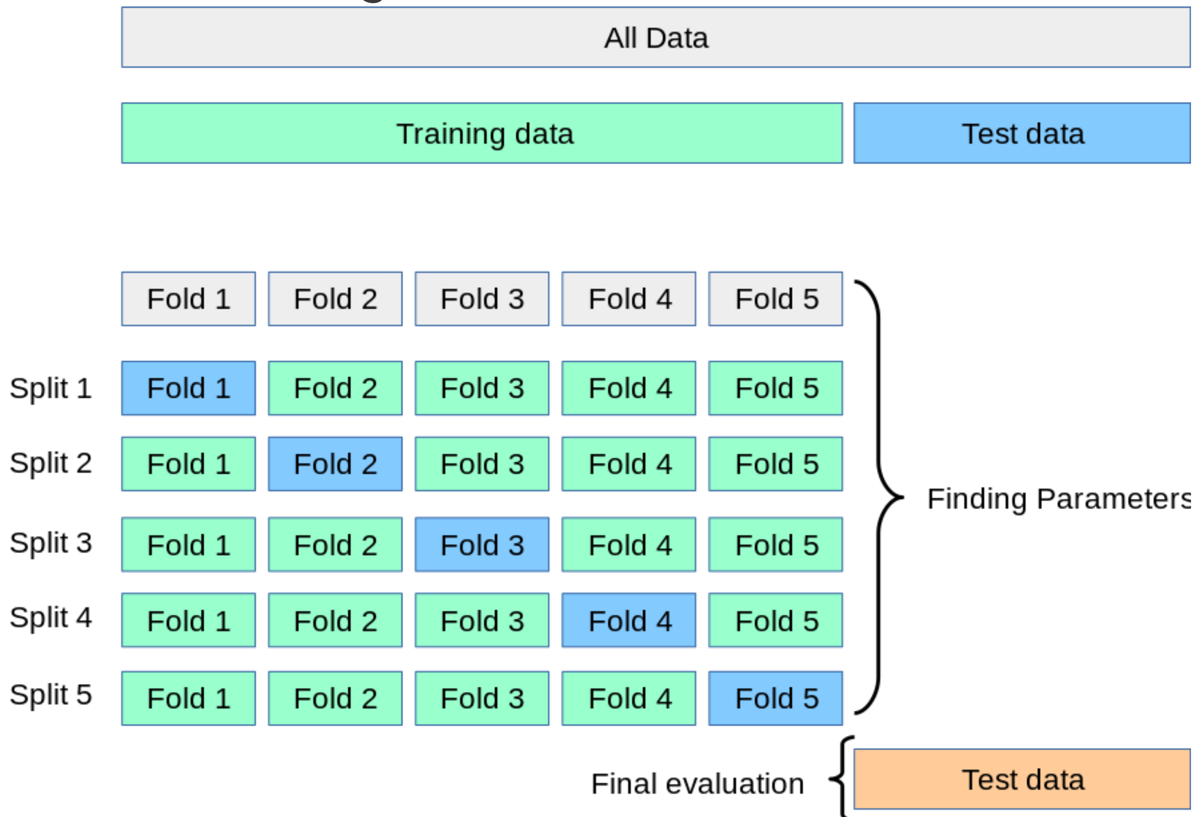
Adding constraints to the model to prevent overfitting by penalizing complex models.

Information Criteria

Using statistical measures to compare models based on their complexity and goodness of fit.

Cross-validation

- It is a technique used in machine learning to assess the performance and generalization ability of a model
- It involves partitioning the dataset into multiple subsets, training the model on some of these subsets, and evaluating it on the remaining subsets



Importance of Cross-validation

1 Overfitting Prevention

It helps prevent overfitting by evaluating the model's performance on multiple subsets of the data, ensuring that the model is not overly specialized to the training data

