الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

AI for Interdisciplinary Research

Part 2: Opening the Box (1)

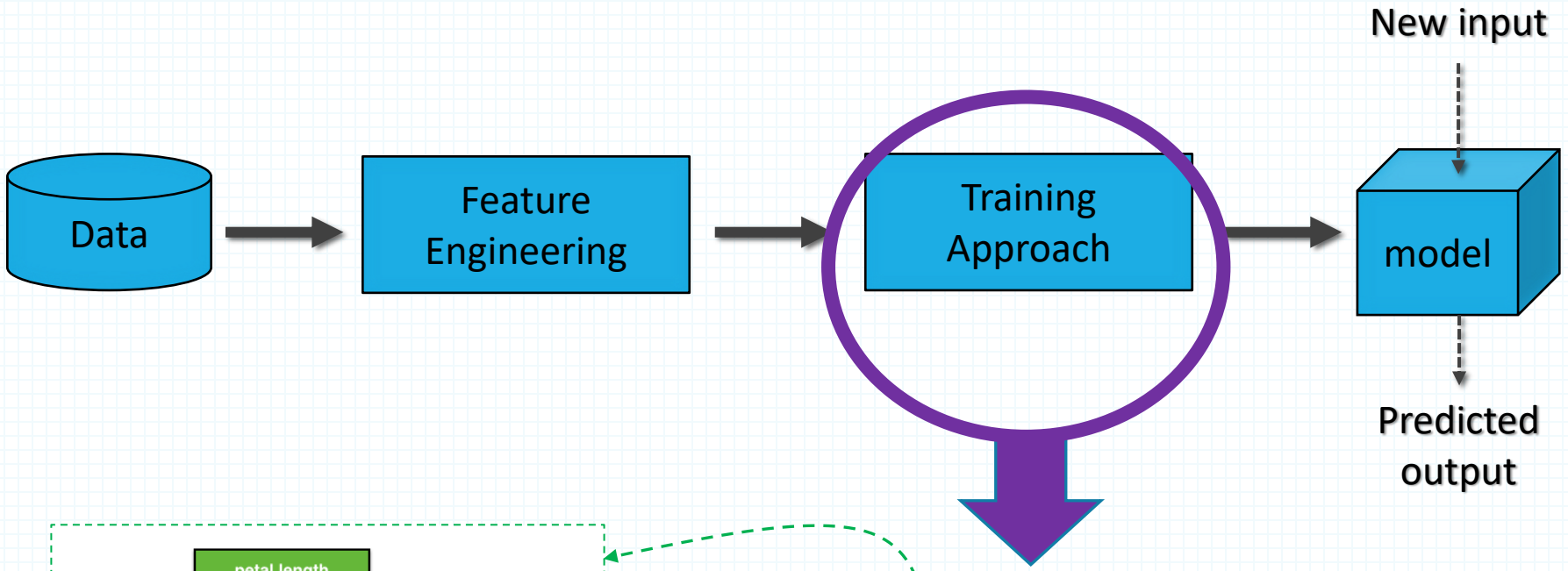
Dr. Riad Sonbol

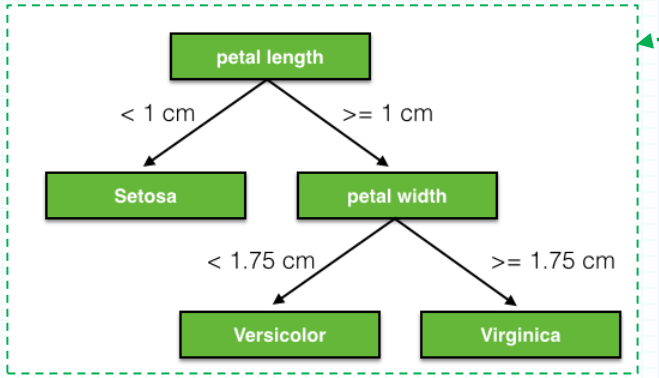
April 2026

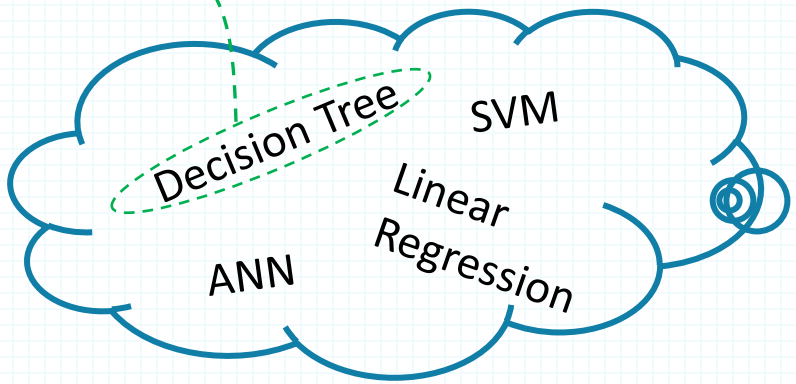
Road Map

- AI as a Black Box – Concepts and Foundations
- Opening the Box (1) – Traditional Machine Learning
- Opening the Box (2) – Deep Learning
- Modern AI and Research Directions

Traditional ML Pipeline





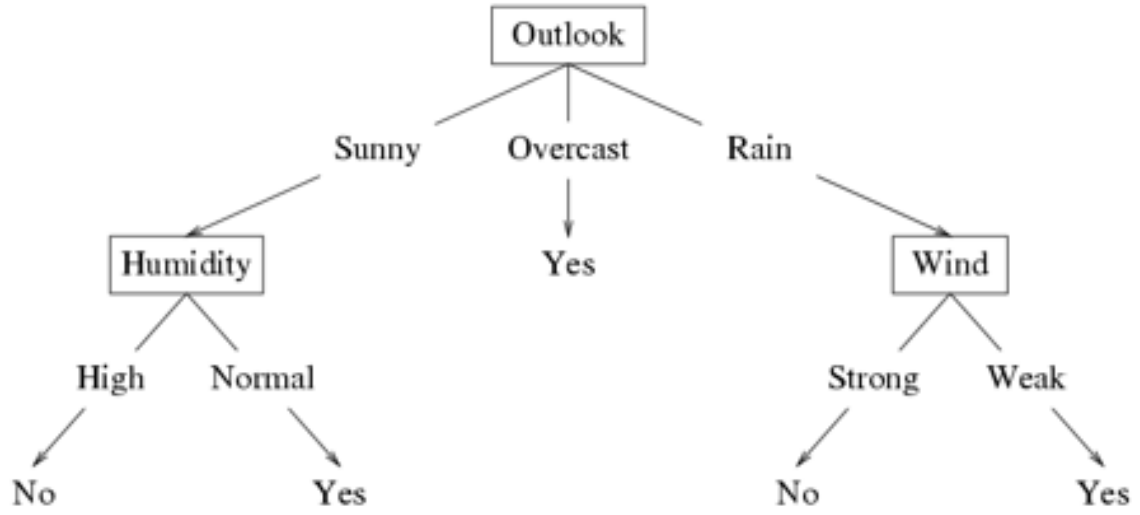


```python
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

# 1. Load a sample dataset (Iris)
iris = datasets.load_iris()
X = iris.data
y = iris.target

# 2. Split data into training (80%) and testing (20%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_

# 3. Initialize the SVM Classifier
# 'linear' is common for simple data; 'rbf' is default for complex patterns
clf = SVC(kernel='linear')

# 4. Train the model
clf.fit(X_train, y_train)

# 5. Make predictions and check accuracy
predictions = clf.predict(X_test)
print(f"Accuracy: {accuracy_score(y_test, predictions):.2f}")
```

أشجار القرار
Decision Tree

Decision Tree Hypothesis Space

- Internal nodes test the value of particular features xi and branch according to the results of the test.
- Leaf nodes specify the class.



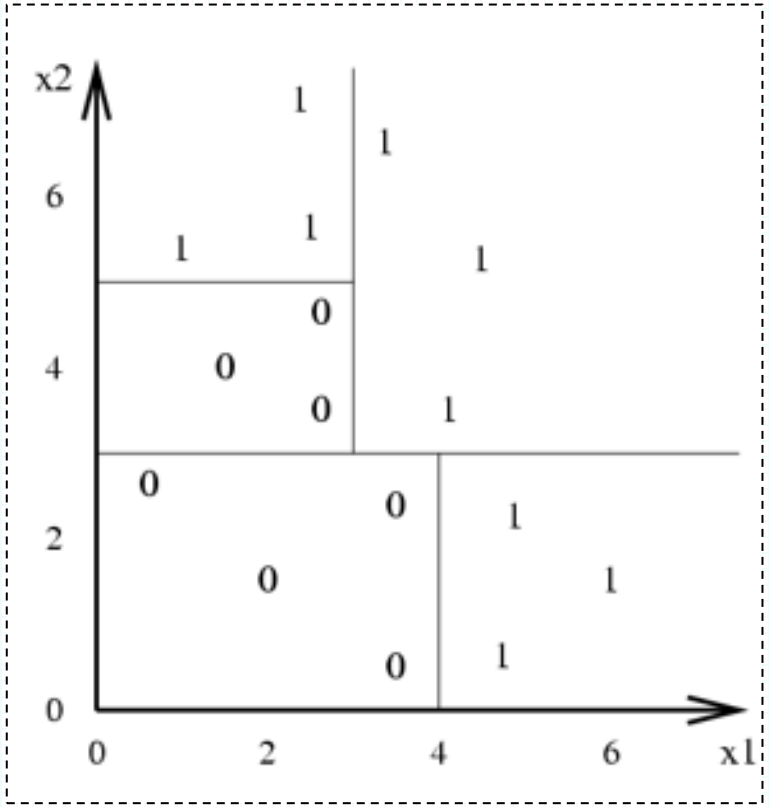
Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (Sunny, Hot, High, Strong)$ will be classified as **No**. The **Temperature** feature is irrelevant.

