WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

Advanced Programming for Data Science

Week 9: Model Building and Tuning
Information Systems and Management
Warwick Business School

Model Improvement

Improving the model

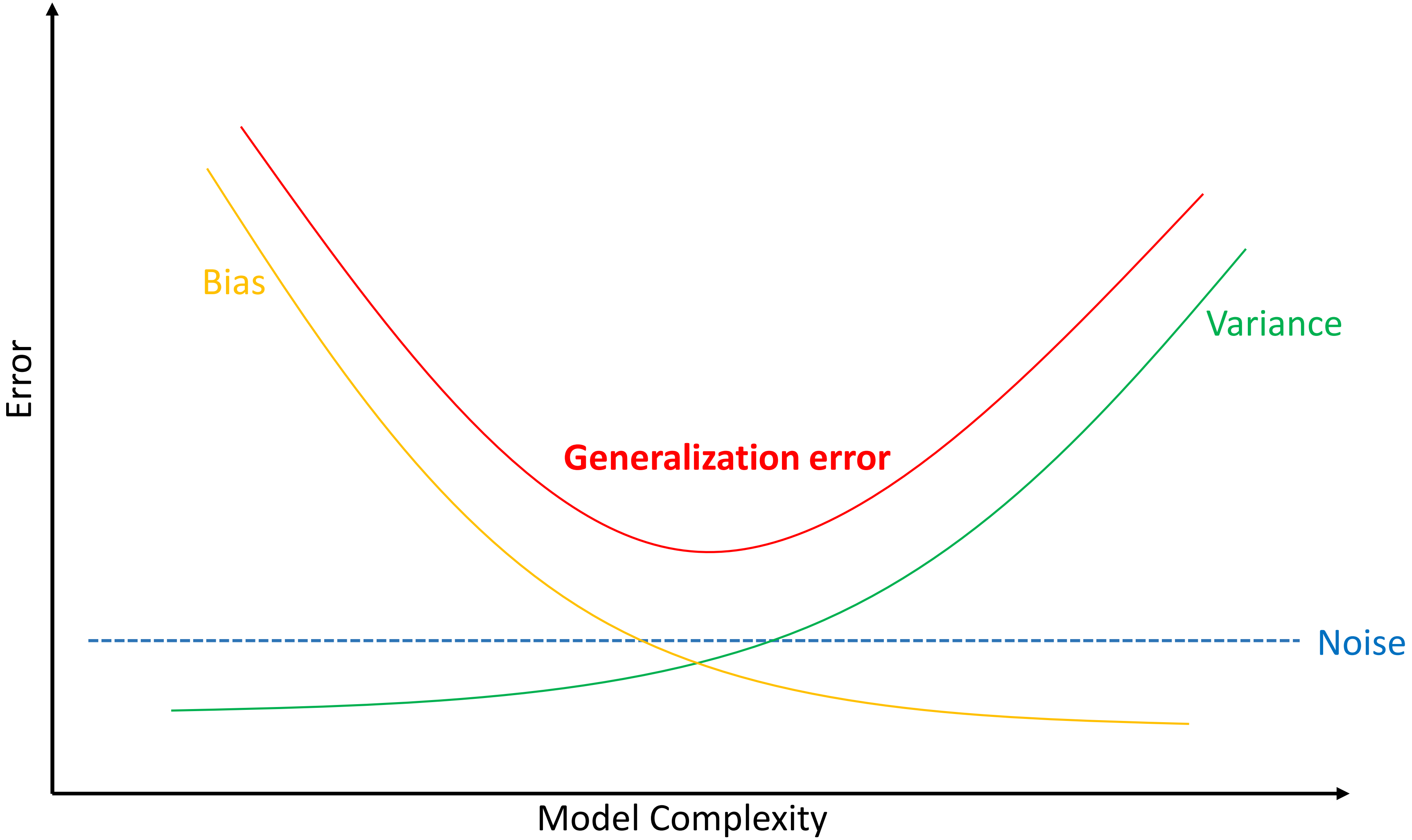
- Sources of errors.
  - Noise

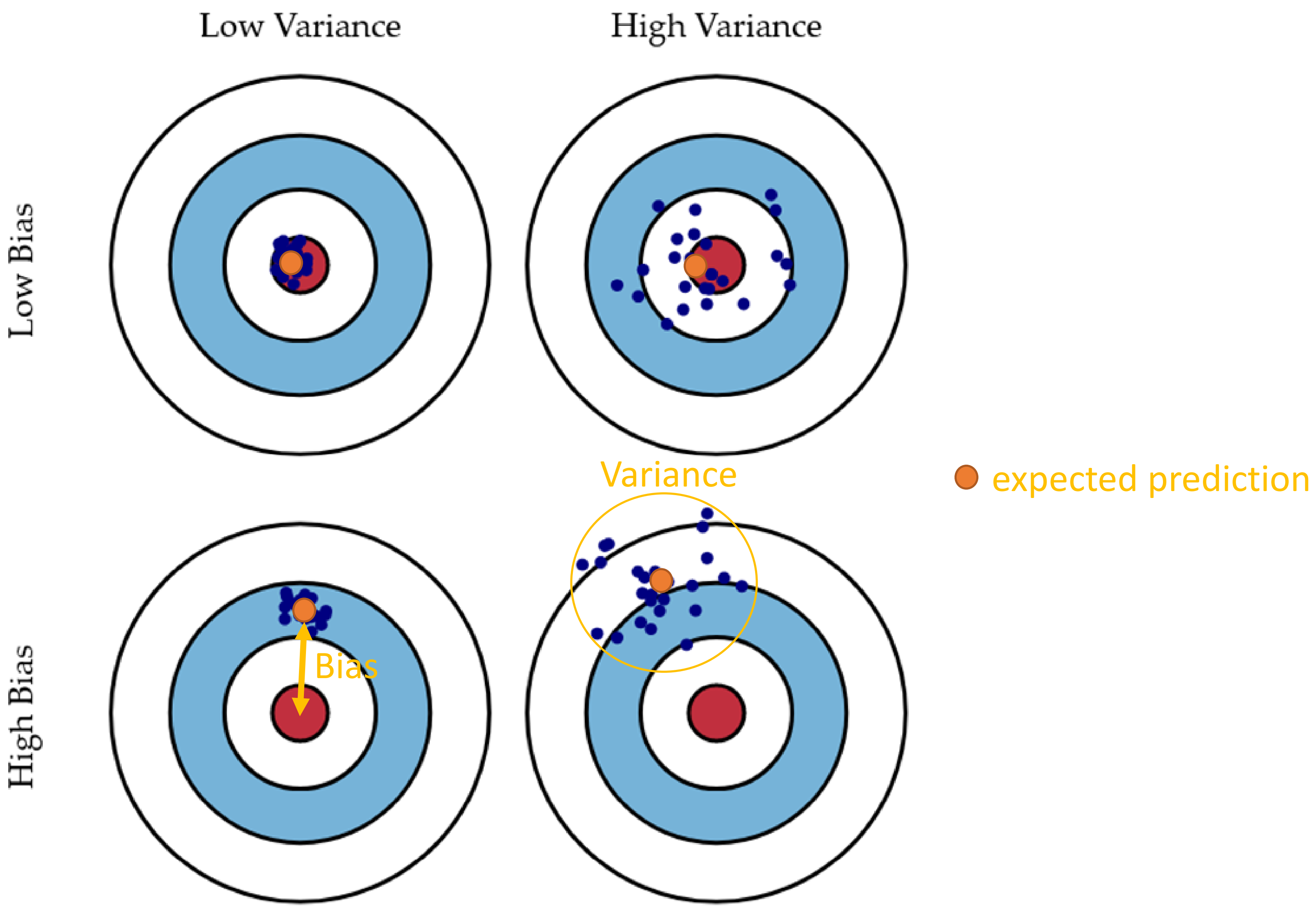
    Irreducible error caused by unobserved factors.
  - Variance

    Variability of model prediction.
  - Bias

    Difference between the expected prediction and the actual value.

Trade-off between bias and variance