2 Performance Estimation

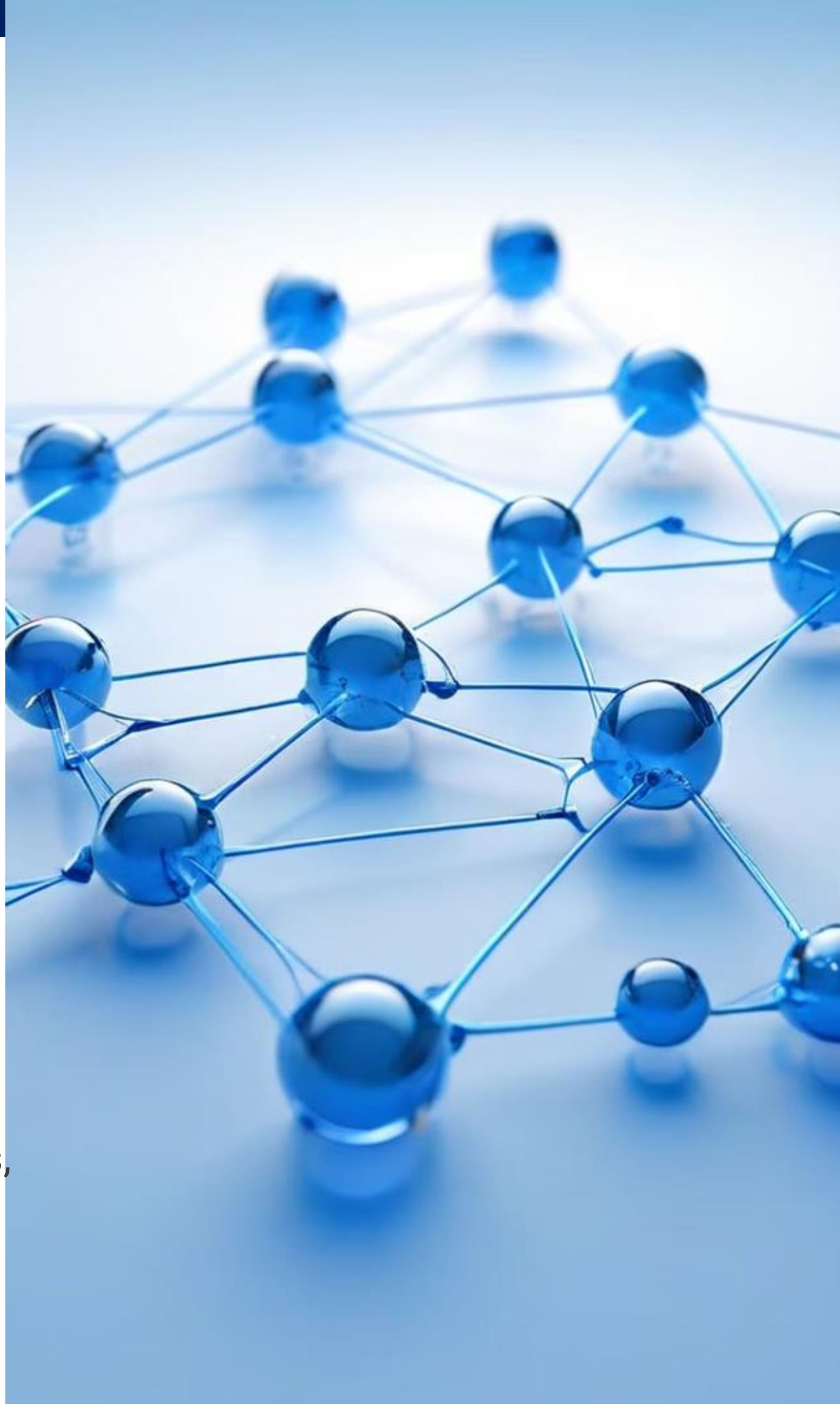
It provides a more reliable estimate of the model's true performance on unseen data, as it assesses the model's ability to generalize to different parts of the data

