Department of Computational and Data Sciences

INDIAN INSTITUTE OF SCIENCE

भारतीय विज्ञान संस्थान

AI: Module 01 Week 01

Deepak Subramani

Assistant Professor

Dept. of Computational and Data Science

Indian Institute of Science Bengaluru

Week 01

- Part 1
  - Decision Tree – Structure, Prediction, Attributes
  - Decision Boundary – Fundamental Concept in AI for Classification
  - Cart Training Algorithm
- Part 2
  - Metrics of performance – Regression and Classification
  - Development-Testing Paradigm
  - Overfit vs Underfit – Regularization to prevent overfit
  - Hyperparameter Selection through k-fold CV

ML: Mental Model

Data that can be collected

$$\begin{tikzcd} X \arrow[r] & \boxed{\text{Real World}} \arrow[r] & y \end{tikzcd}$$

Quantity that must be predicted to make money

Data that can be collected

$$\begin{tikzcd} X \arrow[r] & \boxed{h_\theta(X;\theta)} \arrow[r] & \hat{y} \end{tikzcd}$$

Machine's Prediction

The Machine Learning Workflow

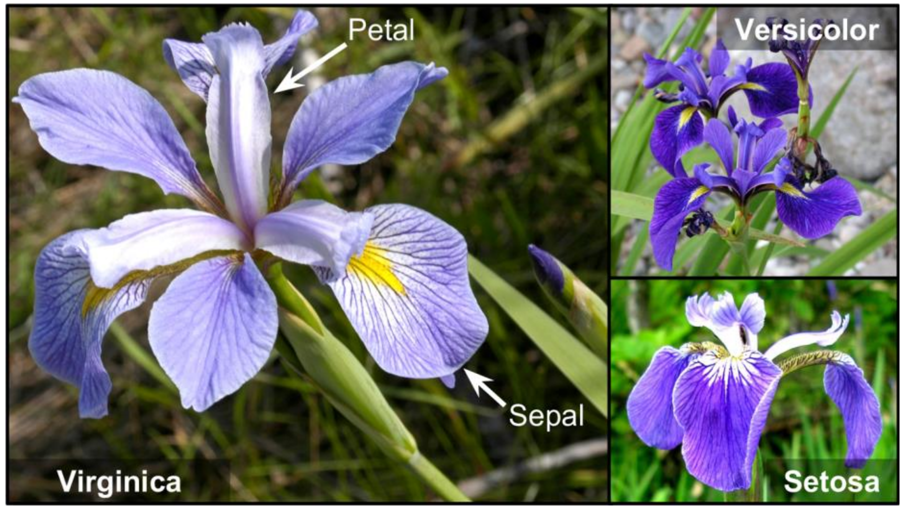
1. Frame the ML problem by looking at the business need
   a. Identify subproblems (One/more of the 5 tasks a computer can do)
   b. Establish a current baseline (What is currently done?)
   c. Define success
2. Gather the data and do Data Munging/Wrangling + Baselines
   a. Explore the data
   b. Clean data and prepare for the downstream ML models
   c. Establish a data, domain and SoTA baseline
3. Explore different models, improve them through Cross Validation and perhaps new model design
4. Form an ensemble of multiple models and solutions
5. Present your solution
   a. Say a story with the data
6. Deploy

Decision Trees

- Decision Trees are data driven models for classification and regression
- They are a versatile ML Algorithm capable of fitting complex data
- They are trained by a greedy optimization algorithm called CART
- Plan of action:
  1. See commands for training DT in sklearn
  2. Predict with DT and see how they make a decision
  3. Understand the math of the optimization algorithm
  4. Discuss pros and cons

Iris Dataset

Sepal Length	Sepal Width	Petal Length	Petal Width	Species



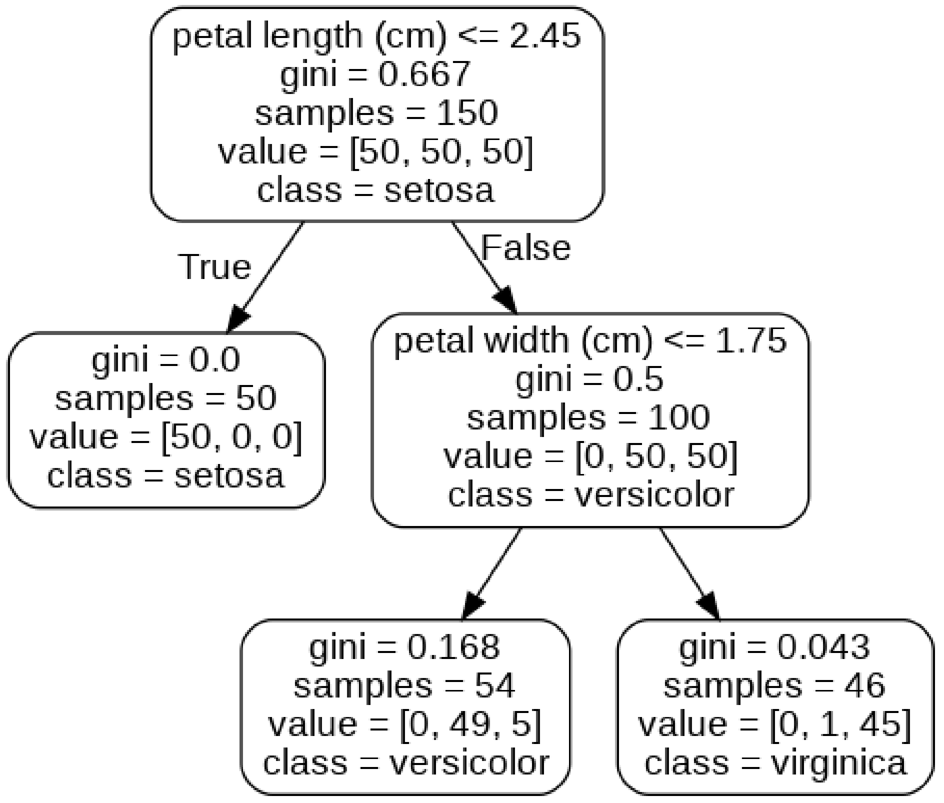
- The problem we did so far is a classification problem
- $Species = h_\theta(SL, SW, PL, PW; \theta)$