An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Bias and Variance Trade-off

- Assume the true relationship between input X and output Y is:

$$Y = f(X) + \varepsilon$$

- The estimated model from the sample data is $f^*(X)$.
- The expected squared prediction error for a given x is:

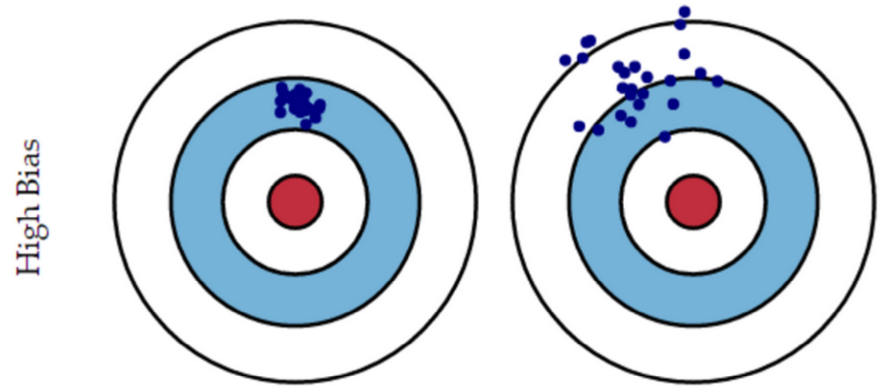
$$Error(x) = E[(f^*(x) - f(x))^2]$$

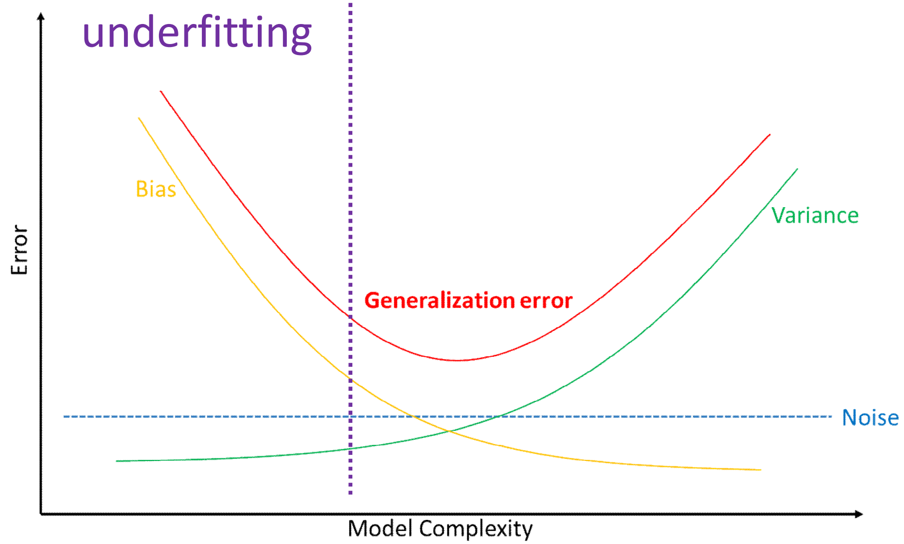
- This can be decomposed into bias and variance components:

$$Error(x) = \underbrace{\boxed{(E[f^*(x)] - f(x))^2}}_{\text{Bias}^2} + \underbrace{\boxed{E[(f^*(x) - E[f^*(x)])^2]}}_{\text{Variance}}$$