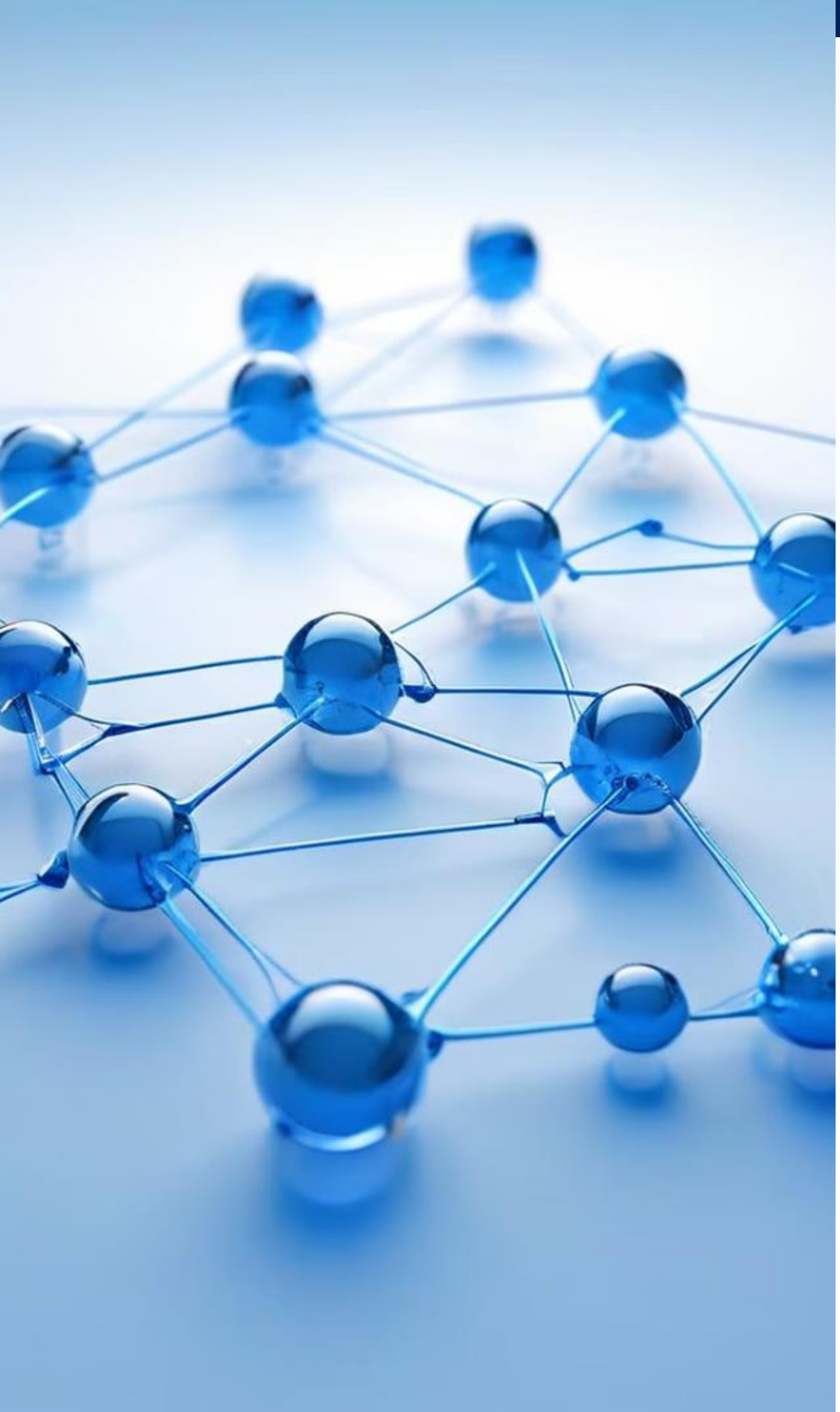
3 Model Selection

Cross-validation helps select the best model among several candidates by comparing their performance on the cross-validation folds, leading to a more robust and optimal model.

4 Hyperparameter Tuning

It aids in hyperparameter tuning by evaluating the performance of the model with different hyperparameter settings on the cross-validation folds, helping to optimize the model's performance.





Types of Cross-Validation

Train-Test Split

The simplest technique where the data is split into two parts, a training set for model development and a test set for performance evaluation.

K-Fold Cross-Validation

The data is divided into k folds, with each fold being used as a test set while the remaining k-1 folds are used for training, resulting in k different performance estimates.

Leave-One-Out Cross-Validation

A special case of k-fold cross-validation where k is equal to the number of data points, leading to n separate test sets, each containing a single data point.

Stratified Cross-Validation

Ensures that the class distribution in each fold is representative of the overall class distribution in the data, particularly important when dealing with imbalanced datasets.