Training a DT in sklearn

- from sklearn.datasets import load_iris
- from sklearn.tree import DecisionTreeClassifier
- iris = load_iris()
- X = iris.data[:,2:] #Make 2 features in the data
- y = iris.target
- tree_clf = DecisionTreeClassifier(max_depth=2)
- tree_clf.fit(X,y)

Making Predictions with DT

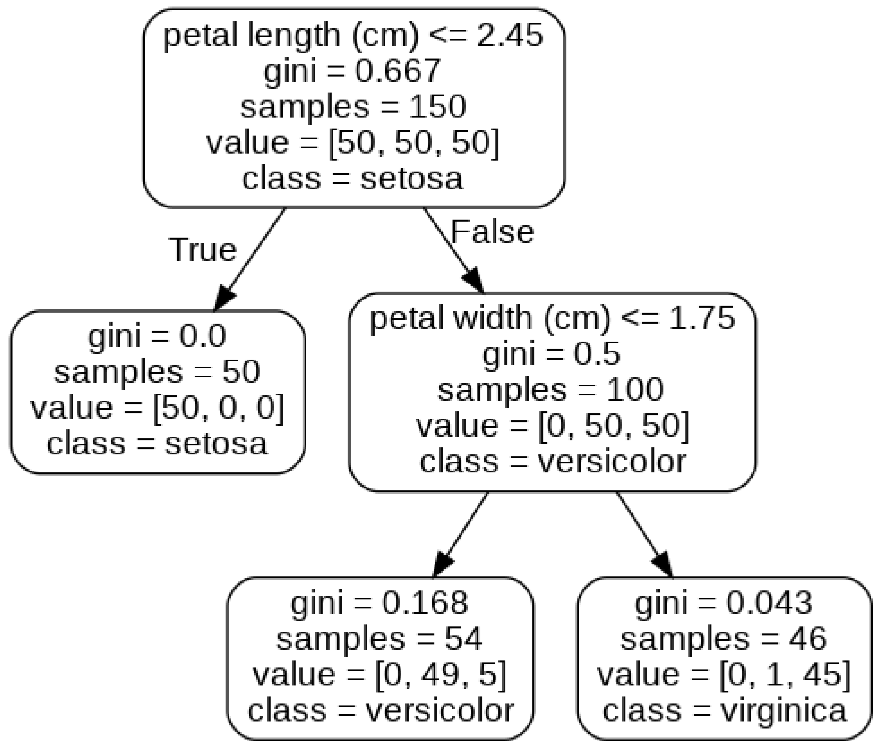
- How does DT classify a new data point?
- Start from the top, and move down asking the question at each node and following the answer
- Is petal length (cm) <= 2.45?
  - True – move left; False – move right
- Keep moving until you reach a leaf node
  - Leaf node – no children
- The class of the leaf node a datapoint ends up in is its class!
- That simple!