Underfitting

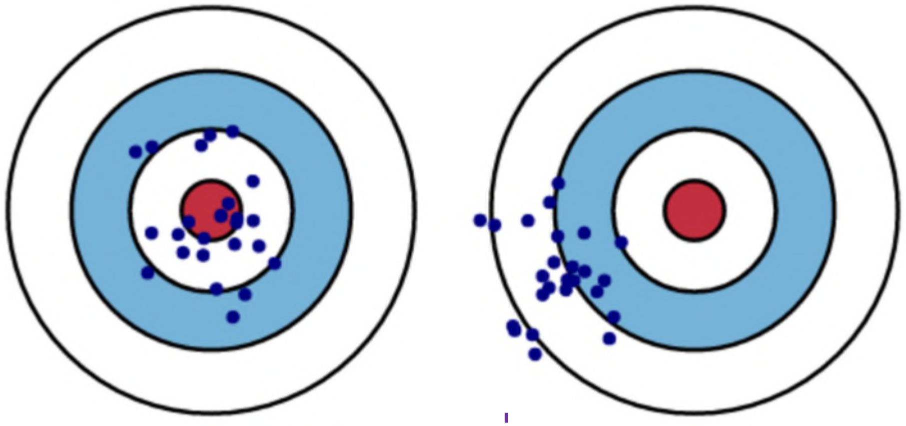
- You model cannot even fit the training dataset well.
  - has high bias
  - often due to oversimplified model, too many assumptions.
- To improve:
  - More features
  - More complex

