Decision Trees II and Bias/Variance and Cross-Validation

Nipun Batra and teaching staff

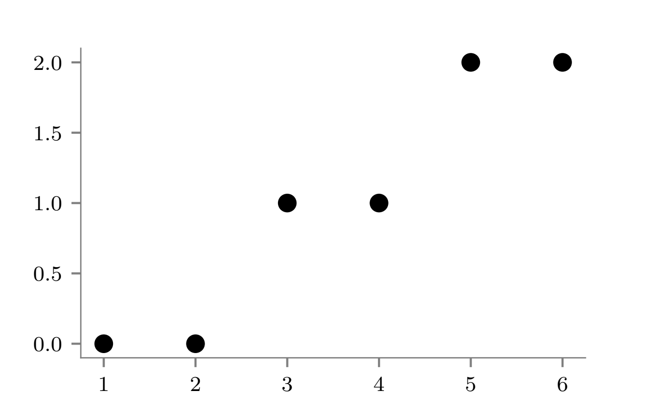
December 23, 2023

IIT Gandhinagar

Real Input Real Output

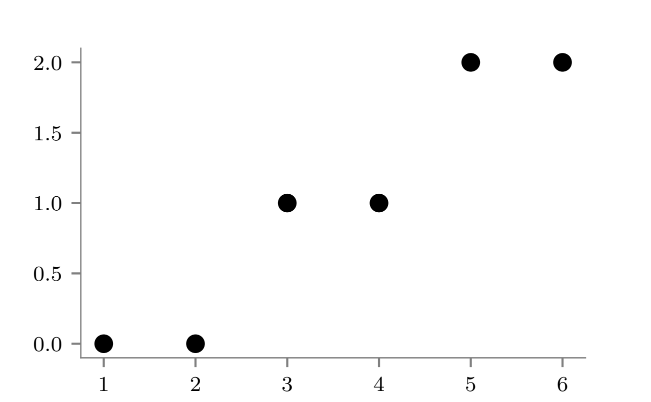
Example 1

Let us consider the dataset given below



Example 1

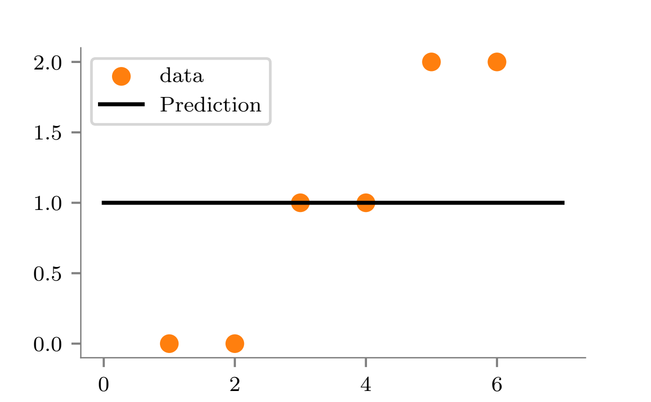
What would be the prediction for decision tree with depth 0?



Example 1

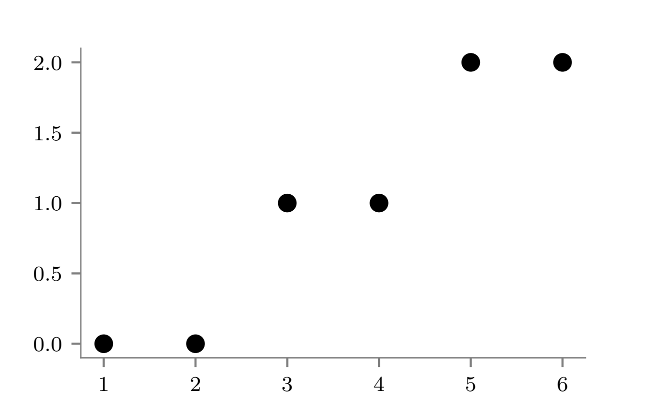
Prediction for decision tree with depth 0.

Horizontal dashed line shows the predicted Y value. It is the average of Y values of all datapoints.