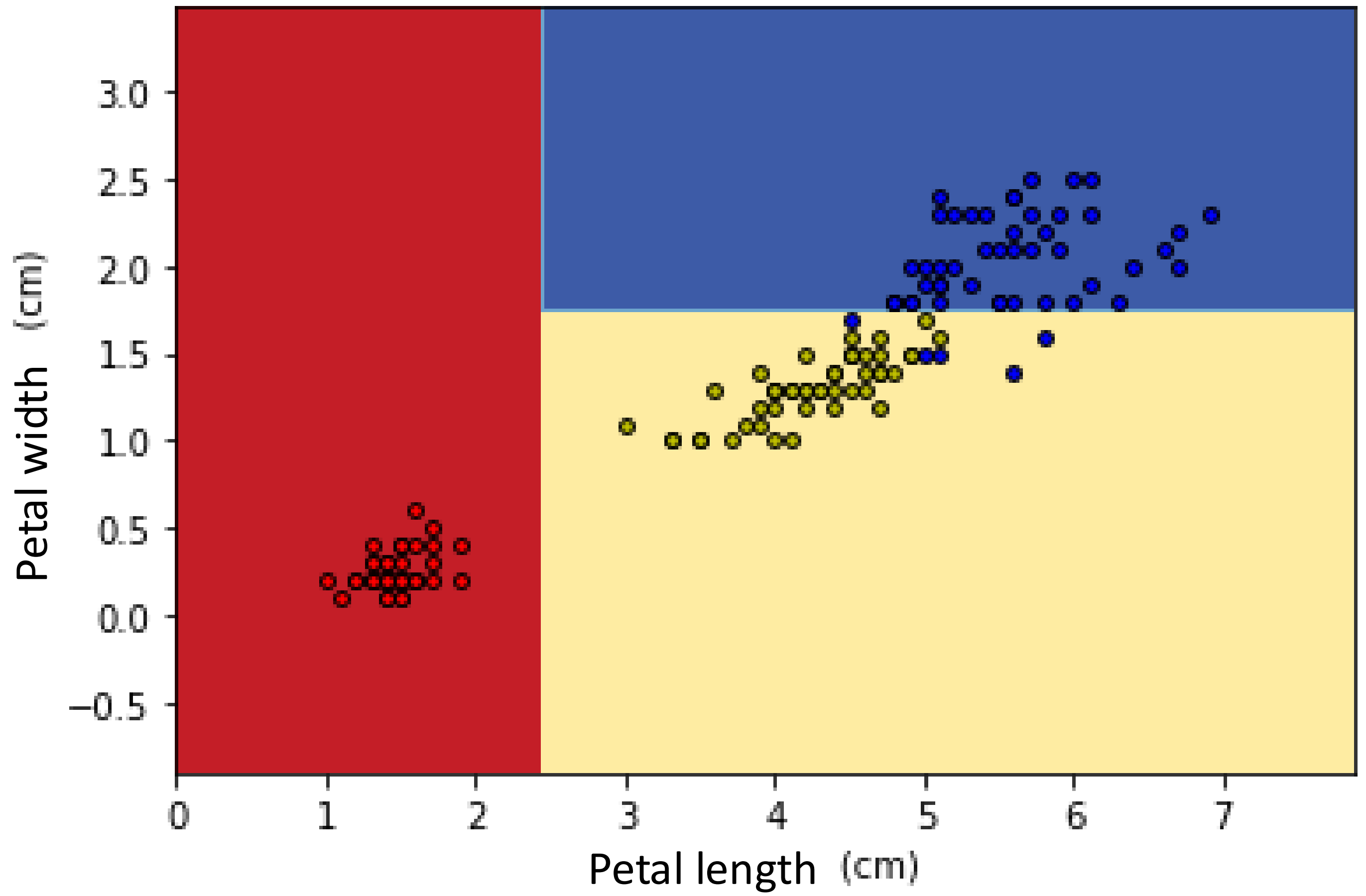


Understanding DT Attributes

- Gini attribute:
  - Measures impurity at a node
  - $G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$
  - $p_{i,k}$ is the fraction of kth class in ith node. n=total classes
- Samples attribute:
  - Number of training instances for which this question was applied
- Value attribute:
  - How many training instances of each class this node applies to
- Class attribute:
  - The label which will be applied to the instance if we stop there
- Probability of a class at a particular node
  - Value/samples

Decision Boundaries





CART Training Algorithm

1. At every node, split the training set into two subsets based on one single feature x_k and a threshold t_k on it
2. The pair (x_k, t_k) is chosen by what results in the purest subset measured by Gini Coefficient or Entropy

$$J(x_k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

$G_{l/r}$ is the impurity and $m_{l/r}$ is the number of instances

3. Continue the same at the two child nodes
4. Stop when no further split reduces impurity or maximum depth is reached

Example of a node split

- PL<=2.45
  - True (left) – Samples 50 Value = [50,0,0]
  - False (right) – Samples 100 Value = [0,50,50]
  - G_left =
  - G_right =
  - J =
- PL<=3.5
  - True (left) – Samples 55 Value = [50, 5, 0]
  - False (right) – Samples 95 Value = [0, 45, 50]
  - G_left =
  - G_right =
  - J =

What is Gini Impurity?

- The Gini impurity measures the probability that a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset.
- Worked out example
  - Let us consider class A, B and C.
  - we randomly pick a point and label it according to the distribution.
  - Let the label distribution be (0.3,0.5,0.2).
  - probability of picking up a data point randomly with label A is 0.3. The probability of labeling it B or C is 1-0.3=0.7.
  - probability of picking up a data point randomly with label B is 0.5. The probability of labeling it A or C is 1-0.5=0.5.
  - probability of picking up a data point randomly with label C is 0.2. The probability of labeling it A or B is 1-0.2=0.8.
- The total probabilty of misclassification of a randomly selected data point is
- $p(mis) = 0.3 * (1 - 0.3) + 0.5 * (1 - 0.5) + 0.2 * (1 - 0.2) = 0.3 + 0.5 + 0.2 - (0.3\text{^}2 + 0.5\text{^}2 + 0.2\text{^}2) = 1 - (0.3\text{^}2 + 0.5\text{^}2 + 0.2\text{^}2) = 1 - \Sigma_{k=1}^{K} p_k^2 = 1 - 0.38 = 0.62$
- It ranges from 0 to 0.5 (for binary classification) or 0 to 1 - 1/J (for multi-class problems).
- 0 represents a pure node (all elements belong to the same class). Higher values indicate more mixed classes.