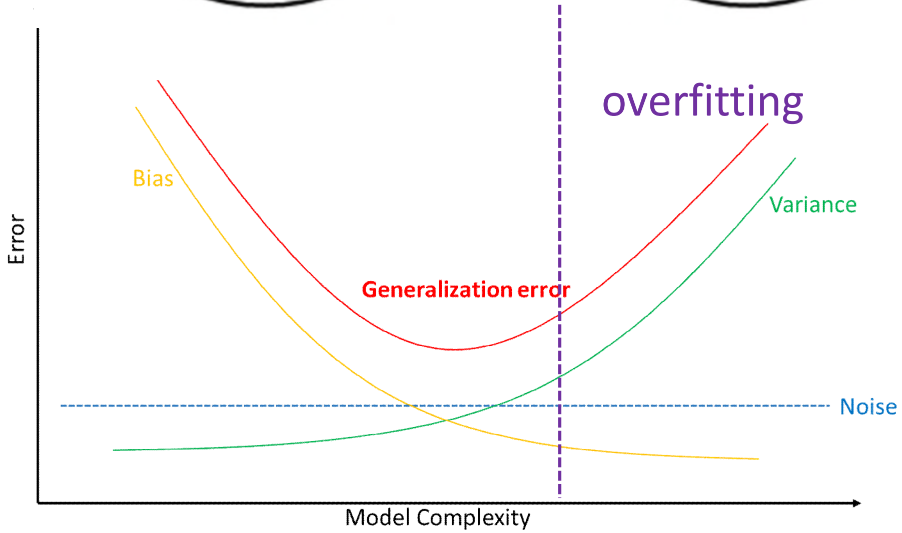




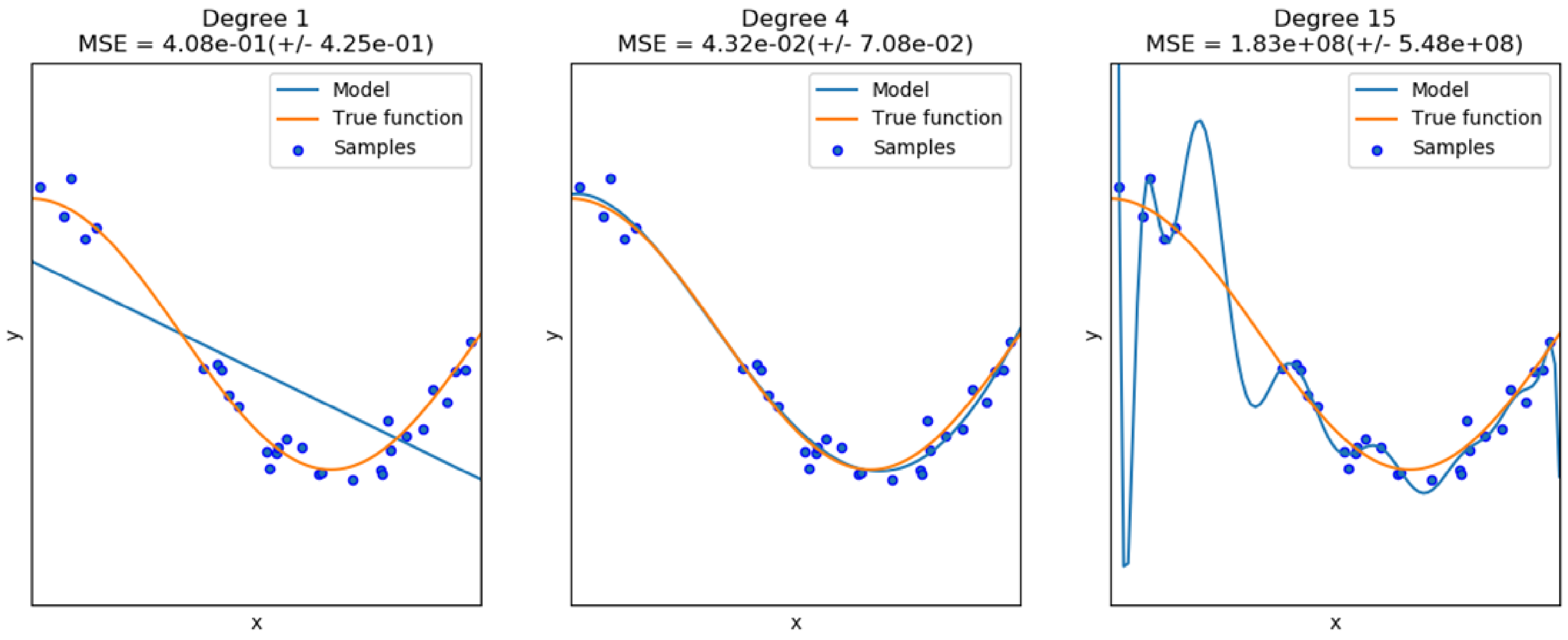
overfitting

- You model fit training dataset too well, but poorly on testing dataset.
  - **Probably** high variance.
  - Overcomplicated model, too flexible.
- To improve:
  - More data
  - Less complex <- Regularization





Underfitting vs. Overfitting



https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html

Improving Model

- Data
- Feature
- Assumption

Improving Decision Tree

DecisionTreeClassifier

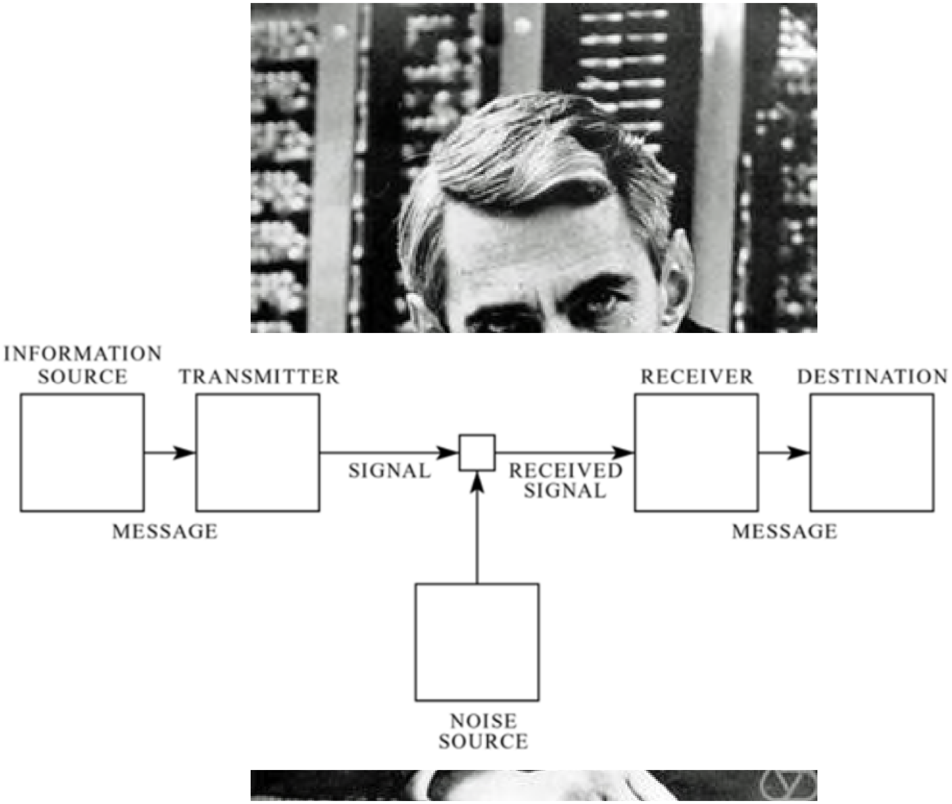
- *class* sklearn.tree.DecisionTreeClassifier(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None,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

Entropy in Information Theory

- Information theory

"Mathematical Theory of Communication"

1. Shannon Limit
2. Architecture of Communication Systems
3. Digital Representation (bit: **bi**nary dig**it**)
4. Entropy