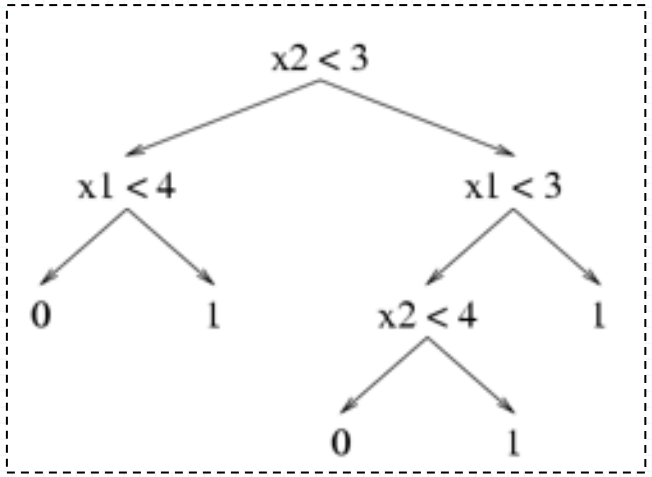
PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Tree Decision Boundaries

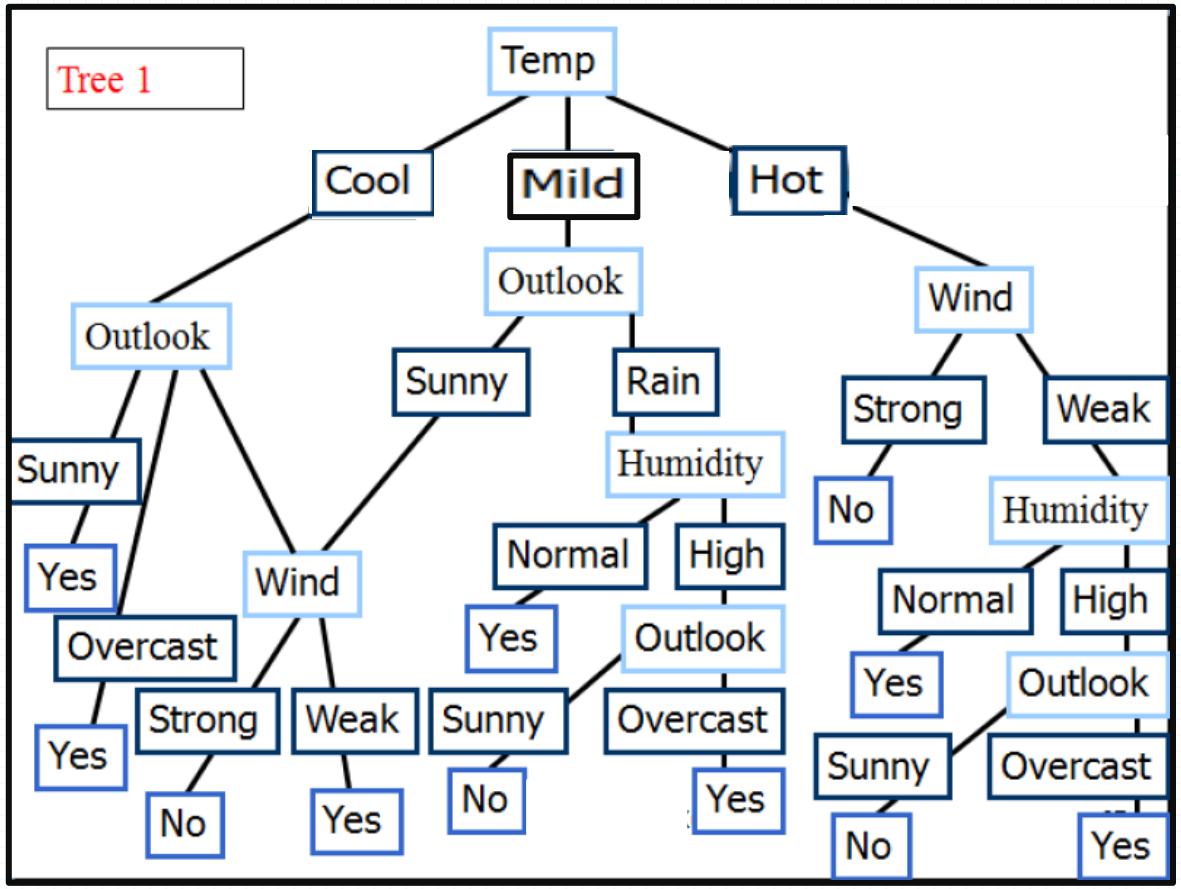
- Decision trees divide the feature space into axis-parallel rectangles, and label each rectangle with one of the K class.

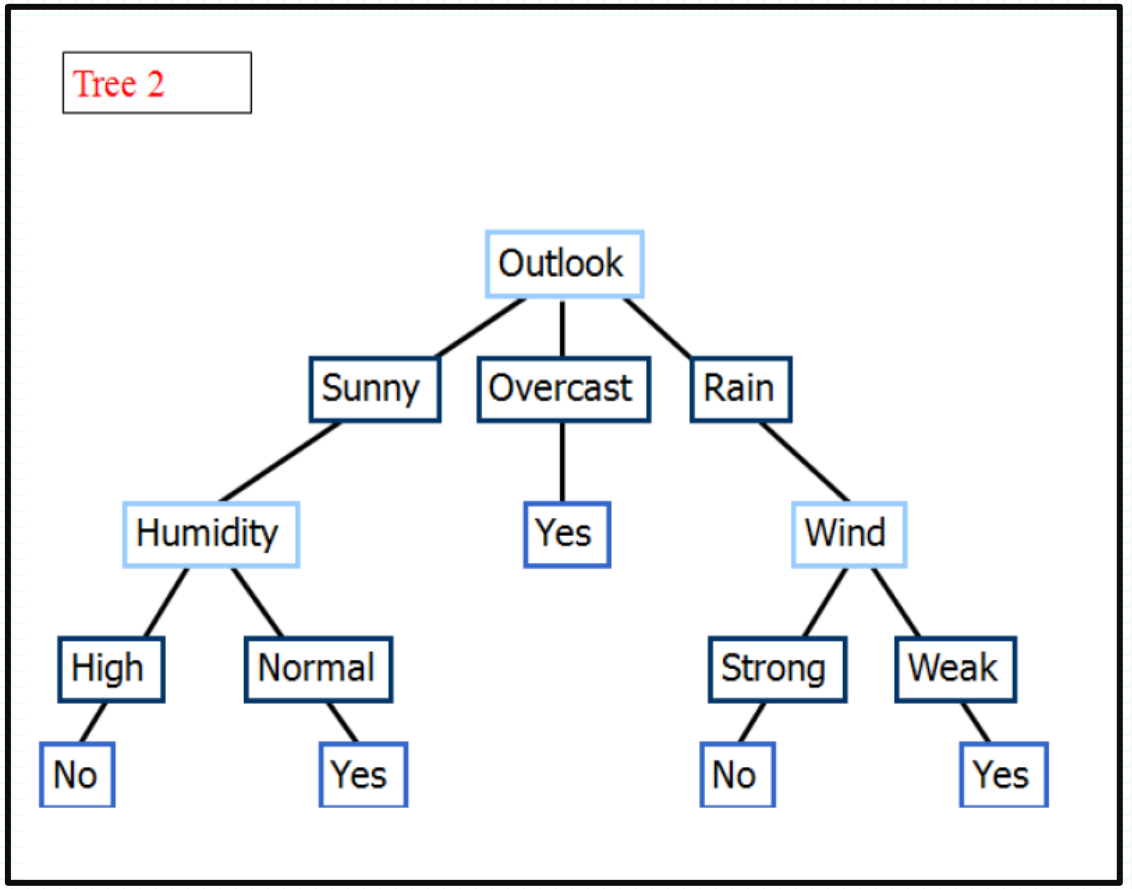




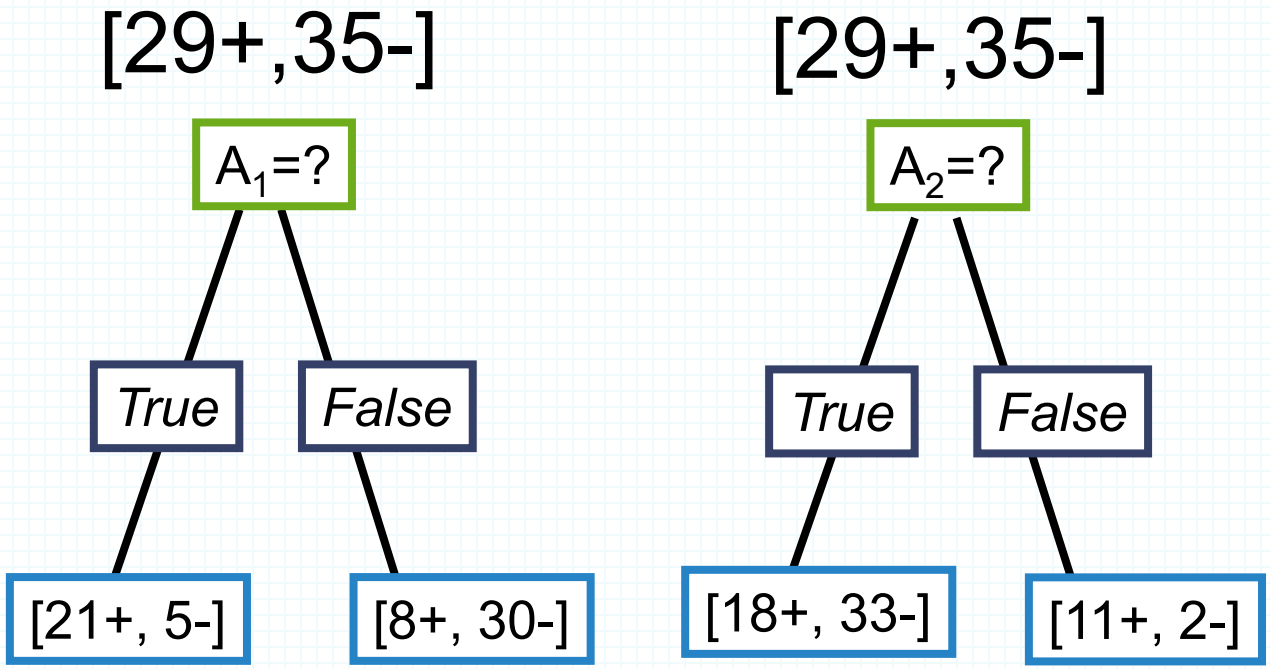
- Advantages:
  - explicit relationship among features
  - human interpretable model
- Limitations?