Week 01

- Part 1
  - Decision Tree – Structure, Prediction, Attributes
  - Decision Boundary – Fundamental Concept in AI for Classification
  - Cart Training Algorithm
  - Collection of Decision Trees – Random Forest
- Part 2
  - Metrics of performance – Regression and Classification
  - Development-Testing Paradigm
  - Overfit vs Underfit – Regularization to prevent overfit
  - Hyperparameter Selection through k-fold CV

Evaluation Metrics – Classification Confusion Matrix

	GT+	GT-
CLS+	True Positive	False Positive
CLS-	False Negative	True Negative

Example:
+ : Versicolor
- : Not Versicolor

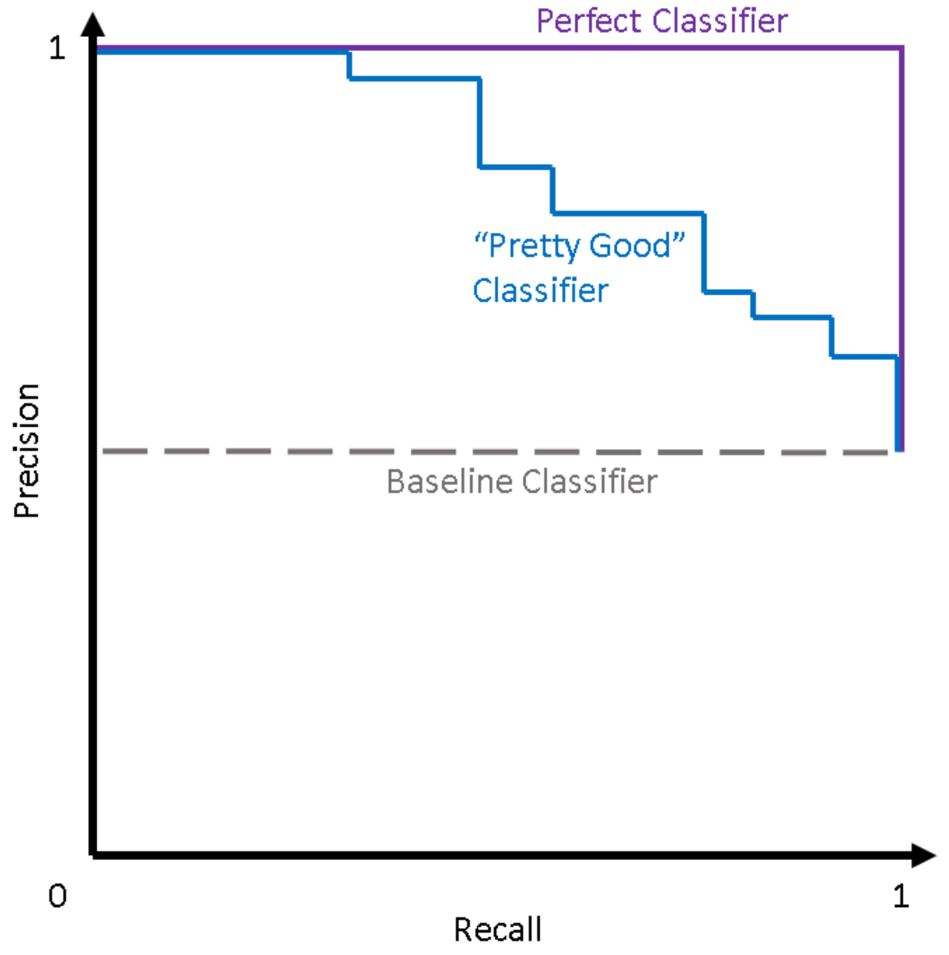
- Recall = TP/(TP+FN) – Also called Sensitivity in Statistics
- Precision = TP/(TP+FP)
- F1 Score = Harmonic mean of Recall and Precision

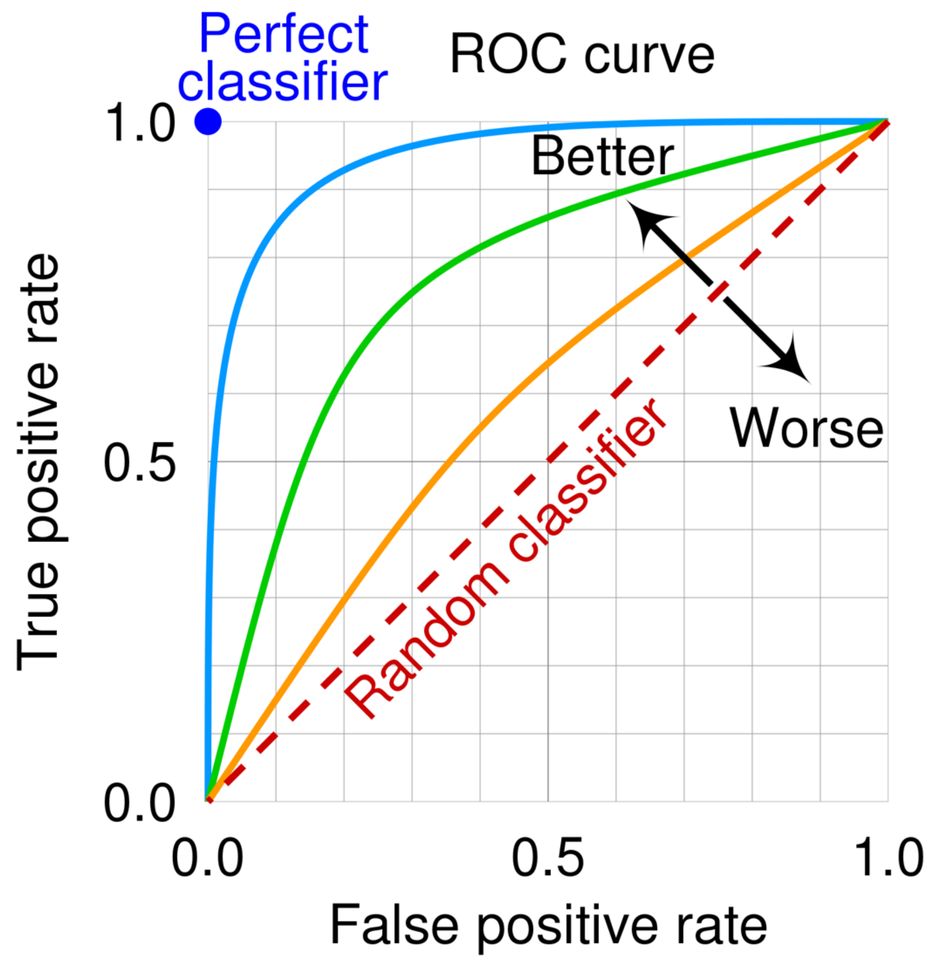
- False Negative Rate = FN/(TP+FN)