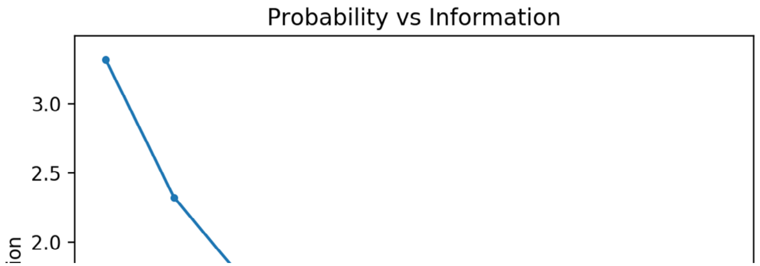


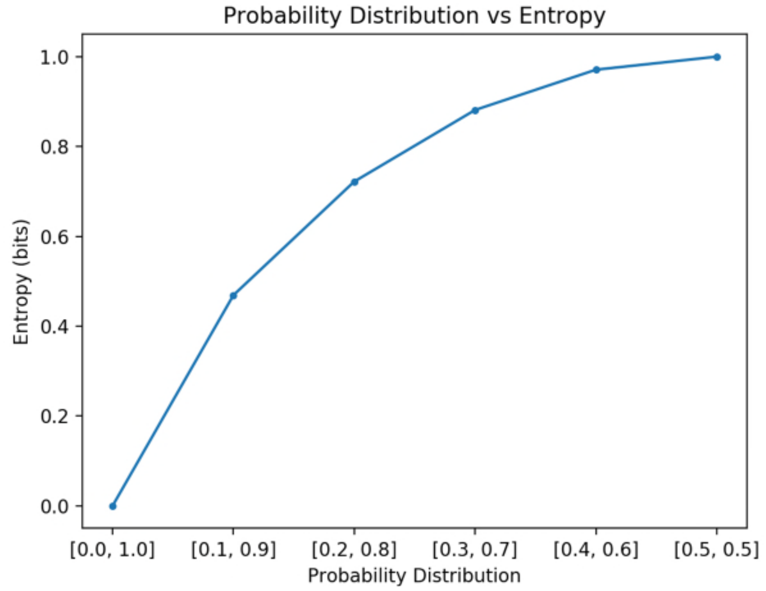
Claude Shannon
https://en.wikipedia.org/wiki/Claude_Shannon

Splitting method 1: Entropy

- Quantifying information
  - how much **surprise** there is in an event
    - Rare event: low probability and high informatio
    - Common event: high probability and low information
    - information(x) = -log(p(x))
  - Entropy
    - measure of uncertainty
    - how much information there is in a random variable

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$



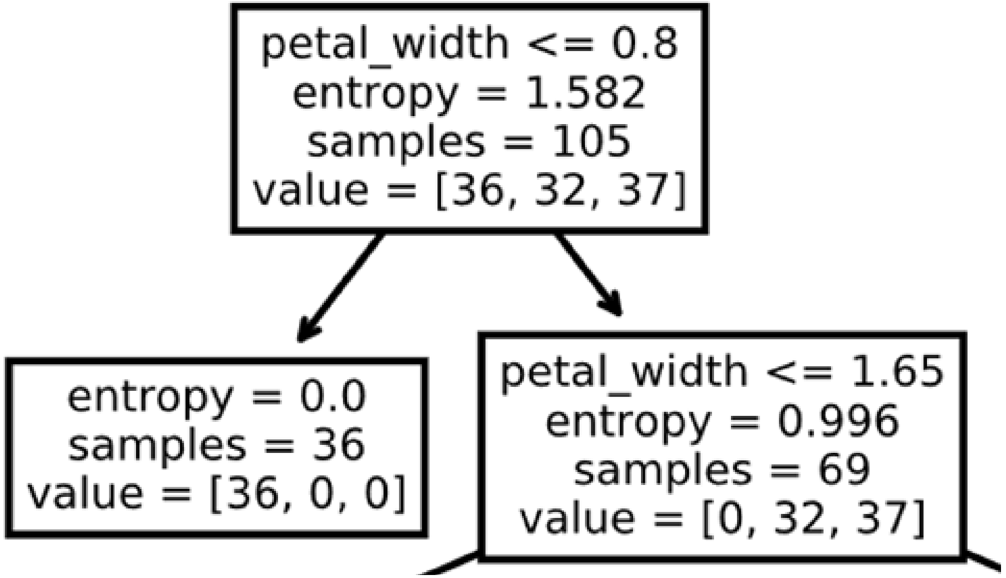


Splitting method 1: Entropy

- Information gain
- Entropy at a given node t:

$$Entropy(t) = -\sum_{j=1}^{n} p_j \cdot log_2 p_j$$

- p_j is the relative frequency of class *j* at node *t*
- $Entropy_{max} = log_2 n$
  - Records are equally distributed among all classes
  - Impure
- $Entropy_{min} = 0$
  - All records belong to one class
  - Pure
- Also known as
  - Information Gain
  - Uncertainty
  - Level of randomness



Splitting method 1: Entropy

- Quality of split at node p into k partitions (children) is

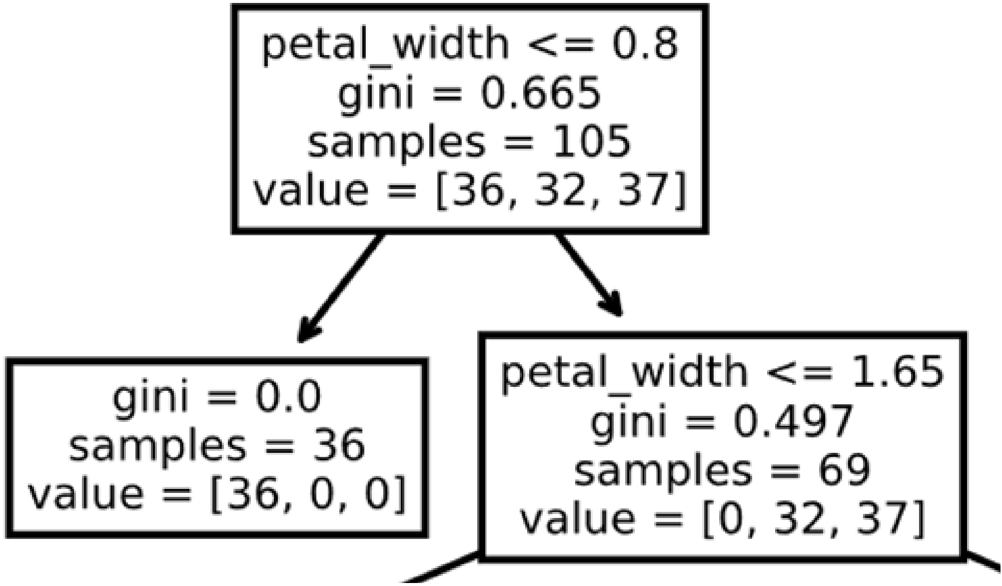
$$GAINsplit = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i)\right)$$

  - n_i = # of records in child i
  - n = # of records at parent node p
- Choose the split that maximises gain
- Disadvantage
  - Tends to prefer splits that result in large number of partitions, each pure but small.