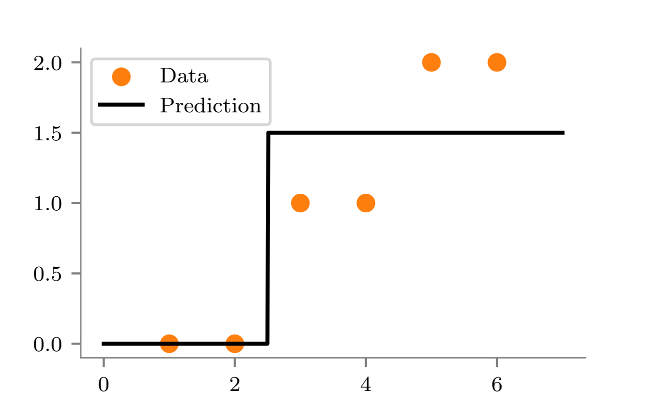
Example 1

What would be the decision tree with depth 1?

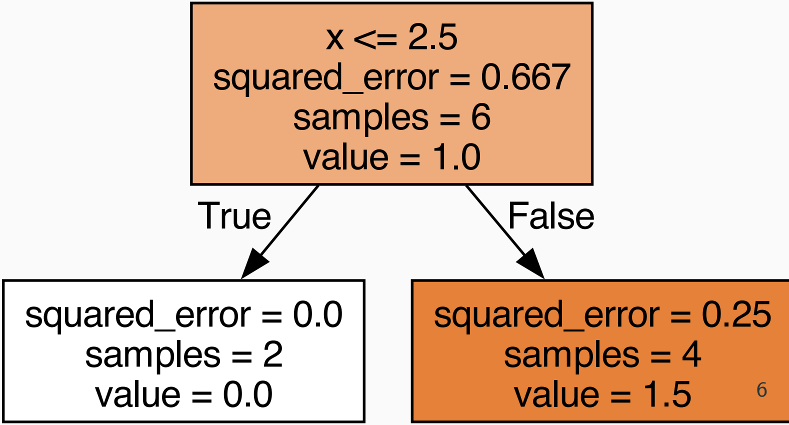


Example 1

Decision tree with depth 1

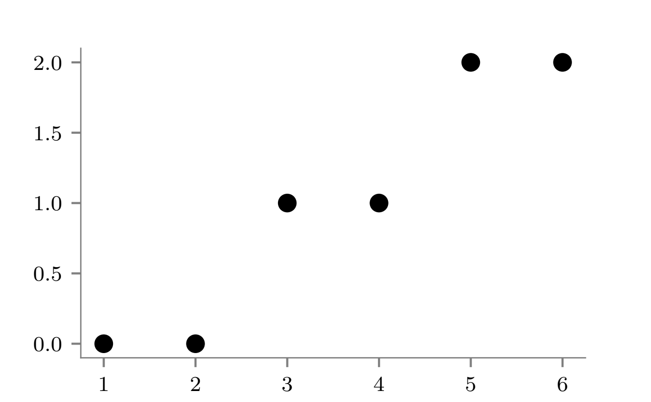


The Decision Boundary



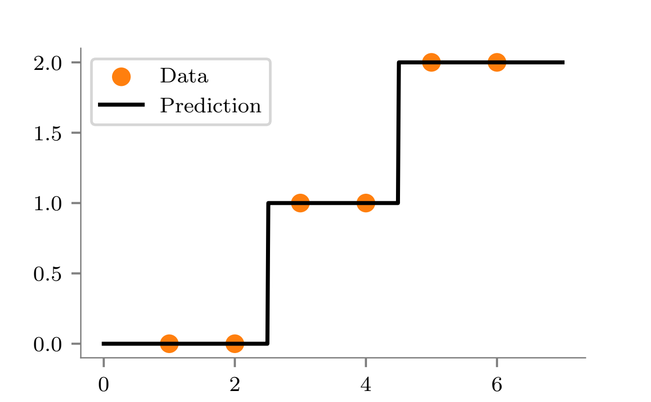
Example 1

What would be the decision tree with depth 2 ?