- Specificity = TN/(FP+TN)
- False Positive Rate = FP/(FP+TN)

TP=	FP=
FN=	TN=

PR Curve and ROC AUC

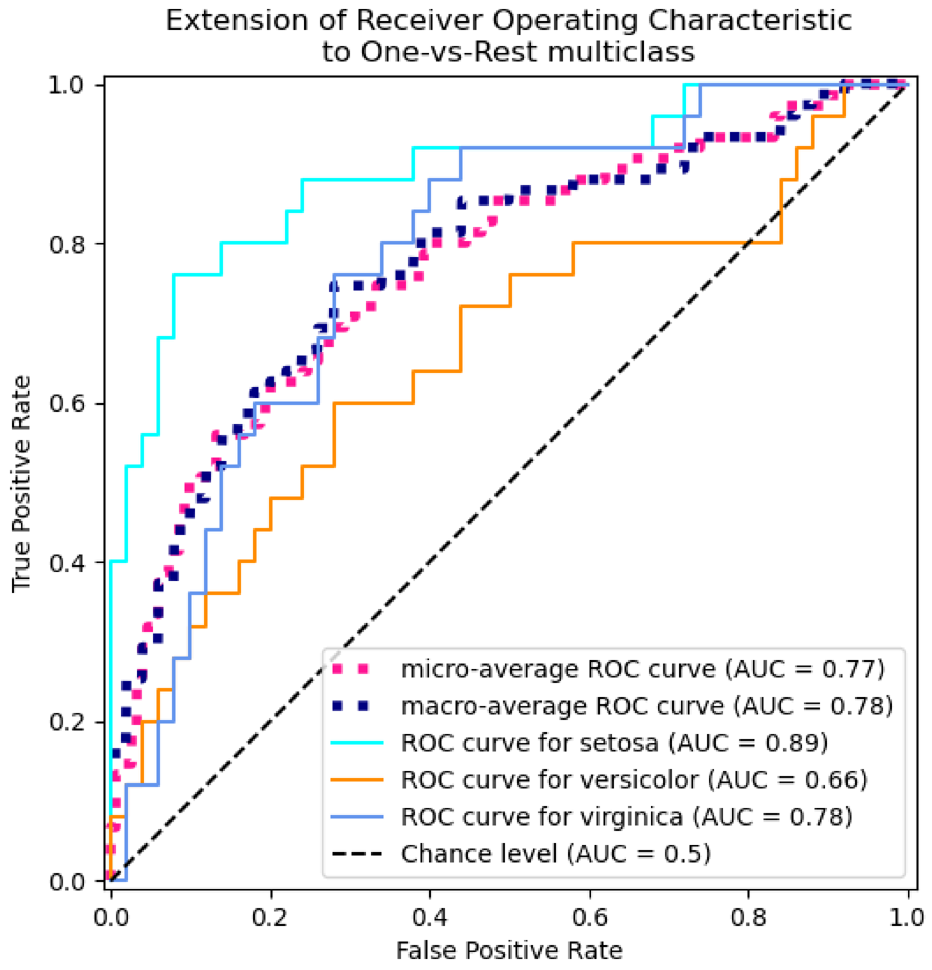




Baseline – predict all instances as +ve class

ROC AUC for Multi Class

- Micro-average
  - Ignores class distinctions and gives equal weight to each instance. It's a global strategy that calculates the performance metric based on the total counts of true positives, false positives, and false negatives across all classes.
- Macro-average
  - Gives equal weight to each class, regardless of the number of instances. It calculates each class's performance metric (e.g., precision, recall) and then takes the arithmetic mean across all classes.