Splitting method 2: Gini

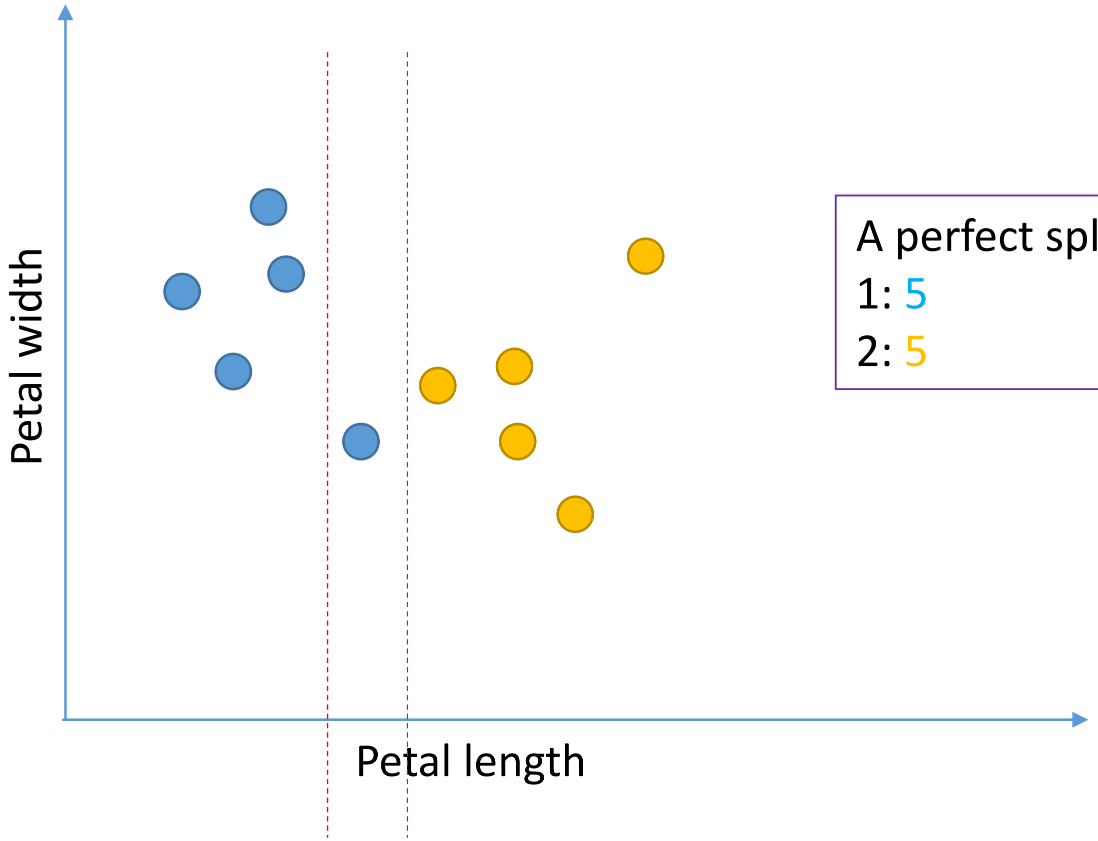
- Gini impurity
- how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset.

$$G = \sum_{i=1}^{n} p(i) \cdot (1 - p(i))$$



Gini coefficient (https://en.wikipedia.org/wiki/Gini_coefficient)