K-Fold Cross Validation

1 Data Splitting

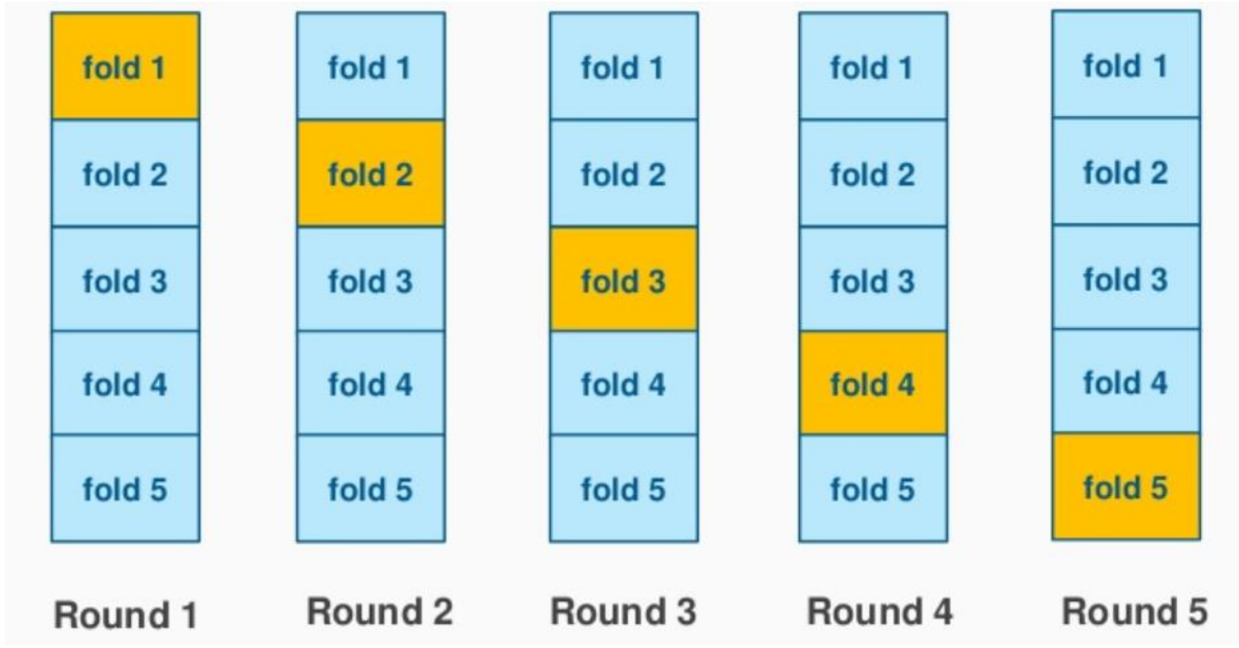
The data is divided into k equal-sized folds, ensuring that each data point belongs to only one fold

2 Model Training & Evaluation

The model is trained k times, with each fold used as the test set and the remaining k-1 folds used for training.

3 Performance Aggregation

The performance metrics from each fold are averaged to obtain a final estimate of the model's performance.



Leave-one-out Cross-validation

1

Data Splitting

The data is split into n folds, where each fold contains a single data point, resulting in n test sets, each with one data point.