Which Attribute is "best"?

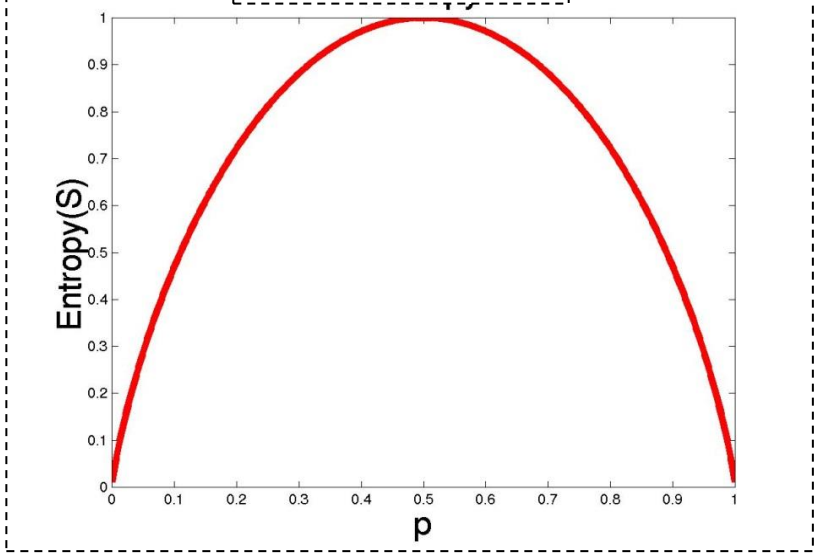




Which Attribute is "best"?



Entropy

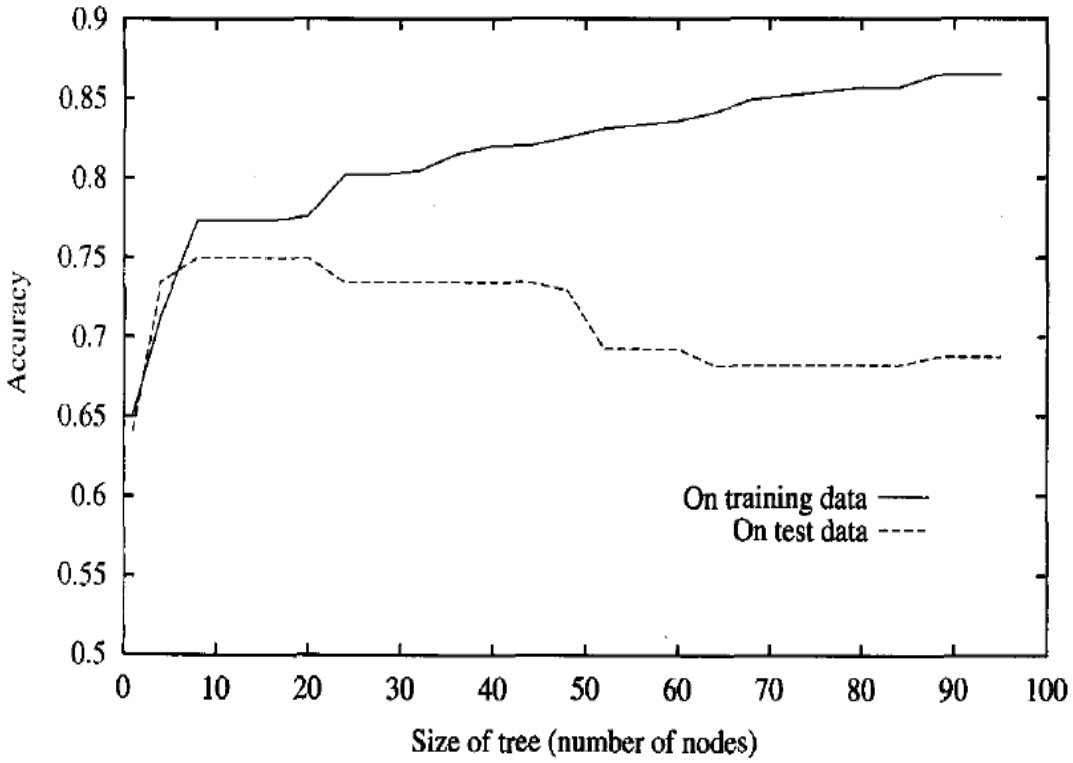


Entropy measures the impurity of S

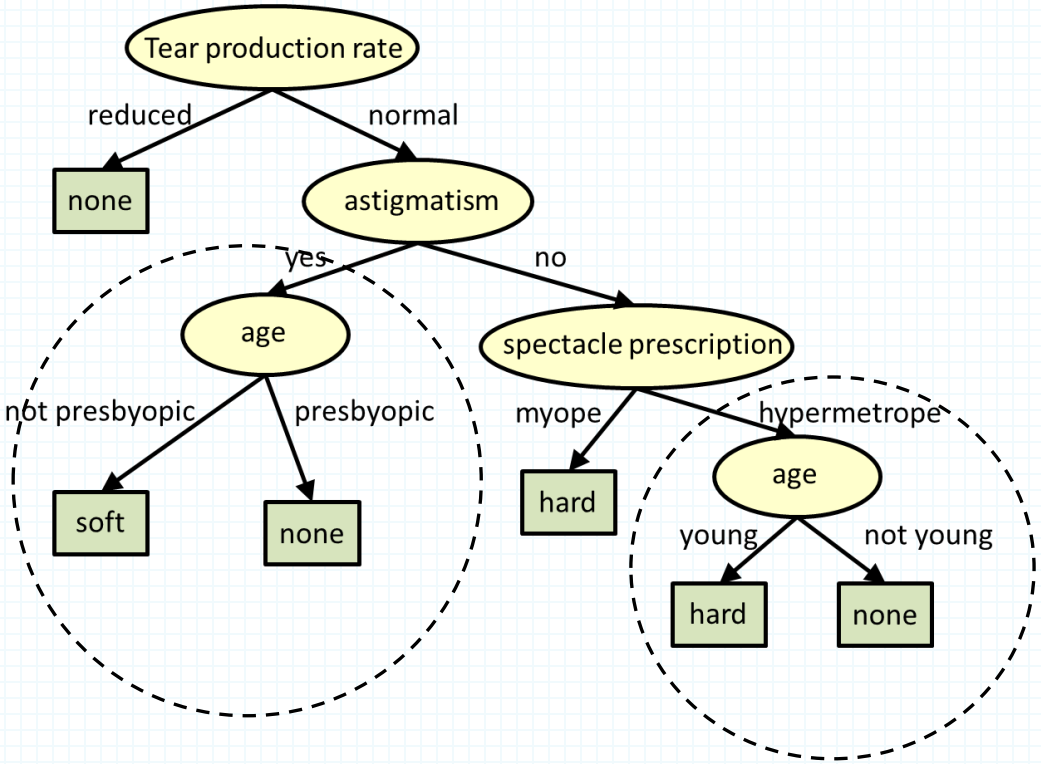
$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$

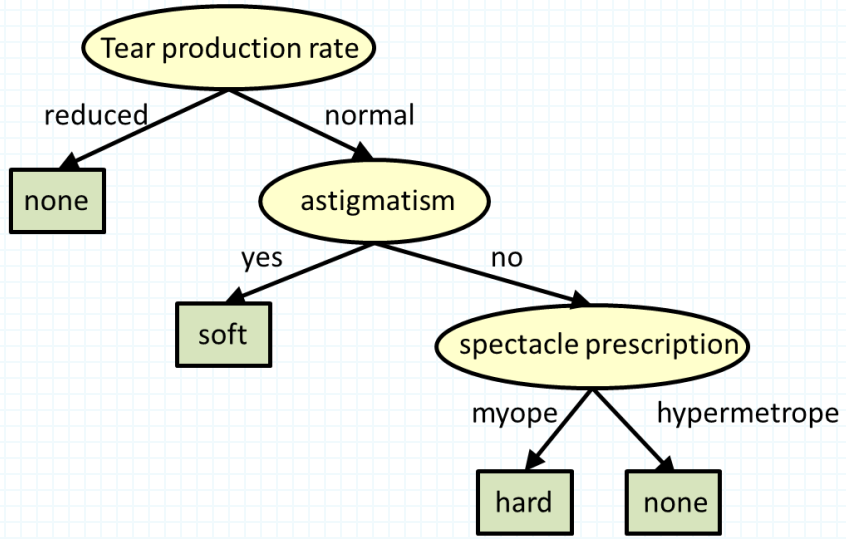
Challenges: Overfitting

- The idea: Stop growing the tree if the improvement is minor
- Pruning Decision Trees involves a set of techniques that can be used to simplify a Decision Tree, and enable it to generalize better (to avoid overtraining).
- Overfitting occurs when the model trains too well and starts learning noise other than the required characteristics