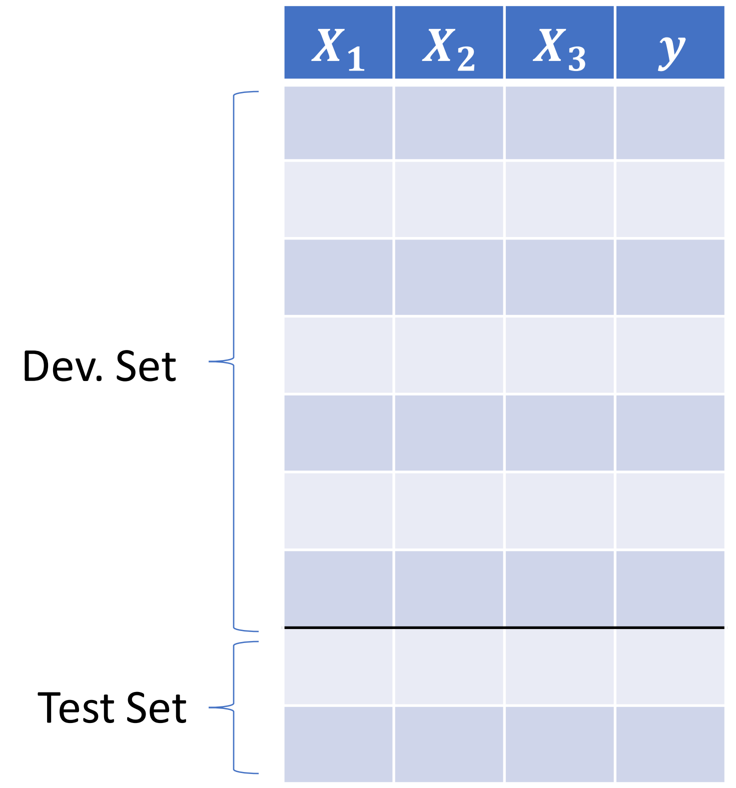


Evaluation Metrics – Regression

- Root Mean Square Error
- Mean Absolute Error
- Relative Errors
- R2 = 1 – MSE/Variance

Development-Testing Paradigm

- Consider a data set with m rows and n columns
- Development Set
  - Used to train a ML model
- Training a model involves
  - Finding parameters or growing trees
  - Uses data and an optimization algorithm working on some loss
- Development involves training and hyper-parameter tuning
  - K-Fold Cross Validation is used
- Testing involves using a developed model on totally unseen data and evaluating an expected real world performance