Gini: Iris example

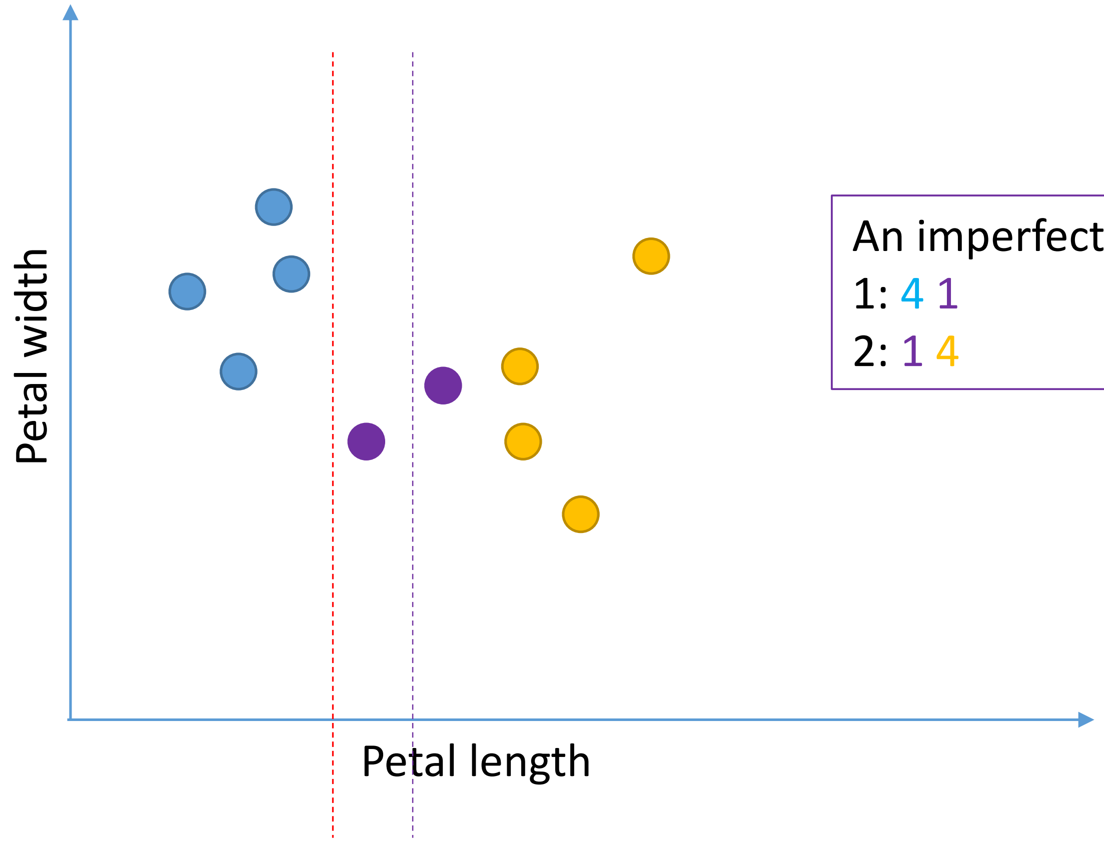


A perfect split:
1: 5
2: 5

>

An imperfect split:
1: 4
2: 1 5

Gini: Iris example



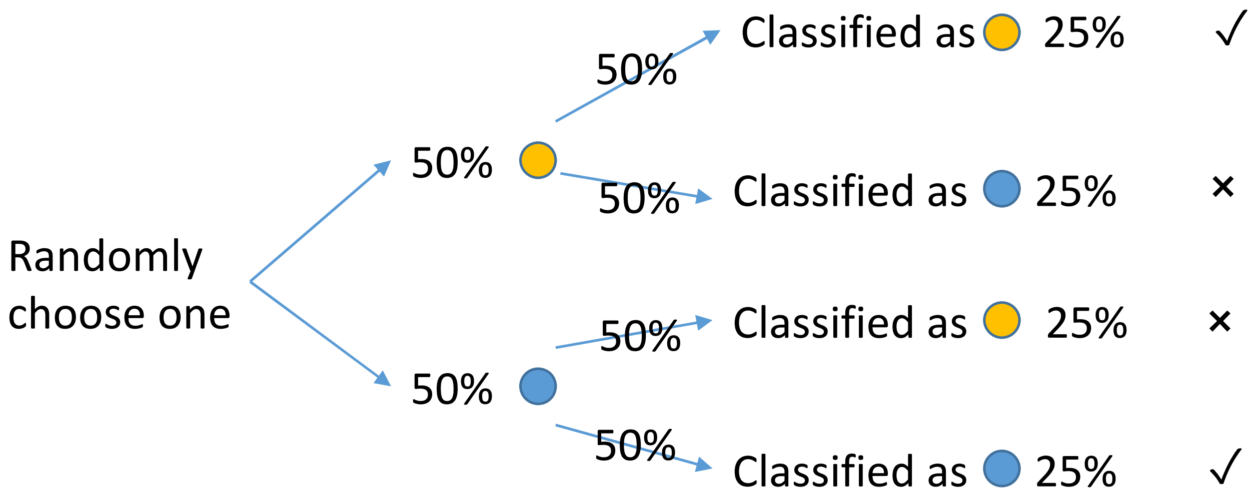
An imperfect split:
1: 4 1
2: 1 4

?

Another imperfect split:
1: 4
2: 2 4

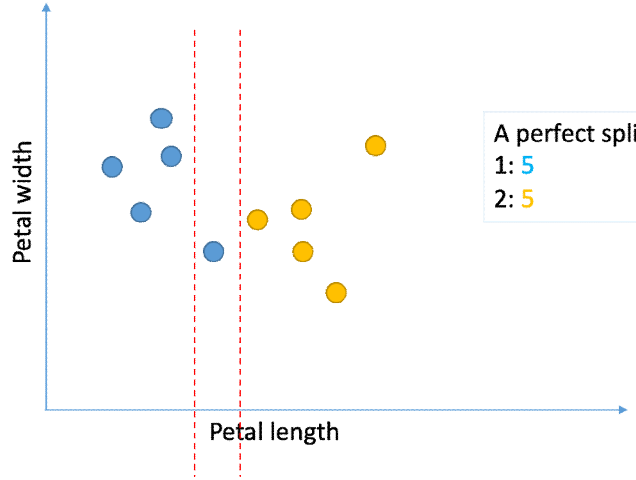
Gini impurity

- how often a **randomly** chosen element from the set would be **incorrectly** labelled if it was **randomly** labelled according to the distribution of labels in the subset.
- Original dataset: 5 5



0.5

- A perfect split
  - Subset 1: 5 Impurity: 0
  - Subset 2: 5 Impurity: 0
- A imperfect split
  - Subset 1: 4 Impurity: 0
  - Subset 2: 1 5 Impurity: 0.278



A perfect split:
1: 5
2: 5

\>

An imperfect split:
1: 4
2: 1 5

$$\frac{1}{6} * \frac{5}{6} + \frac{5}{6} * \frac{1}{6}$$

Quality of Splitting

- Weighted sum of impurity of each branch by the size of subset.
- A perfect split
  - Weighted impurity: 0.5*0 + 0.5*0 = 0
  - Reduction: 0.5 – 0 = 0.5
- An imperfect split
  - Weighted impurity: 0.4*0 + 0.6*0.278 = 0.167
  - Reduction: 0.5 – 0.167 = 0.333