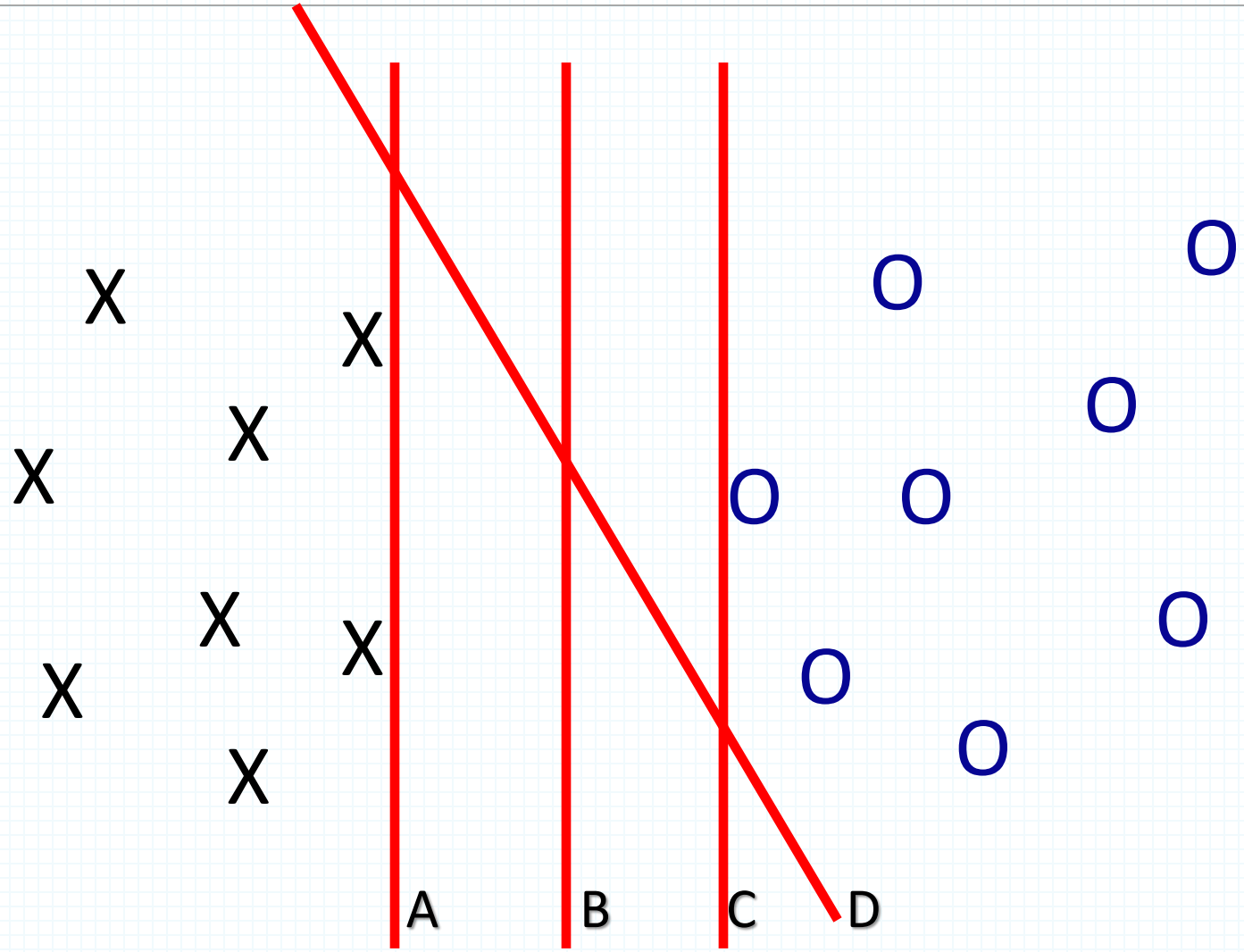
Challenges: (2) Overfitting



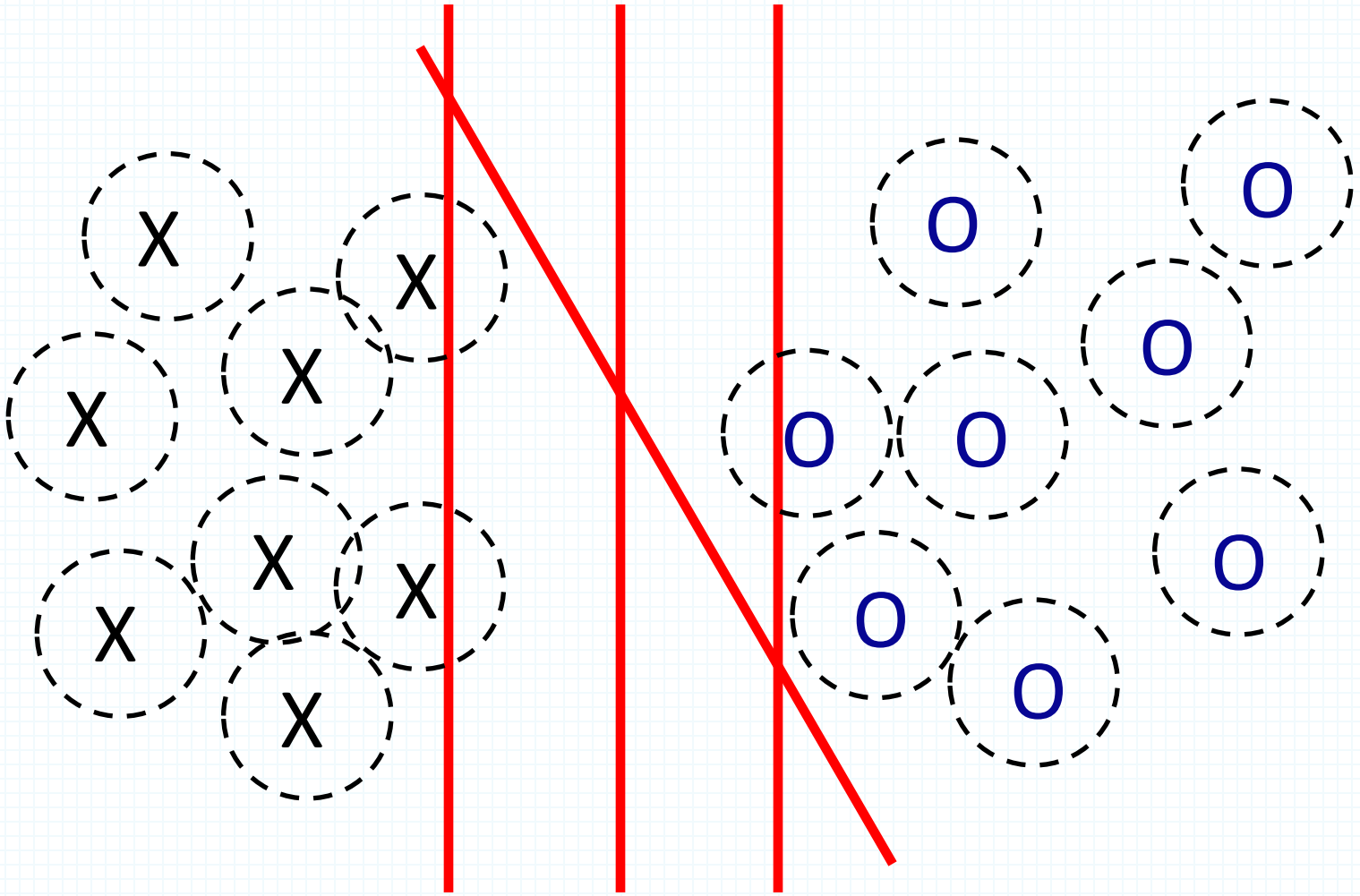


Support Vector Machine (SVM)