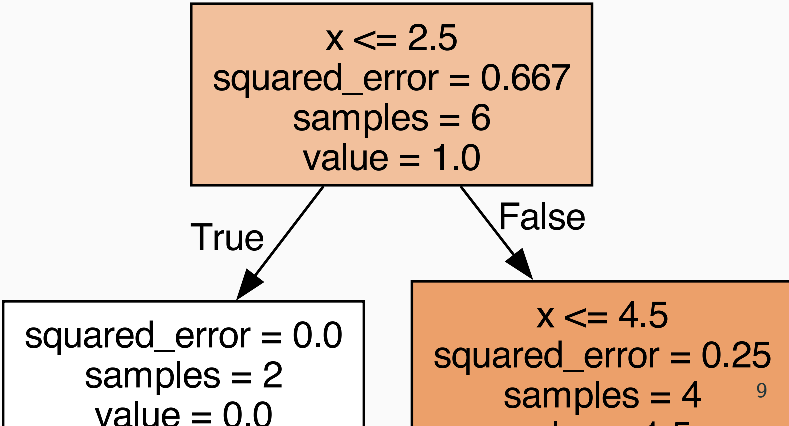
Example 1

Decision tree with depth 1



Example 1

The Decision Boundary



Objective function

Here, Feature is denoted by X and Label by Y.

Let the "decision boundary" or "split" be at $X = S$.

Let the region $X < S$, be region R_1.

Let the region $X > S$, be region R_2.

Objective function

Here, Feature is denoted by X and Label by Y.
Let the "decision boundary" or "split" be at $\mathrm{X} = \mathrm{S}$.
Let the region $\mathrm{X} < \mathrm{S}$, be region R_1.
Let the region $\mathrm{X} > \mathrm{S}$, be region R_2.

Then, let
$C_1 = \text{Mean }(Y_i|X_i \in R_1)$
$C_2 = \text{Mean }(Y_i|X_i \in R_2)$

Objective function

Here, Feature is denoted by X and Label by Y.

Let the "decision boundary" or "split" be at $\mathrm{X} = \mathrm{S}$.

Let the region $\mathrm{X} < \mathrm{S}$, be region R_1.

Let the region $\mathrm{X} > \mathrm{S}$, be region R_2.

Then, let

$C_1 = \text{Mean } (Y_i | X_i \in R_1)$

$C_2 = \text{Mean } (Y_i | X_i \in R_2)$

$\text{Loss} = \sum_i ((Y_i - C_1 | X_i \in R_1)^2 + (Y_i - C_2 | X_i \in R_2)^2)$

Objective function

Here, Feature is denoted by X and Label by Y.

Let the "decision boundary" or "split" be at X = S.

Let the region X < S, be region R_1.

Let the region X > S, be region R_2.

Then, let

$C_1 = \text{Mean } (Y_i | X_i \in R_1)$

$C_2 = \text{Mean } (Y_i | X_i \in R_2)$

$\text{Loss} = \sum_i ((Y_i - C_1 | X_i \in R_1)^2 + (Y_i - C_2 | X_i \in R_2)^2)$

Our objective is to minimize the loss and find

$min_S \sum_i \left((Y_i - C_1 | X_i \in R_1)^2 + (Y_i - C_2 | X_i \in R_2)^2\right)$

How to find optimal split “S”?

How to find optimal split “S”?

1. Sort all datapoints (X,Y) in increasing order of X.

How to find optimal split “S”?

1. Sort all datapoints (X,Y) in increasing order of X.

2. Evaluate the loss function for all

$$S = \frac{X_i + X_{i+1}}{2}$$

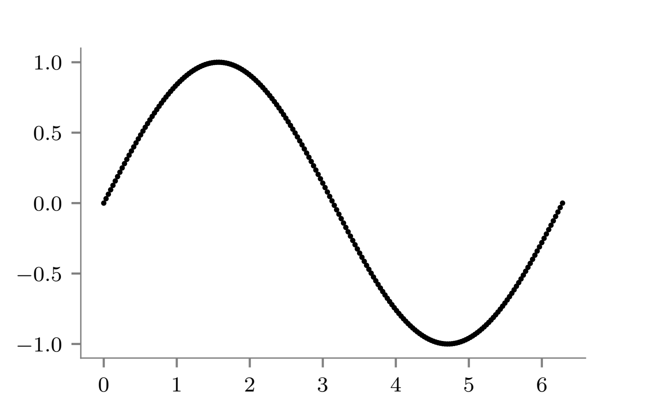
   and the select the S with minimum loss.