Maximize the reduction: Gini Gain

Entropy vs. Gini

- They will give you almost the same results in most cases.
  - ❖Gini focuses more on misclassification.
  - ❖Entropy works better with highly skewed data.
- Gini is normally preferred, also the default for Sklearn's DecisionTreeClassifier, due to computational intensity.

$$E = -\sum_{j=1}^{n} p_j \cdot log_2 p_j$$

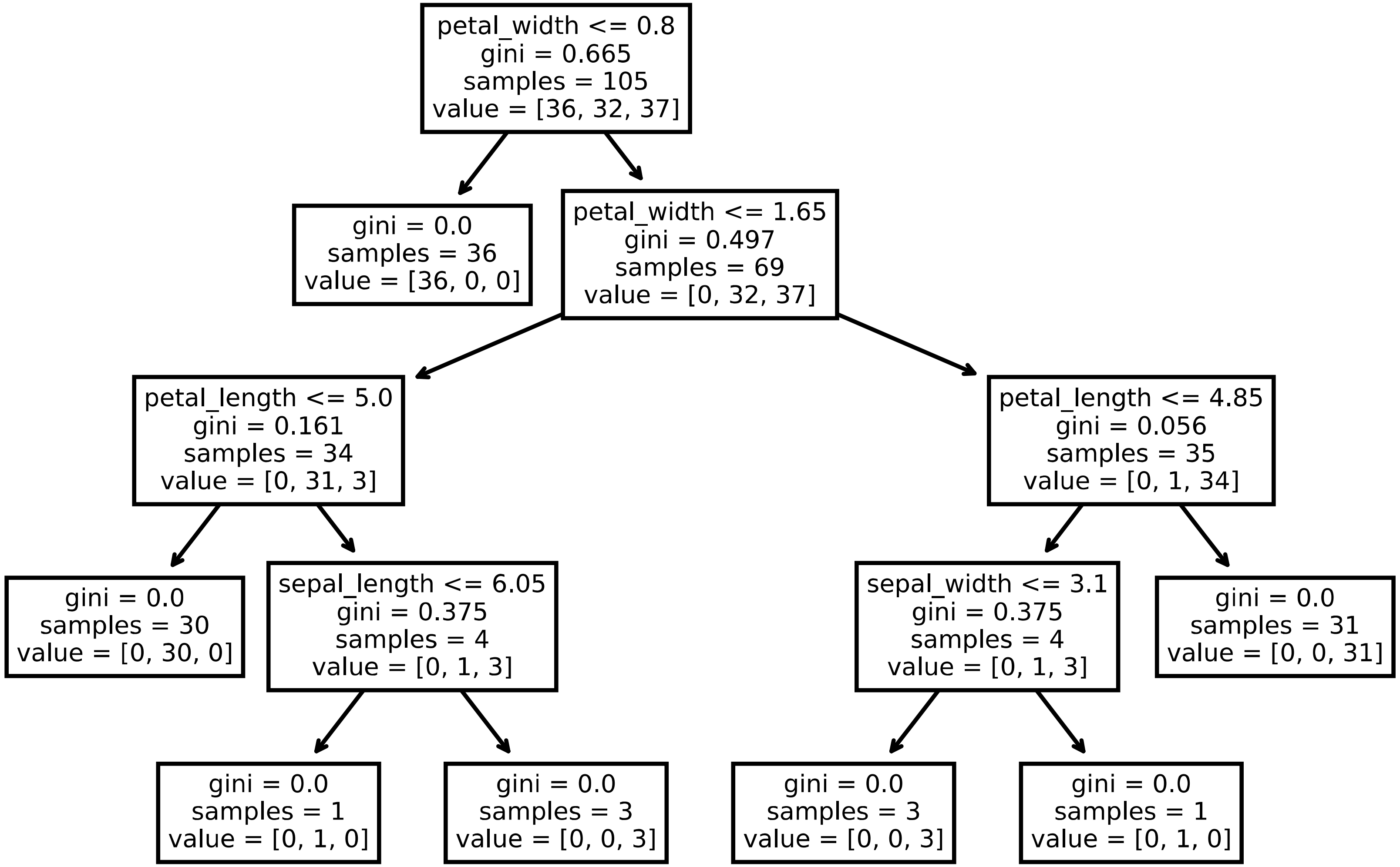
$$G = \sum_{i=1}^{n} p(i) \cdot (1 - p(i))$$

Splitter

- Best vs. random

Assumption and Model Complexity: Tree Pruning

- max_depth & max_features
- min_samples_leaf & min_samples_split
- min_impurity_decrease


petal_width <= 0.8
gini = 0.665
samples = 105
value = [36, 32, 37]
gini = 0.0
samples = 36
value = [36, 0, 0]
petal_width <= 1.65
gini = 0.497
samples = 69
value = [0, 32, 37]
petal_length <= 5.0
gini = 0.161
samples = 34
value = [0, 31, 3]
petal_length <= 4.85
gini = 0.056
samples = 35
value = [0, 1, 34]
gini = 0.0
samples = 30
value = [0, 30, 0]
sepal_length <= 6.05
gini = 0.375
samples = 4
value = [0, 1, 3]
sepal_width <= 3.1
gini = 0.375
samples = 4
value = [0, 1, 3]
gini = 0.0
samples = 31
value = [0, 0, 31]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 3
value = [0, 0, 3]
gini = 0.0
samples = 3
value = [0, 0, 3]
gini = 0.0
samples = 1
value = [0, 1, 0]