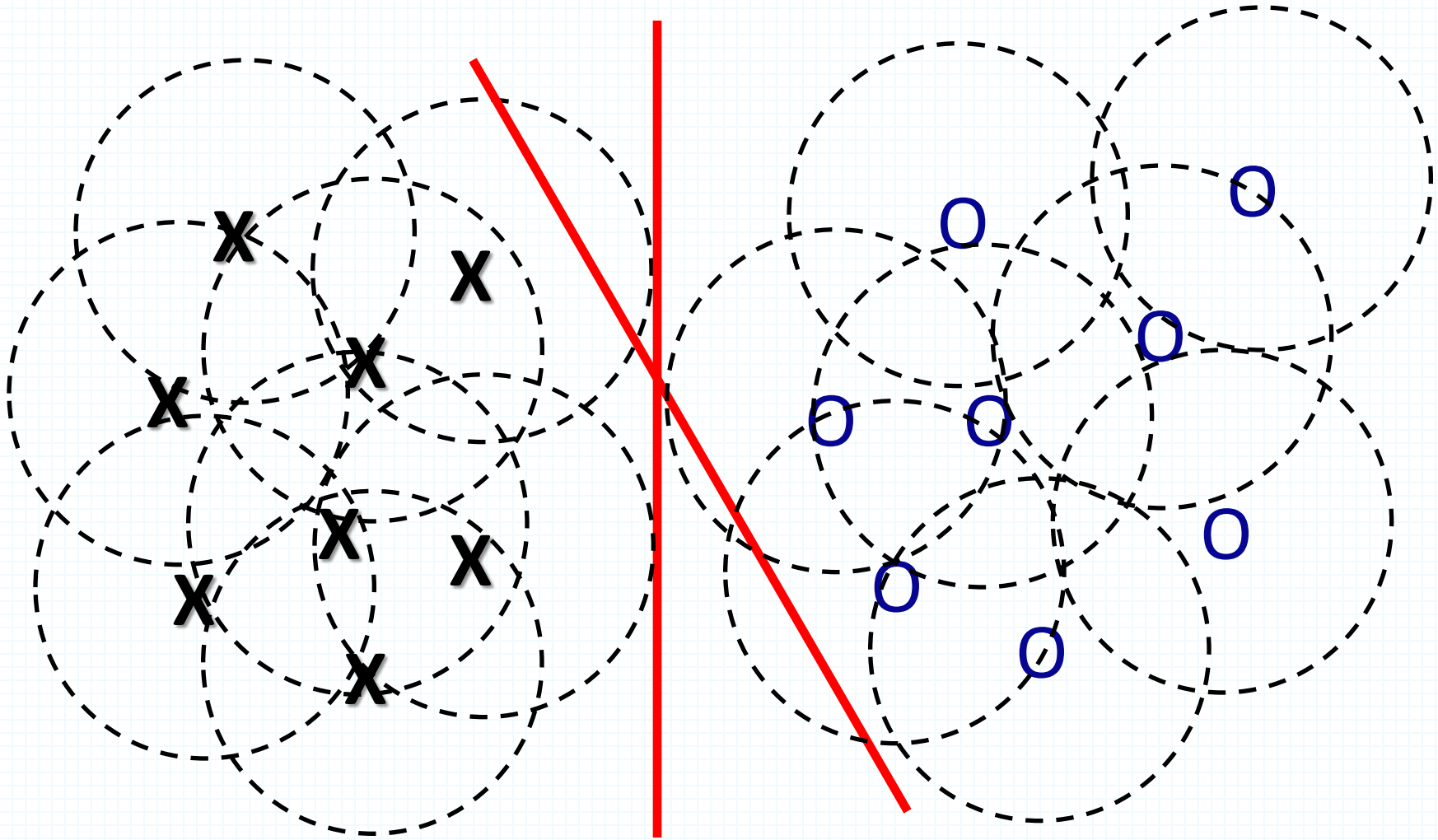
Intuitions



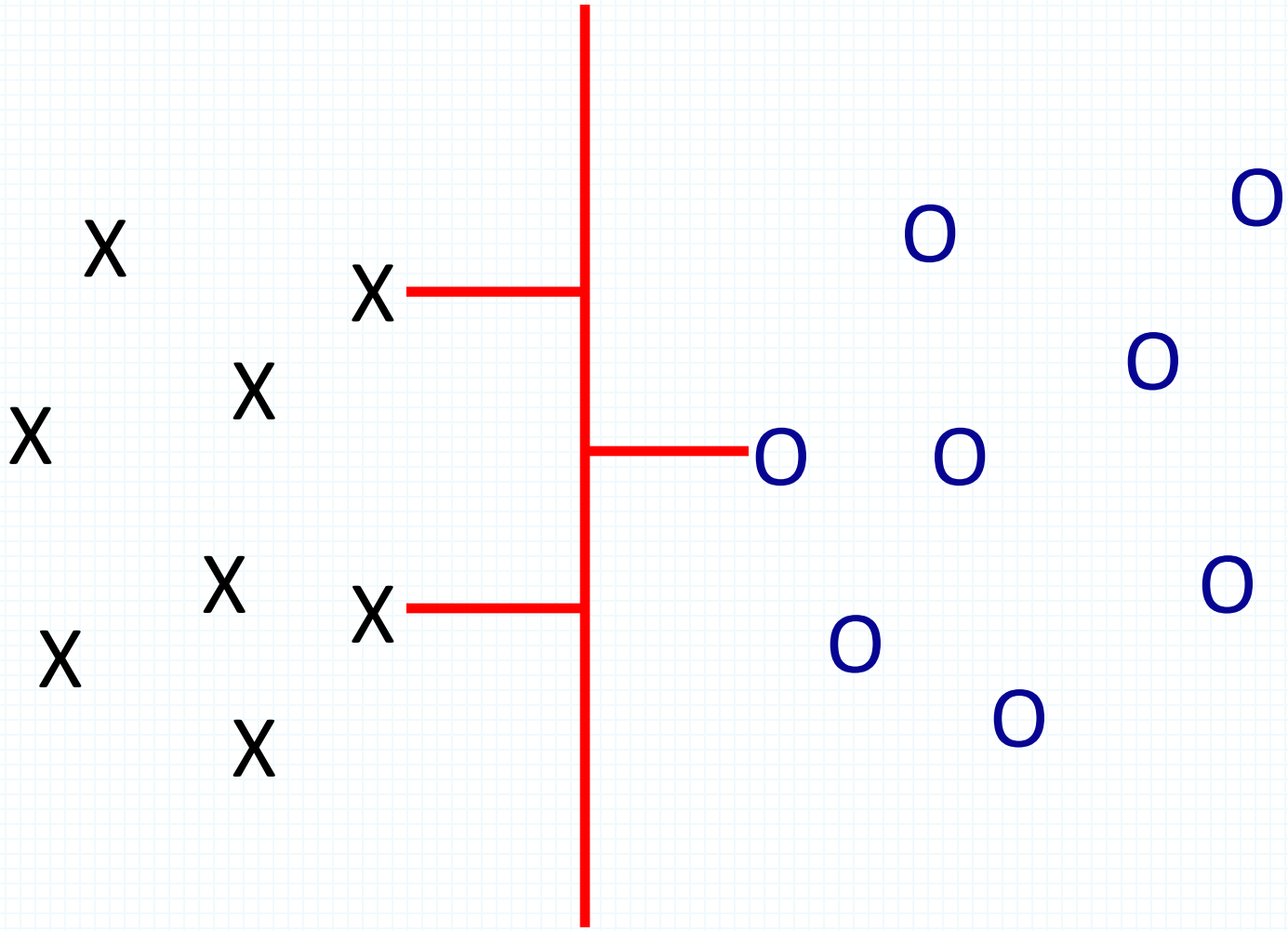
Ruling Out Some Separators



Lots of Noise

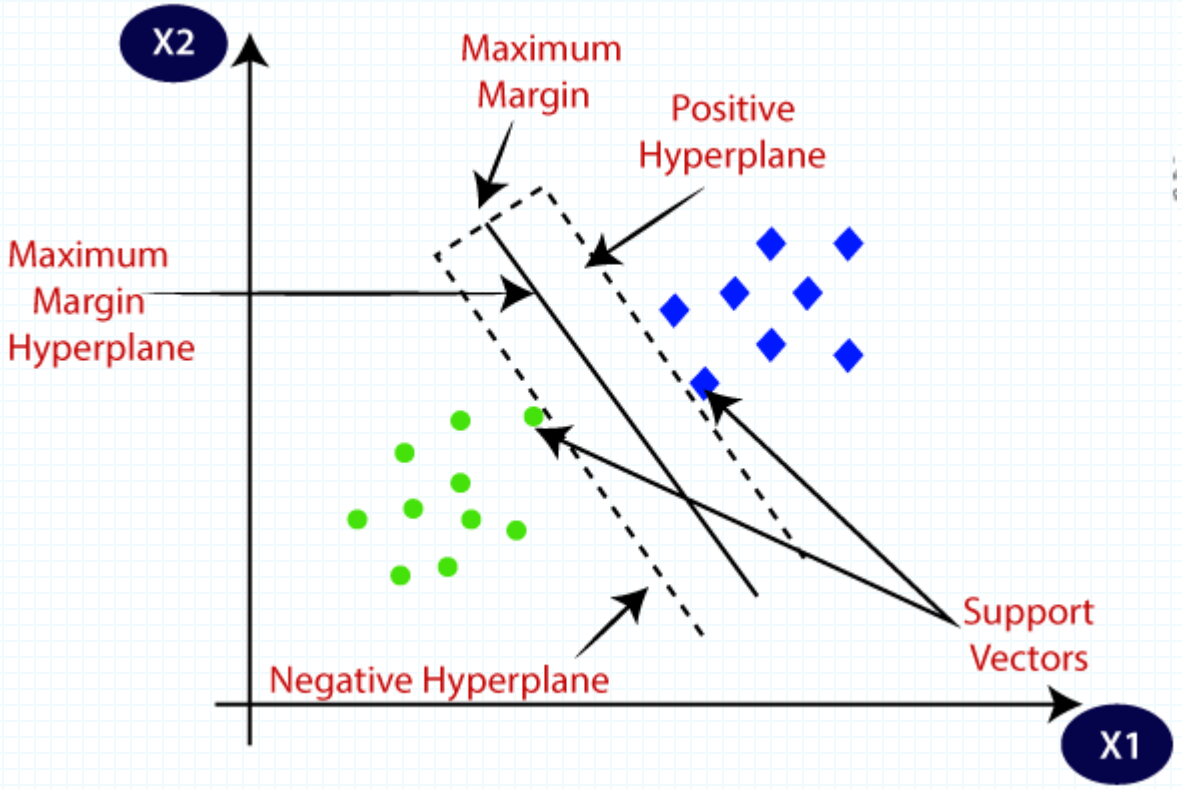


Maximizing the Margin



Terms

- **Support Vectors:**
  - These are the points that are closest to the hyperplane.
  - A separating line will be defined with the help of these data points.
- **Margin:**
  - It is the distance between the hyperplane and the observations closest to the hyperplane (support vectors).
  - In SVM large margin is considered a good margin.
  - There are two types of margins **hard margin** and **soft margin.**

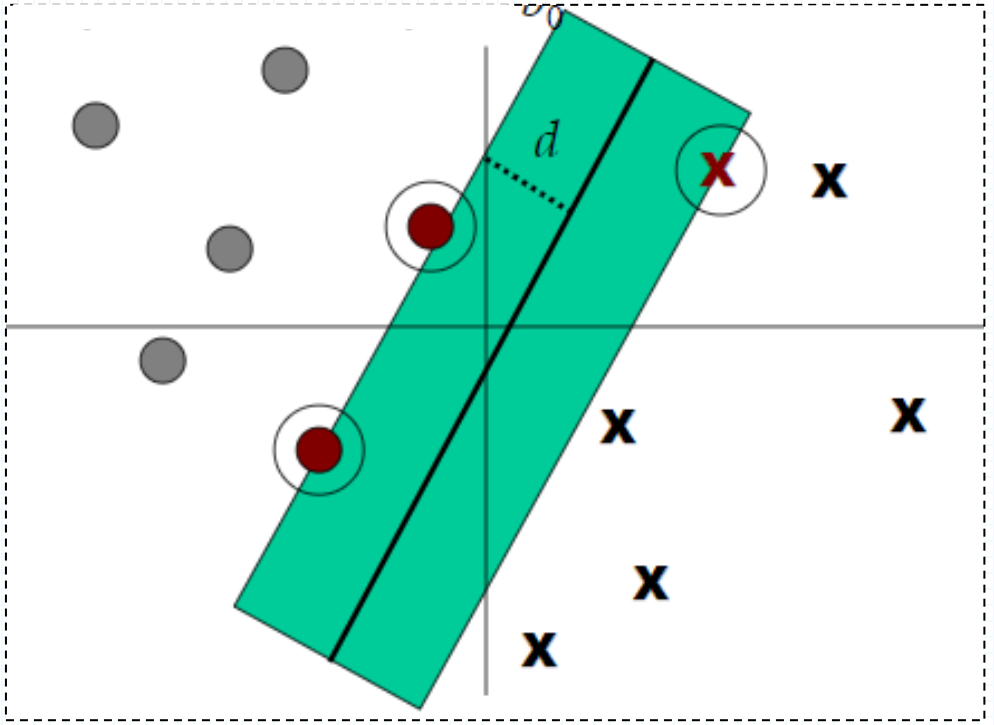


Intuition

- **Our problem:**

Maximizing the shortest distance to the closest positive or negative point.