A Question!

Draw a regression tree for Y = sin(X), $0 \leq X \leq 2\pi$

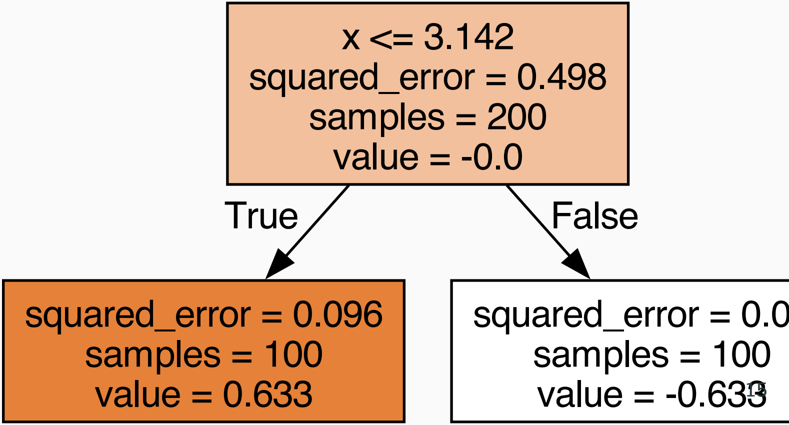
A Question!

Dataset of Y = sin(X), $0 \leq X \leq 7$ with 10,000 points



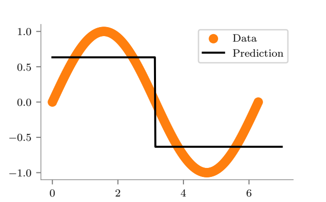
A Question!

Regression tree of depth 1



A Question!

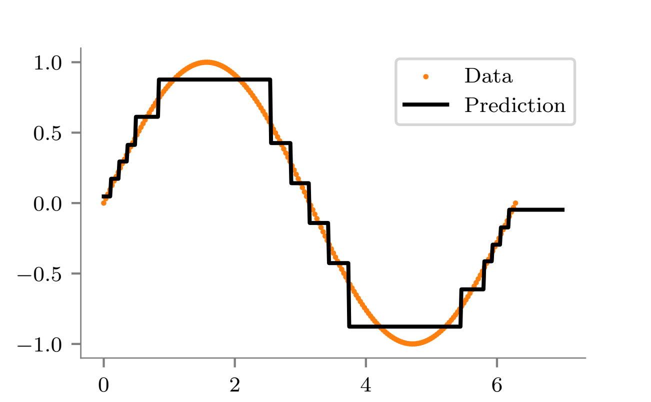
Decision Boundary



A Question!

Regression tree with no depth limit is too big to fit in a slide.

It has of depth 4. The decision boundaries are in figure below.

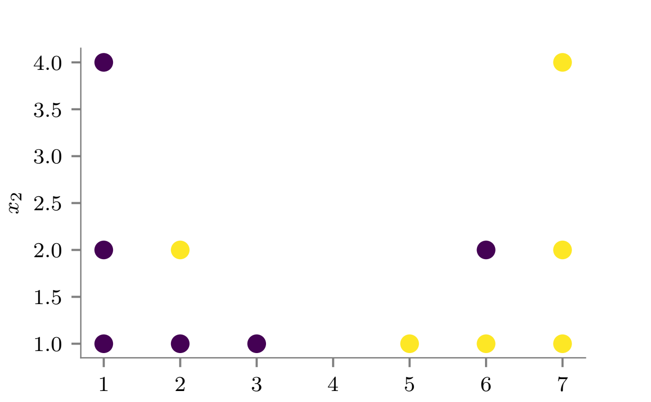


Summary

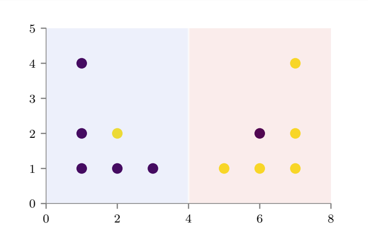
- Interpretability an important goal
- Decision trees: well known interpretable models
- Learning optimal tree is hard
- Greedy approach:
- Recursively split to maximize "performance gain"
- Issues:
  - Can overfit easily!
  - Empirically not as powerful as other methods

A Question!