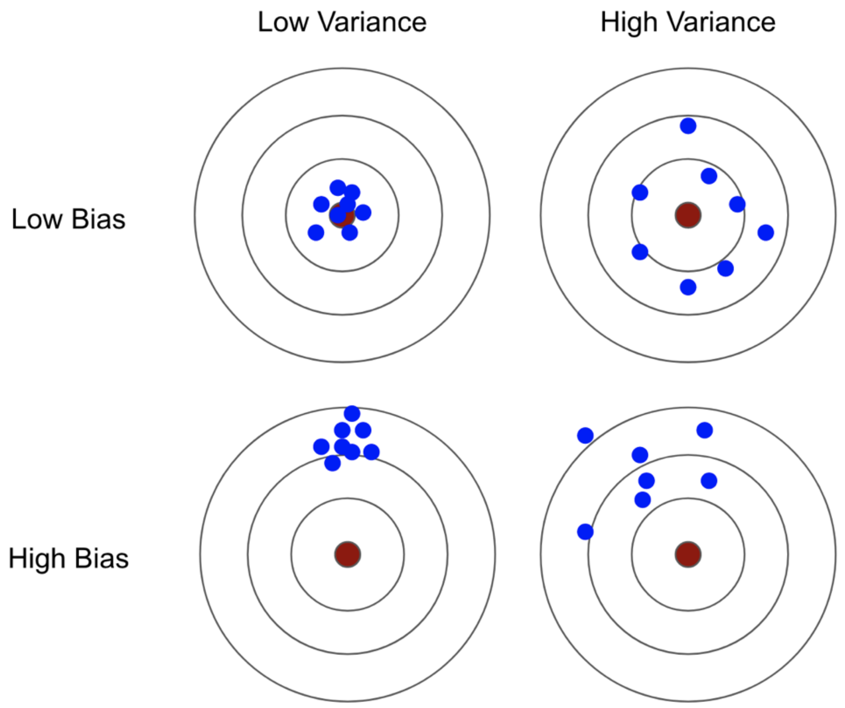


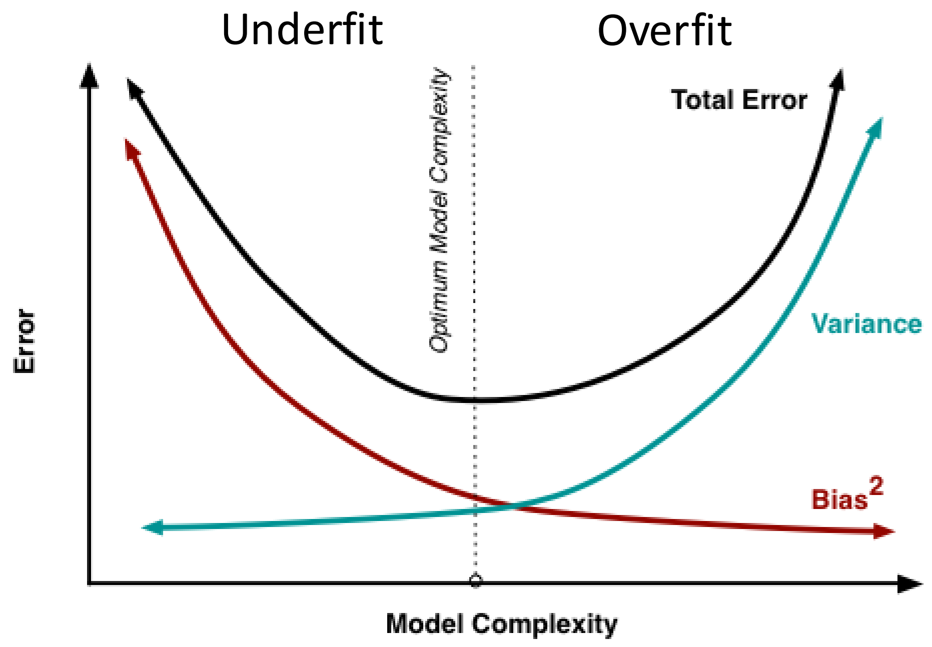
Bias-Variance Tradeoff

- Error = Bias + Variance + Irreducible Error
- Bias: Error from erroneous assumptions in the model
  - High bias -> underfitting
  - Model has limited flexibility to learn
  - Analogy: Overly simplistic assumptions about people make you a biased person
- Variance: Error from sensitivity of model to small perturbations in training data
  - High variance -> overfitting
  - Model has too much flexibility to learn
  - If you have a lot of data and over complex model, you can make each parameter of the model “by-heart” one data point
  - When a new data point comes, then the model will change wildly to accommodate the new point

Bias/Variance Equation

- $Error \;= E[(y-\hat{y})^2]$ –> RMSE
- $Error = (E[\hat{y}]-y)^2 + E[(\hat{y}-E[\hat{y}])^2] + \sigma_e^2$
- Error = Bias + Variance + Irreducible Error