$$w^* = arg_w max\ [min_n\ d_H(\phi(x_n))]$$

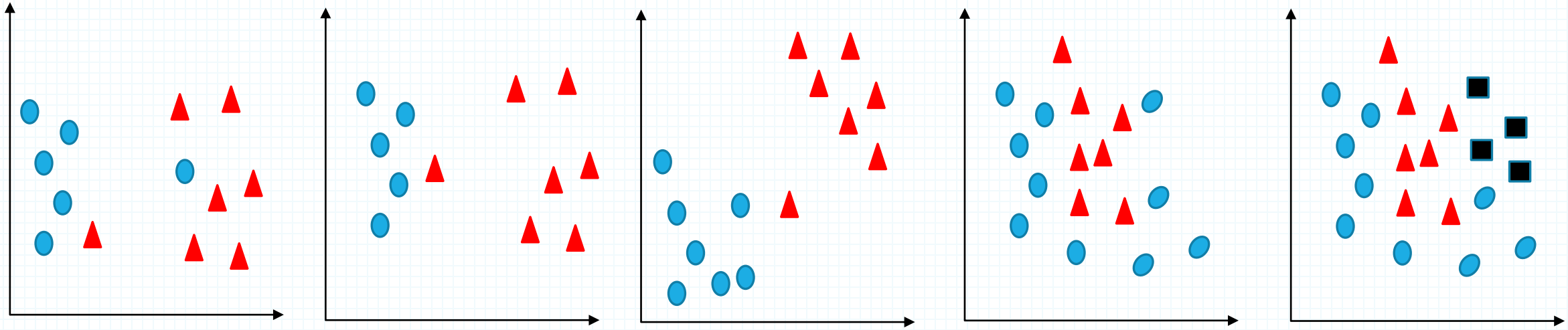


Note that W represents all parameters
i.e. w and b

What if?

What are the problems of the current version for SVM?

(A) (B) (C) (D) (E)



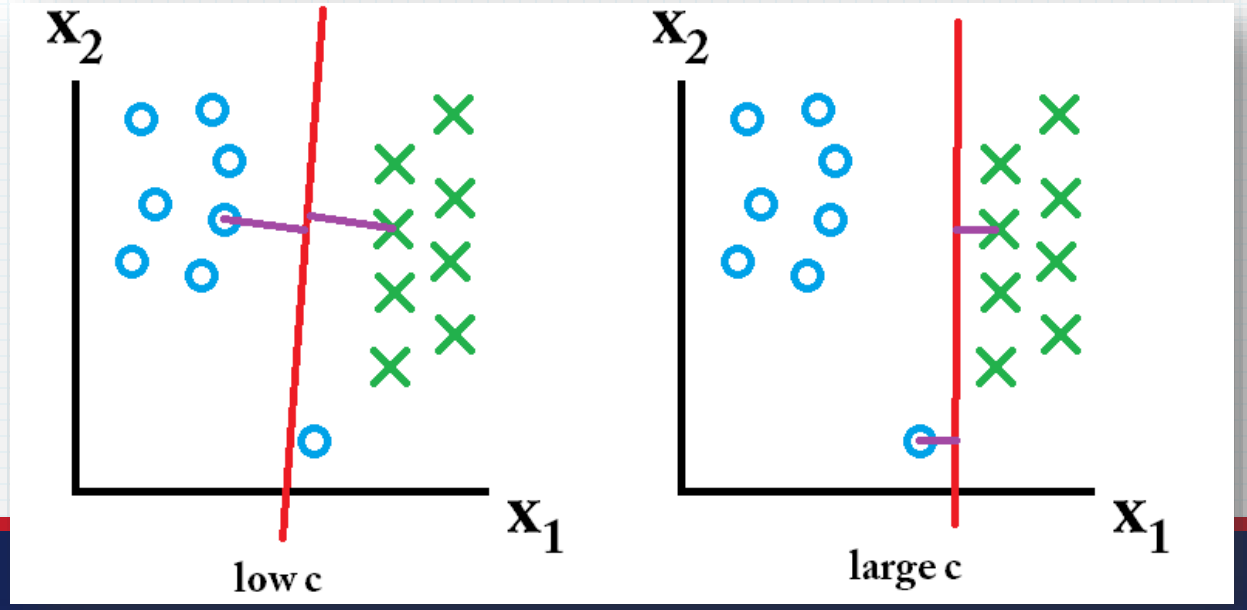
1st Improvement

Soft Margin SVM
(allows few misclassifications)

Soft Margin SVM

- In real-life applications we don't find any dataset which is linearly separable, what we'll find is either an almost linearly separable dataset or a non-linearly separable dataset
- To tackle this problem, we modify SVM to allow few points to be wrongly classified.

SVM Error = Margin Error + C * Classification Error

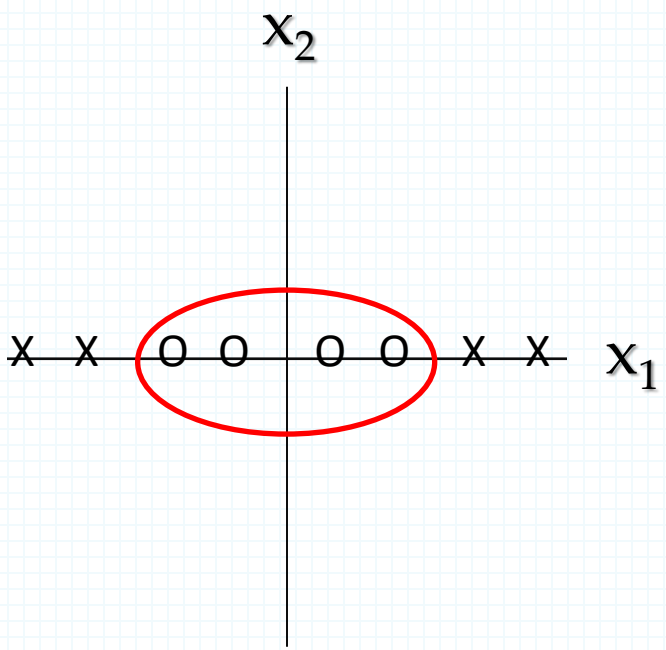


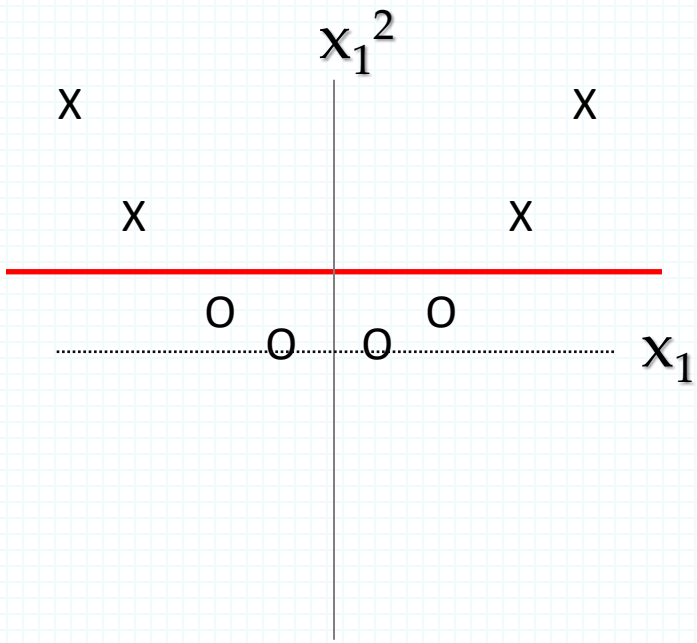
2rd Improvement

How to make a plane curved?

When Linear Separators Fail

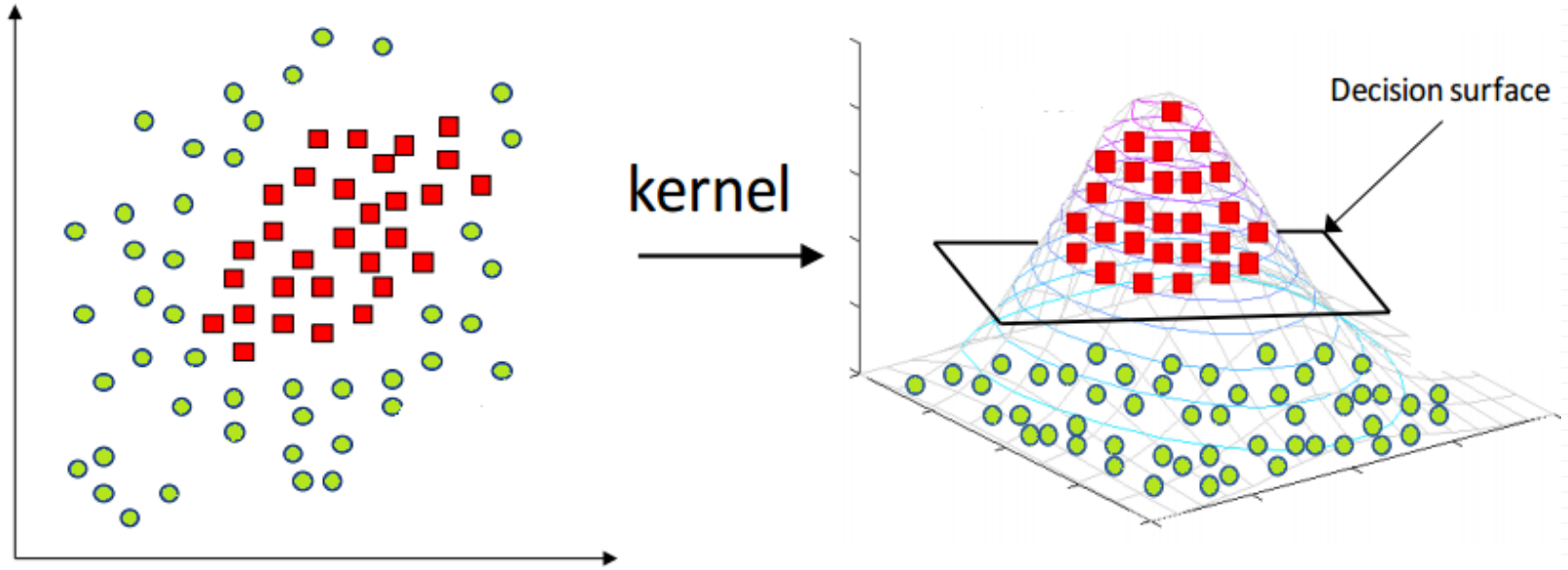
- The most interesting feature of SVM is that it can even work with a non-linear dataset.
- We use “Kernel Trick” which makes it easier to classifies the points.