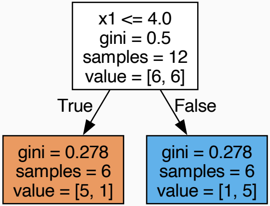
What would be the decision boundary of a decision tree classifier?



Decision Boundary for a tree with depth 1

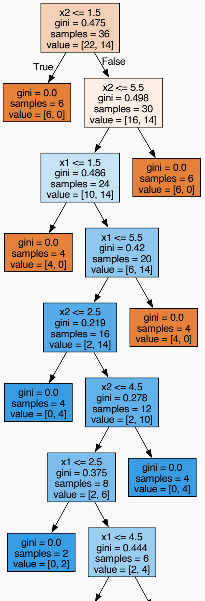


(a) Decision Boundary



(b) Decision Tree

Decision Boundary for a tree with no depth limit



Are deeper trees always better?

As we saw, deeper trees learn more complex decision boundaries.

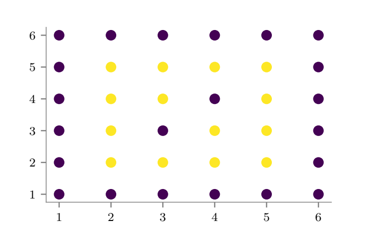
Are deeper trees always better?

As we saw, deeper trees learn more complex decision boundaries.

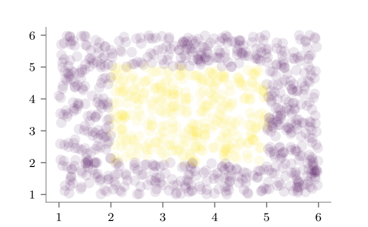
But, sometimes this can lead to *poor generalization*

An example

Consider the dataset below



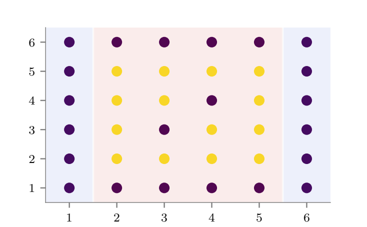
(e) Train Set



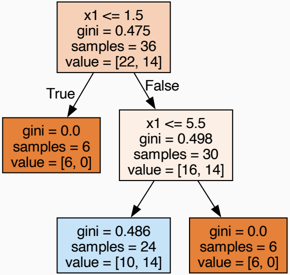
(f) Test Set

Underfitting

Underfitting is also known as *high bias*, since it has a very biased incorrect assumption.



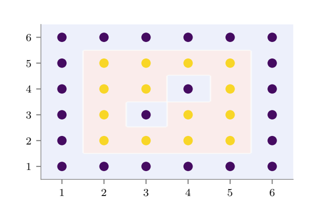
(g) Decision Boundary



(h) Decision Tree

Overfitting

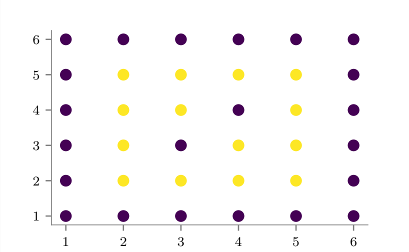
Overfitting is also known as *high variance*, since very small changes in data can lead to very different models.
Decision tree learned has depth of 10.



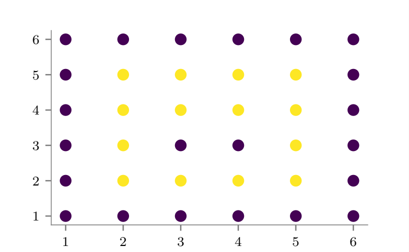
Intution for Variance

A small change in data can lead to very different models.