2

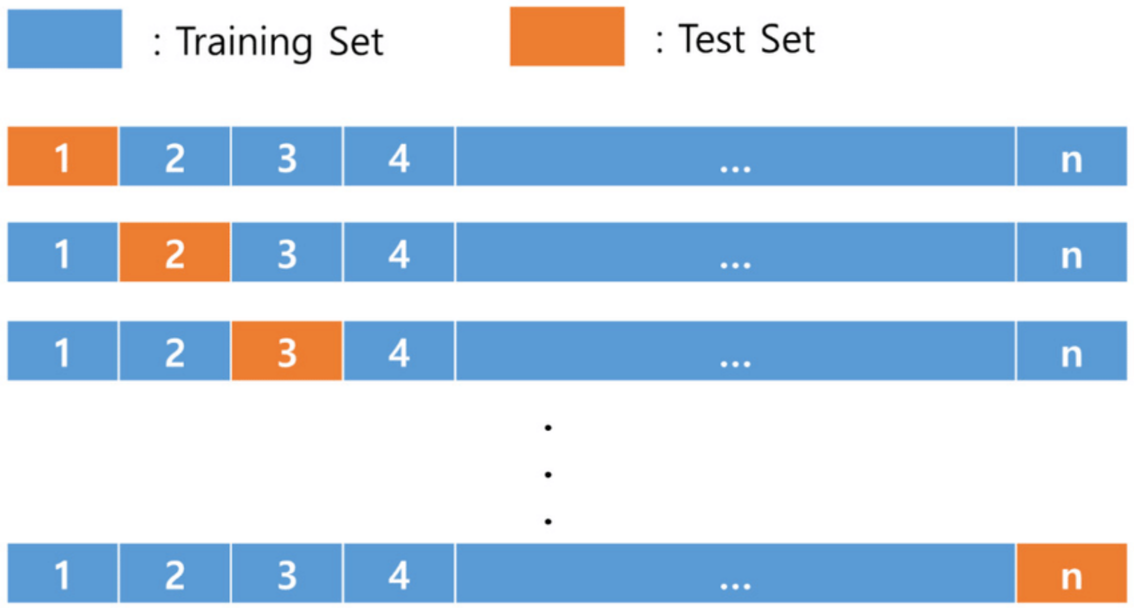
Model Training & Evaluation

The model is trained n times, each time using all but one data point for training and the remaining data point for testing.