The Kernel Trick

- try converting this lower dimension space to a higher dimension space using some quadratic functions which will allow us to find a decision boundary that clearly divides the data points.
- These functions which help us do this are called Kernels and which kernel to use is purely determined by hyperparameter tuning.



Kernel functions: Polynomial kernel

- Following is the formula for the polynomial kernel:

$$f(X1, X2) = (X1^T . X2 + 1)^d$$

- Here d is the degree of the polynomial, which we need to specify manually.
- Suppose we have two features X1 and X2 and output variable as Y, so using polynomial kernel we can write it as:

$$X1^T . X2 = \begin{bmatrix} X1 \\ X2 \end{bmatrix} . [X1 \quad X2]$$

$$= \begin{bmatrix} X1^2 & X1.X2 \\ X1.X2 & X2^2 \end{bmatrix}$$

- So we basically need to find $X1^2$, $X2^2$ and X1.X2, and now we can see that 2 dimensions got converted into 5 dimensions.

Other commonly used kernels

- linear: $\langle x, x' \rangle$.
- polynomial: $(\gamma \langle x, x' \rangle + r)^d$, where d is specified by parameter `degree`, r by `coef0`.
- rbf: $\exp(-\gamma \|x - x'\|^2)$, where γ is specified by parameter `gamma`, must be greater than 0.
- sigmoid $\tanh(\gamma \langle x, x' \rangle + r)$, where r is specified by `coef0`.

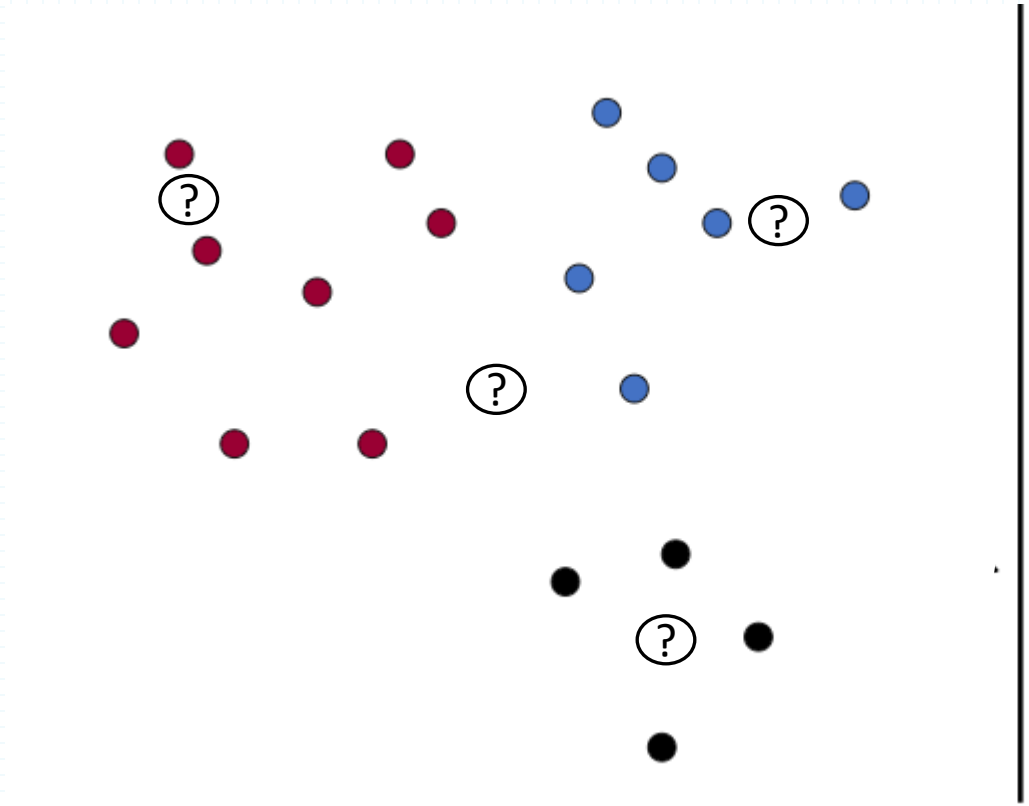
How to choose the right Kernel?

- Choosing a kernel totally depends on what kind of dataset are you working on.
- You can start with a hypothesis that your data is linearly separable and choose a linear kernel function.
- Once you have established it is a problem requiring a non-linear model, the **Radial Basis Function kernel** makes a good default kernel

K-nearest-neighbor

Vector Space Representation

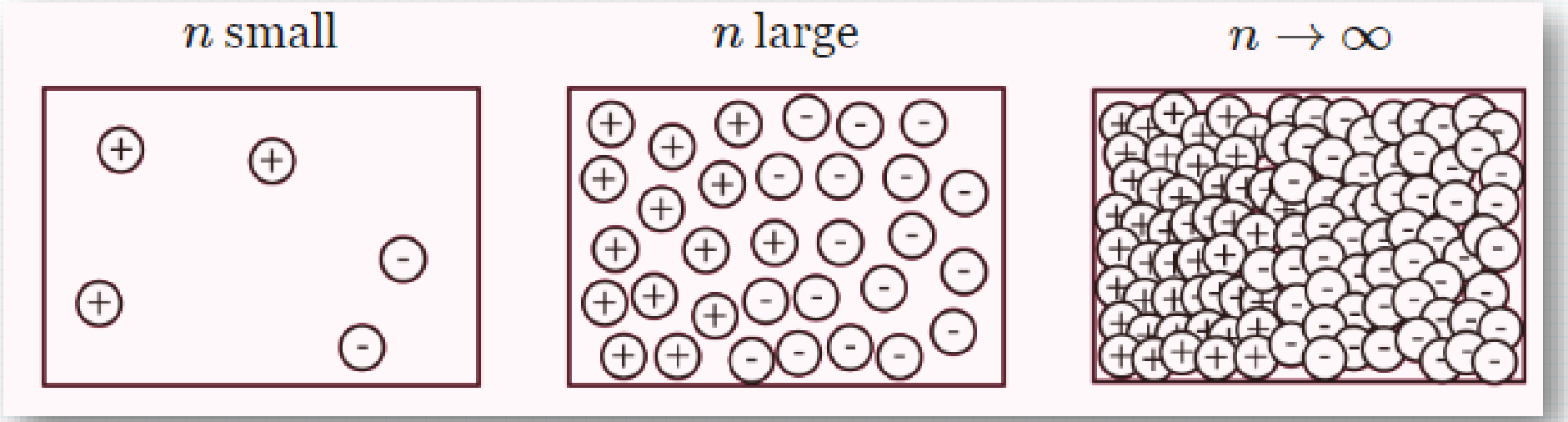
- Each instance is represented as a vector of features.



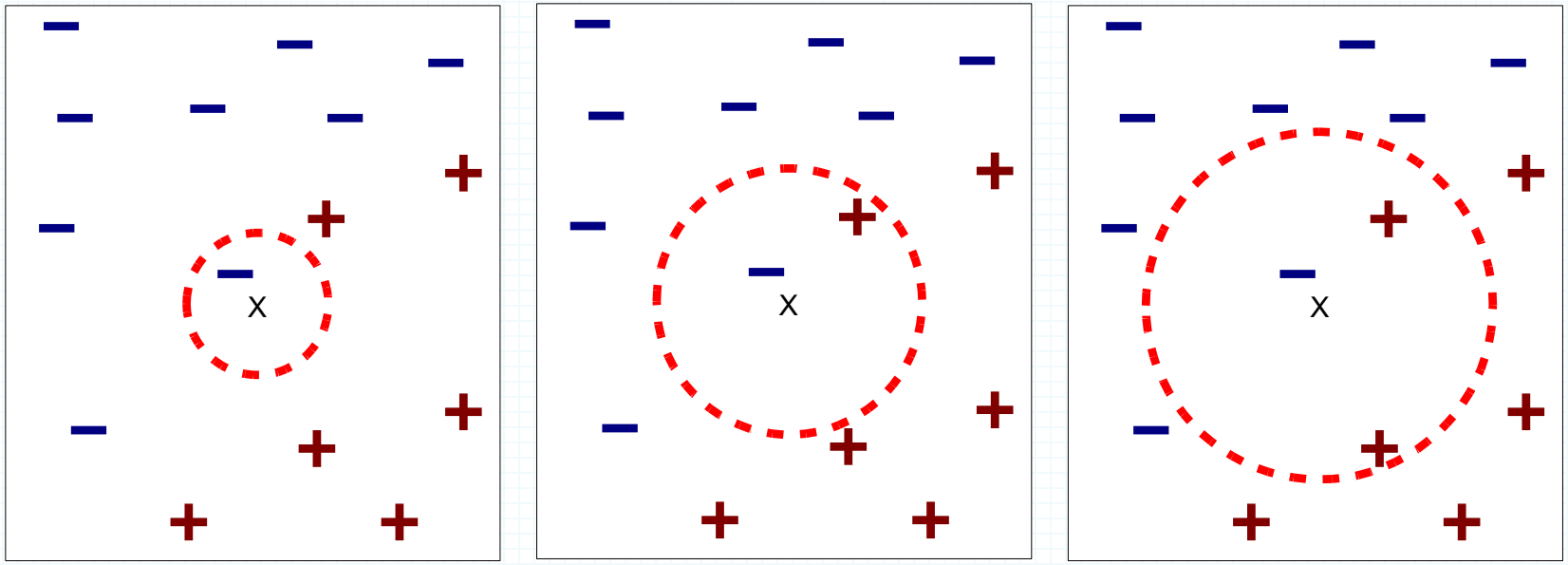
k-NN classification rule is to assign to a test sample the majority category label of its k nearest training samples

Bayes optimal classifier and NN

- Assume (and this is almost never the case) you knew P(y|x), then you would simply predict the most likely label.
- Although the Bayes optimal classifier is as good as it gets, it still can make mistakes.