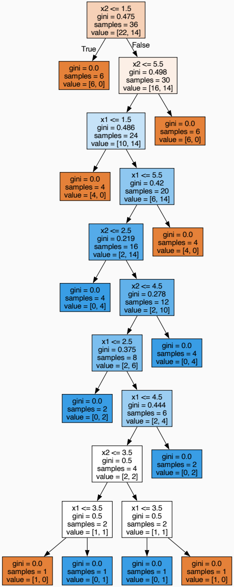
Dataset 1

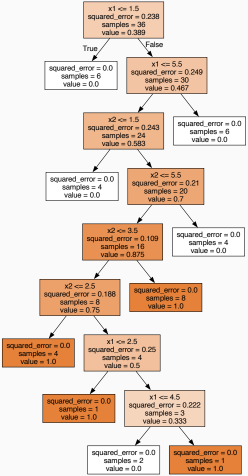


Dataset 2

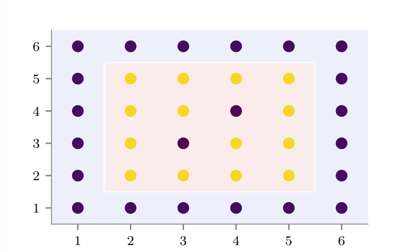


Intution for Variance

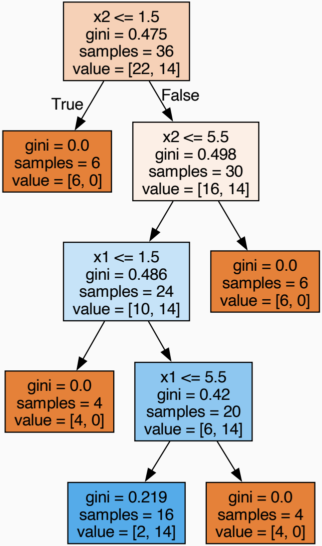




A Good Fit

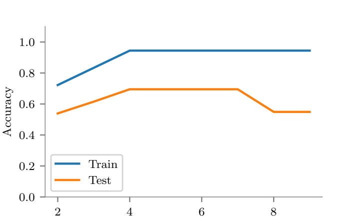


1
2
3
4
5
6

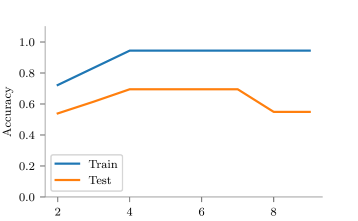


x2 <= 1.5
gini = 0.475
samples = 36
value = [22, 14]
True
False
gini = 0.0
samples = 6
value = [6, 0]
x2 <= 5.5
gini = 0.498
samples = 30
value = [16, 14]
x1 <= 1.5
gini = 0.486
samples = 24
value = [10, 14]
gini = 0.0
samples = 6
value = [6, 0]
gini = 0.0
samples = 4
value = [4, 0]
x1 <= 5.5
gini = 0.42
samples = 20
value = [6, 14]
gini = 0.219
samples = 16
value = [2, 14]
gini = 0.0
samples = 4
value = [4, 0]

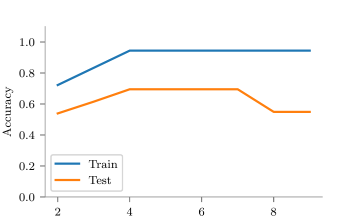
Accuracy vs Depth Curve



Accuracy vs Depth Curve



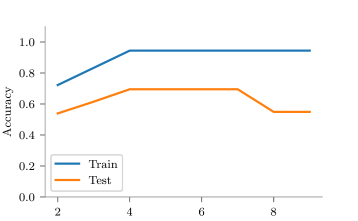
As depth increases, train accuracy improves



As depth increases, train accuracy improves

As depth increases, test accuracy improves till a point

Accuracy vs Depth Curve



As depth increases, train accuracy improves

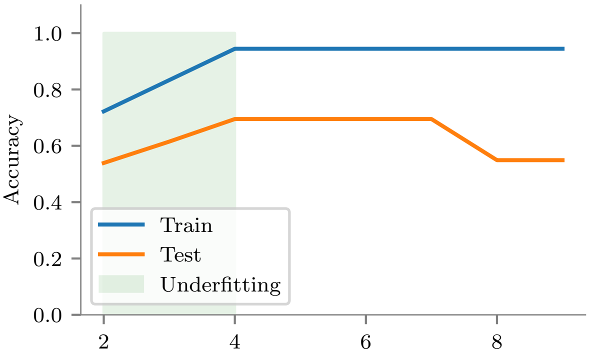
As depth increases, test accuracy improves till a point

At very high depths, test accuracy is not good (overfitting).