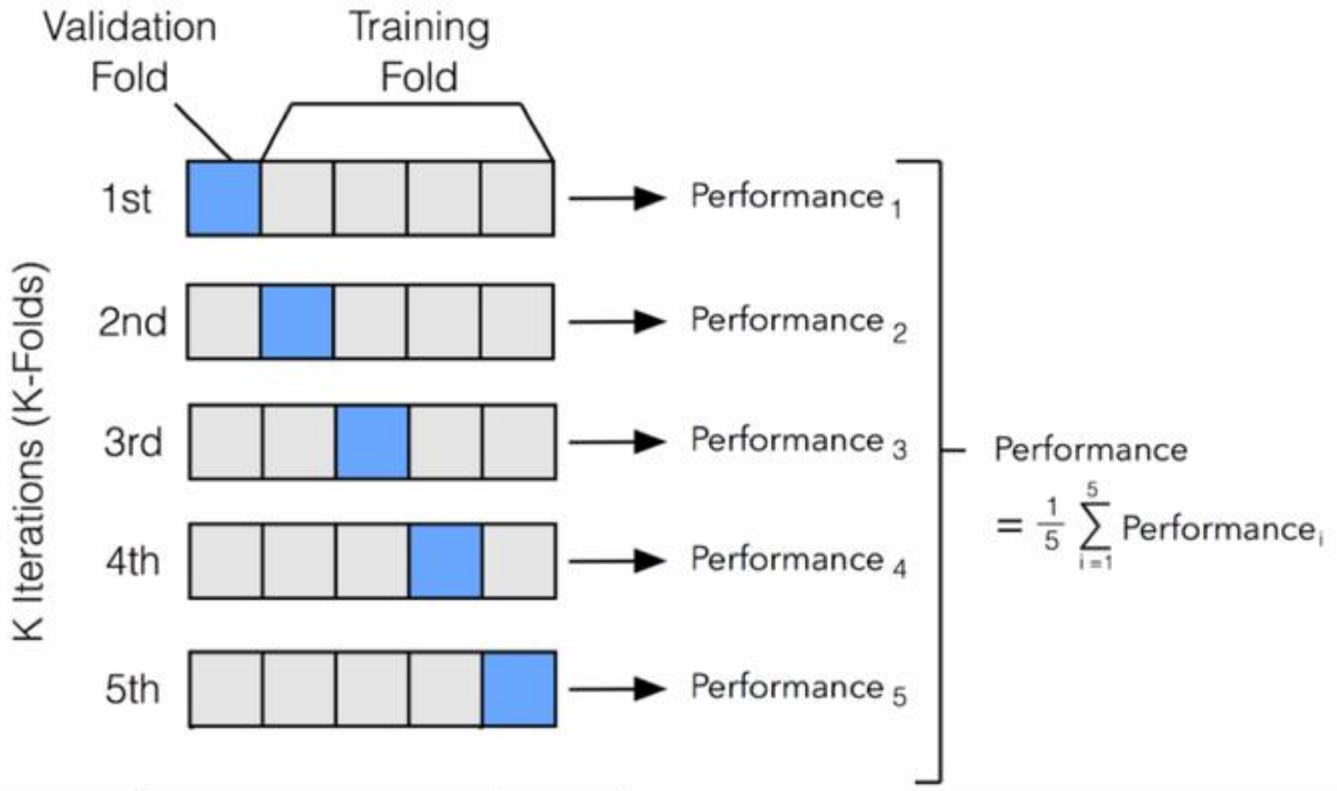




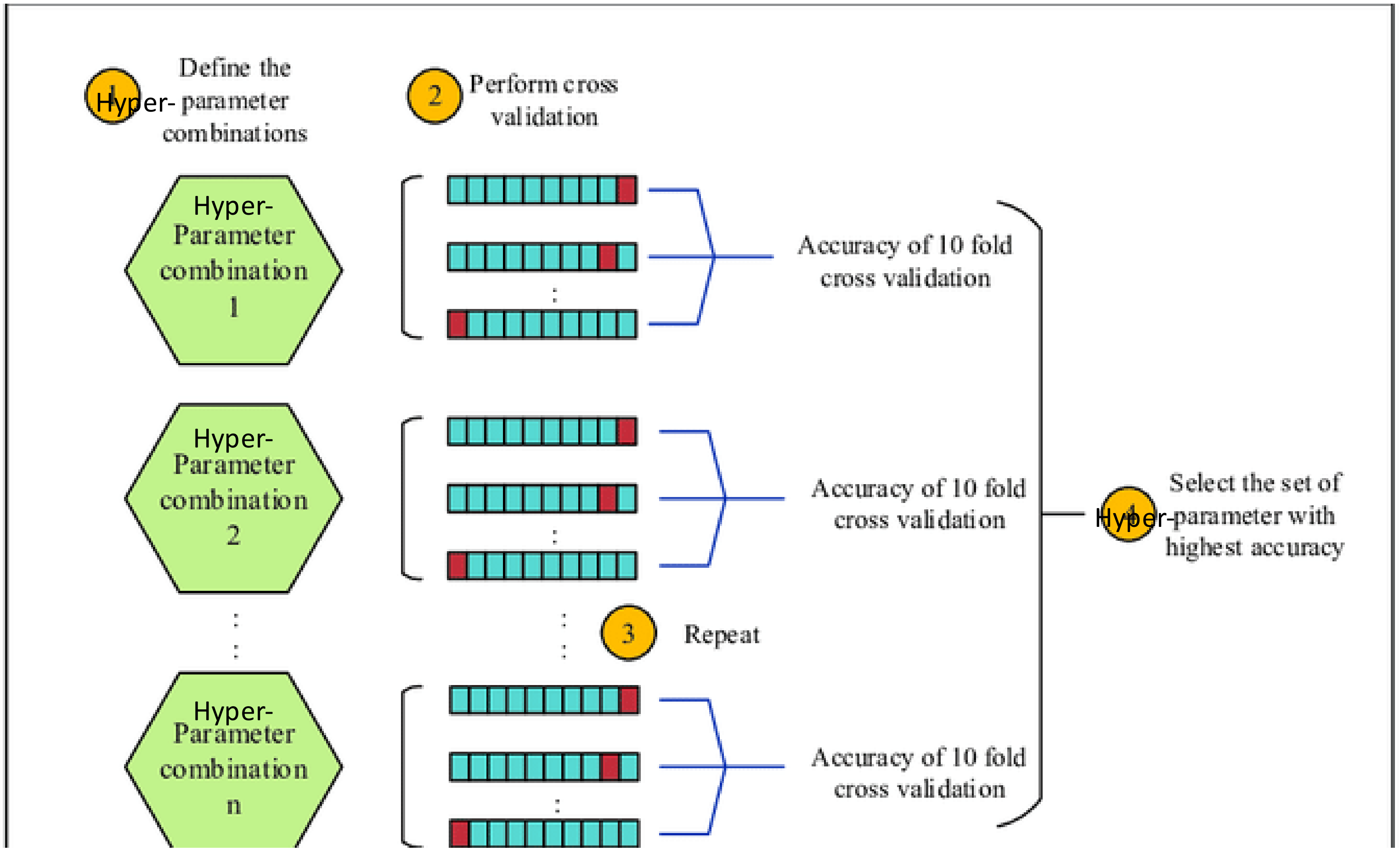
Regularization

- DT are what is called as non-parametric models
    - They don't have a pre-defined number of parameters
    - On the other hand, linear models are parametric, with predefined number of parameters
- Thus DT can overfit any complex training data
- To avoid overfitting, we restrict a DT's degrees of freedom
- Use the following hyperparameters to regularize and avoid overfitting:
    - max_depth: the maximum number of levels
    - min_samples_split: the minimum number of samples a node must have before it can be split,
    - min_samples_leaf: the minimum number of samples a leaf node must have,
    - min_weight_fraction_leaf: same as min_samples_leaf but expressed as a fraction of the total number of weighted instances,
    - max_leaf_nodes: the maximum number of leaf nodes, and
    - max_features: the maximum number of features that are evaluated for splitting at each node.
    - Increasing min_ hyperparameters or reducing max_ hyperparameters will regularize the model.
- Can also build a tree without restrictions and prune

K-Fold Cross Validation



Hyper-parameter Tuning


1 Define the Hyper- parameter combinations
2 Perform cross validation
Hyper- Parameter combination 1
Hyper- Parameter combination 2
Hyper- Parameter combination n
Accuracy of 10 fold cross validation
Accuracy of 10 fold cross validation
Accuracy of 10 fold cross validation
3 Repeat
4 Select the set of Hyper-parameter with highest accuracy

Audience Poll

1. Ensemble Learning can be used only with Decision Trees
   - True, False
2. Feature importance can be calculated ONLY for tree-based models
   - True, False
3. Permutation feature importance measures the importance of each feature by using an adversarial approach of permuting that feature and evaluating the model
   - True, False