3

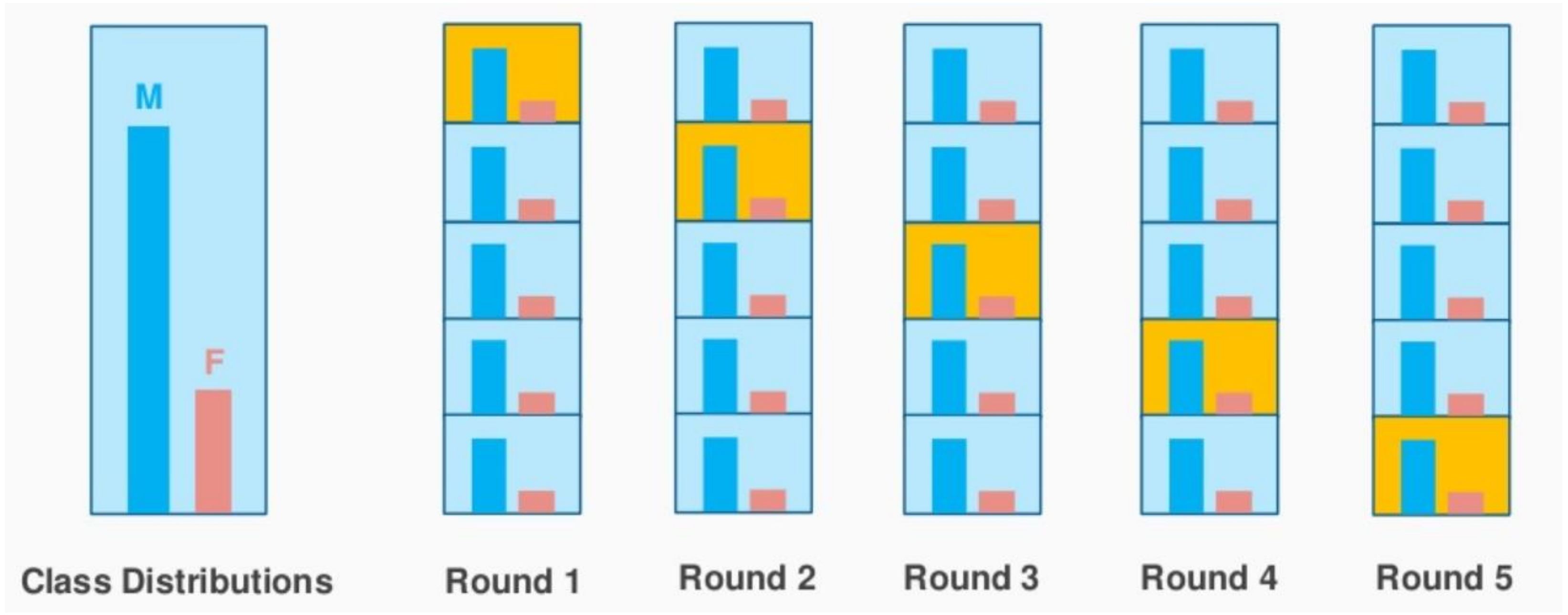
Performance Aggregation

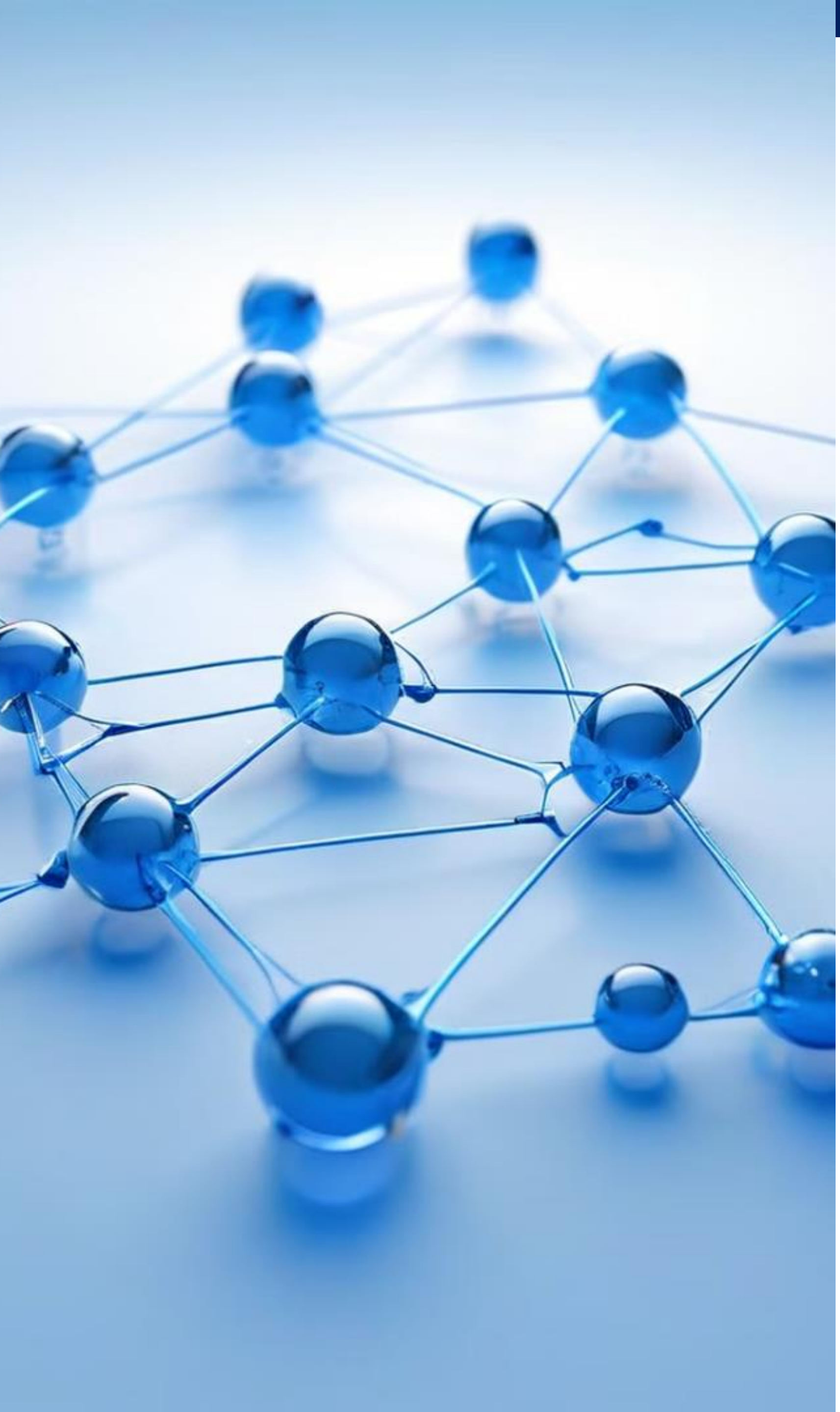
The performance metrics from each fold are averaged to get an overall estimate of the model's performance.



Stratified Cross Validation

K-Fold CV	Stratified CV
May lead to imbalanced folds, particularly with skewed class distributions	Ensures that the class distribution in each fold is representative of the overall class distribution, mitigating issues with imbalanced data
Can introduce bias in performance estimation, especially with imbalanced datasets	Provides a more accurate and unbiased evaluation of model performance, as it considers the relative proportions of different classes in the data





Advantages and Limitations of Cross-Validation

Advantages

Provides a more reliable and robust estimate of model performance, prevents overfitting, helps select the best model, and facilitates hyperparameter tuning.

Limitations

Can be computationally expensive, especially with large datasets, may not be suitable for very small datasets, and may not fully capture the real-world performance of the model.