Accuracy vs Depth Curve : Underfitting

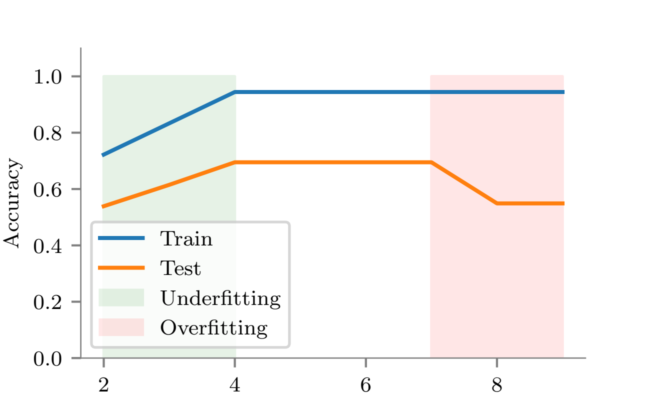
The highlighted region is the underfitting region.

Model is too simple (less depth) to learn from the data.



Accuracy vs Depth Curve : Overfitting

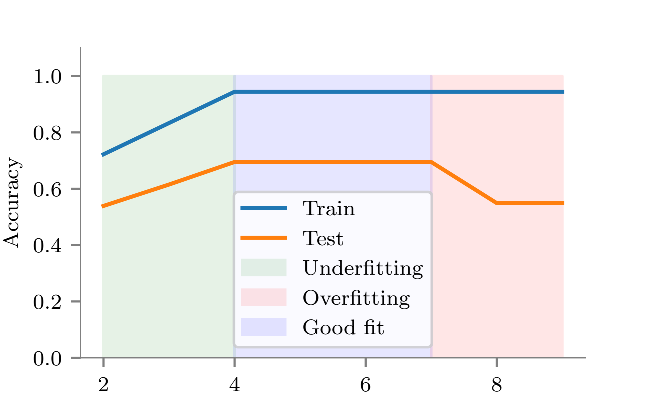
The highlighted region is the overfitting region.
Model is complex (high depth) and hence also learns the anomalies in data.



Accuracy vs Depth Curve

The highlighted region is the good fit region.

We want to maximize test accuracy while being in this region.



The big question!?

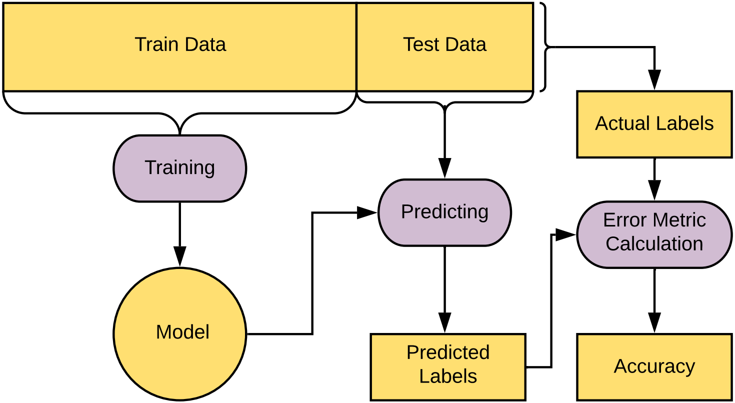
How to find the optimal depth for a decision tree?

The big question!?