Definition of Nearest Neighbor

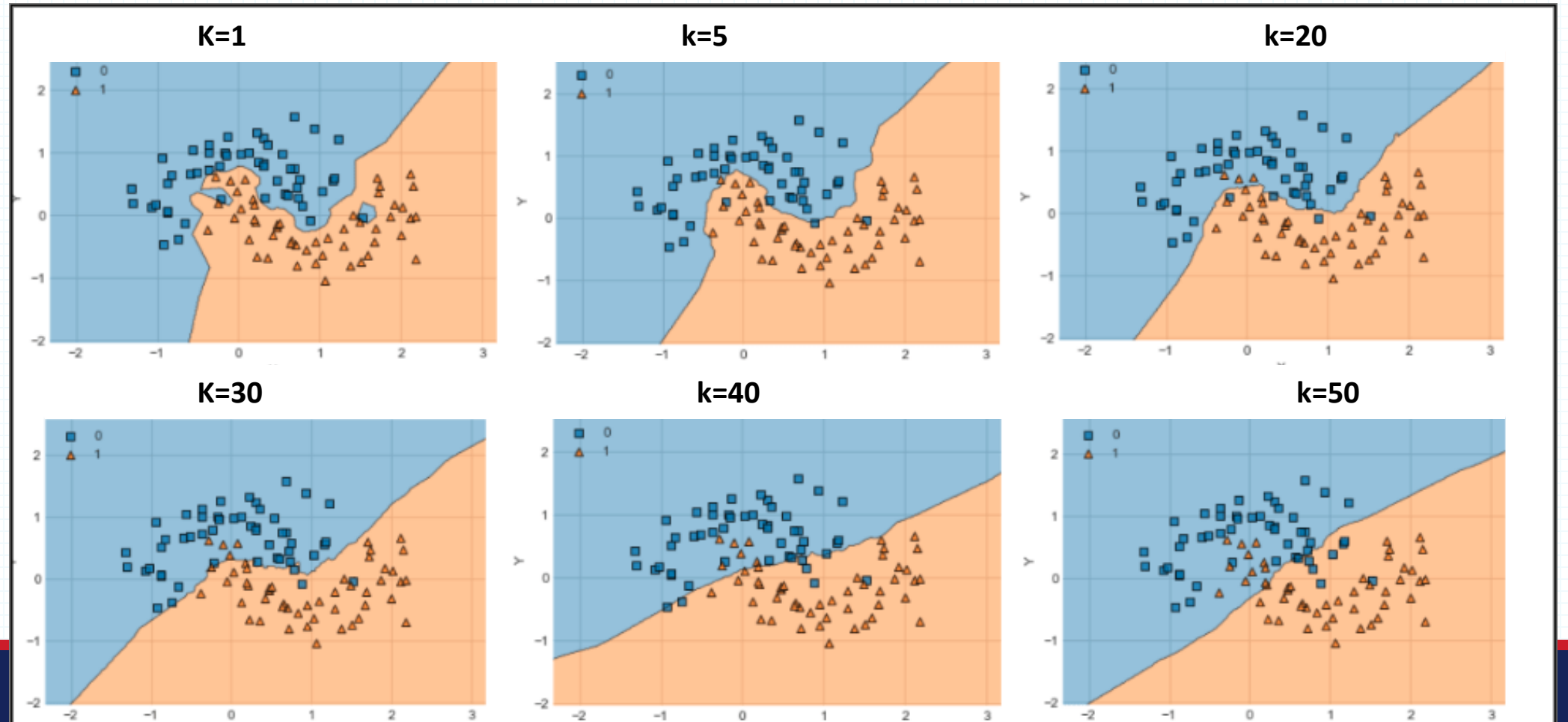


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Form 1-NN to K-NN

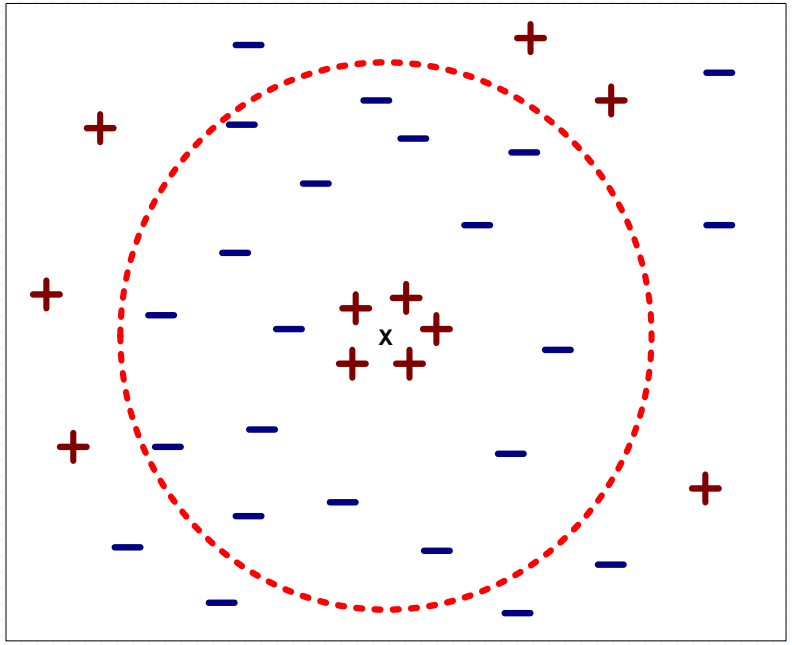
- If you increase k, the areas predicting each class will be more "smoothed“.
- if you increase K too much, you will have an underfitted model.



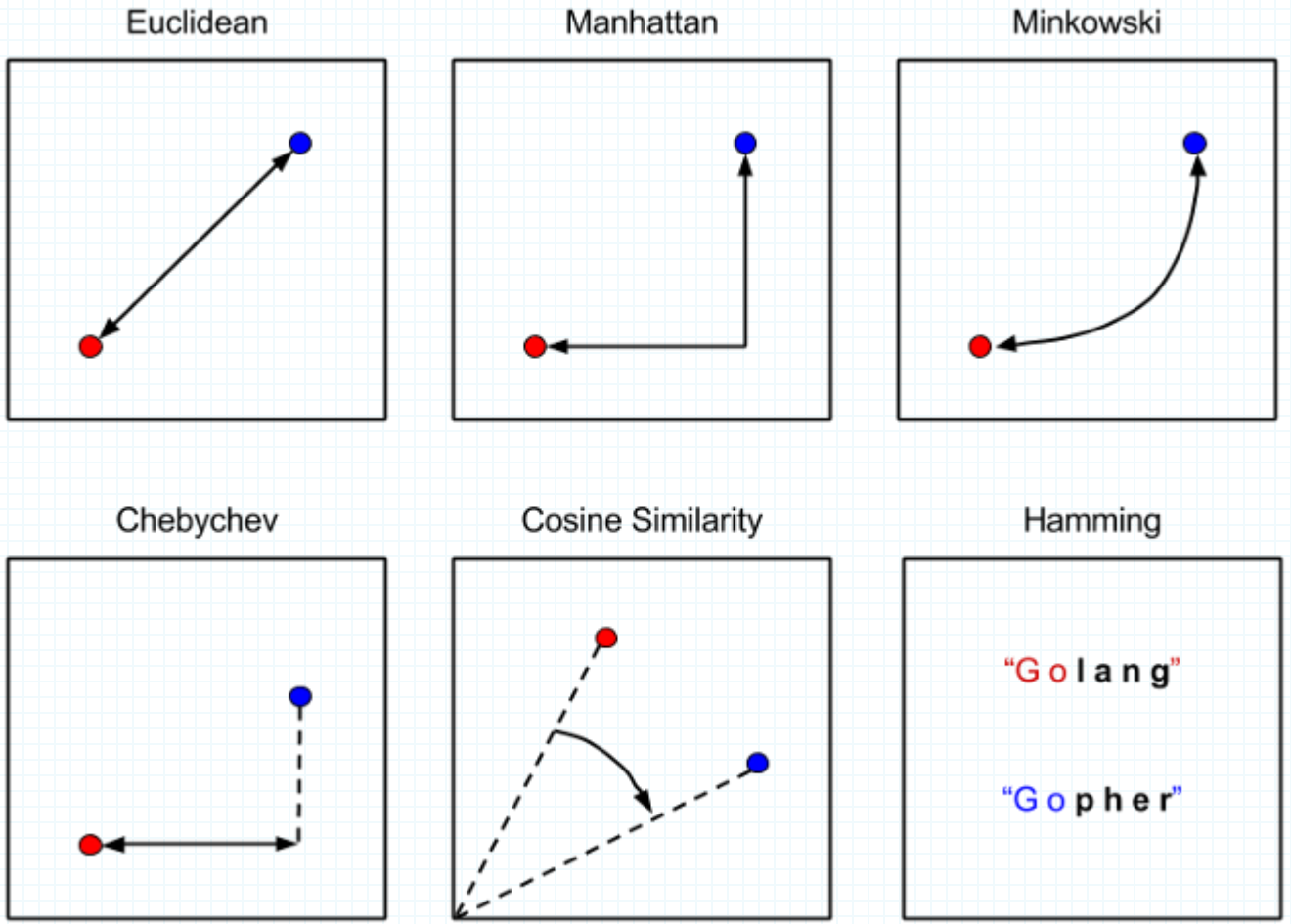
(1) Value of K

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes

Rule of thumb:
K = sqrt(N)
N: number of training points



(2) Distance Metrics



NEXT WEEK… Why Deep Learning?

- What kind of discussion boundaries can we learn using:
  - Decision Trees,
  - KNN,
  - SVM
- Learning highly non-linear functions