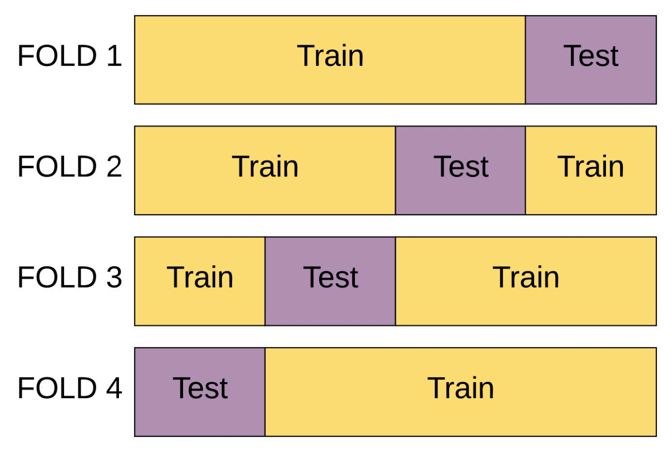
How to find the optimal depth for a decision tree?

Use cross-validation!

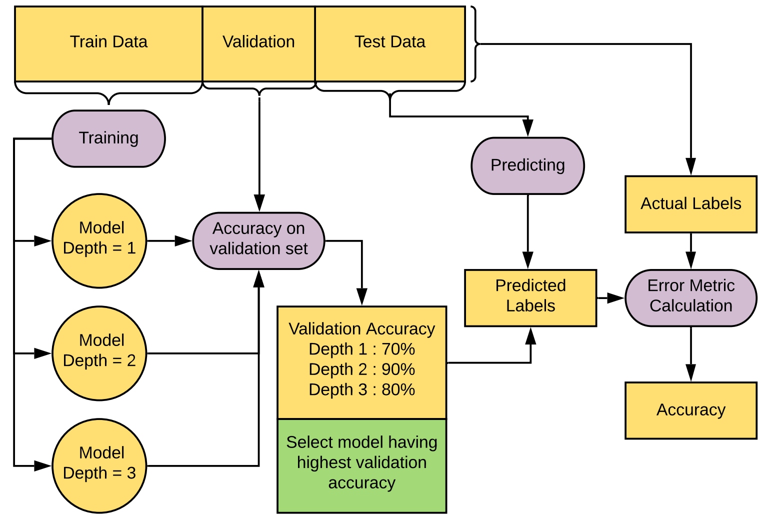
Our General Training Flow



K-Fold cross-validation: Utilise full dataset for testing



The Validation Set

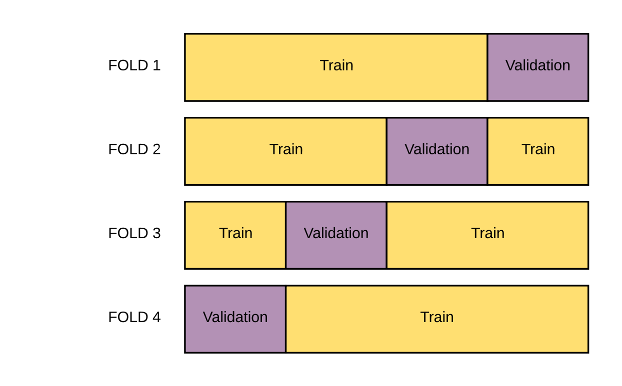


Nested Cross Validation

Divide your training set into K equal parts.

Cyclically use 1 part as "validation set" and the rest for training.

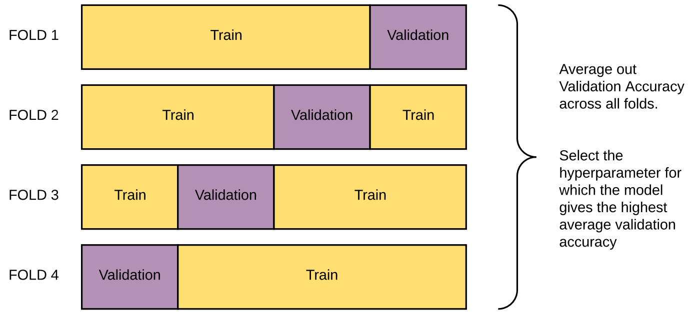
Here $K = 4$



Nested Cross Validation

Average out the validation accuracy across all the folds

Use the model with highest validation accuracy



Next time: Ensemble Learning

- How to combine various models?
- Why to combine multiple models?
- How can we reduce bias?
- How can we reduce variance?