SEENG 44242 -Machine Learning

Classification

sunny

windy

rainy

cloudy

Goal: Predict Category

Data for Classification

Input Variables

Target Variables

Temperature	Humidity	Wind Speed	Weather
79	48	2.7	Sunny
60	80	3.8	Rainy
68	45	17.9	Windy
57	77	4.2	Cloudy

Classification is Supervised

Target Label Output Class
Class Variables Category

Temperature	Humidity	Wind Speed	Weather
79	48	2.7	Sunny
60	80	3.8	Rainy
68	45	17.9	Windy
57	77	4.2	Cloudy

Types of Classification

- Binary Classification – Target has two values
  - Will it rain tomorrow or not?
  - Is this transaction legitimate or fraudulent?
- Multiclass Classification – Target has more than two values
  - What type of product will this customer buy?
  - Is this tweet positive, negative, or neutral

What is a Machine Learning Model?

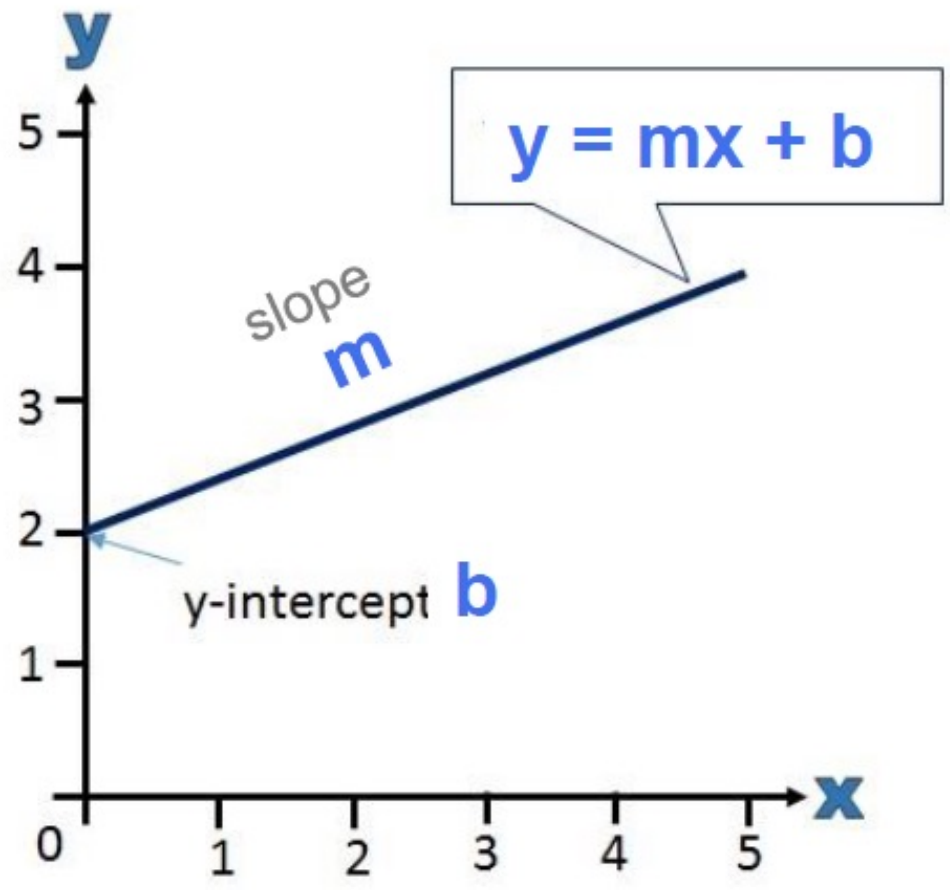
- A mathematical model with parameters that map input to output

parameters

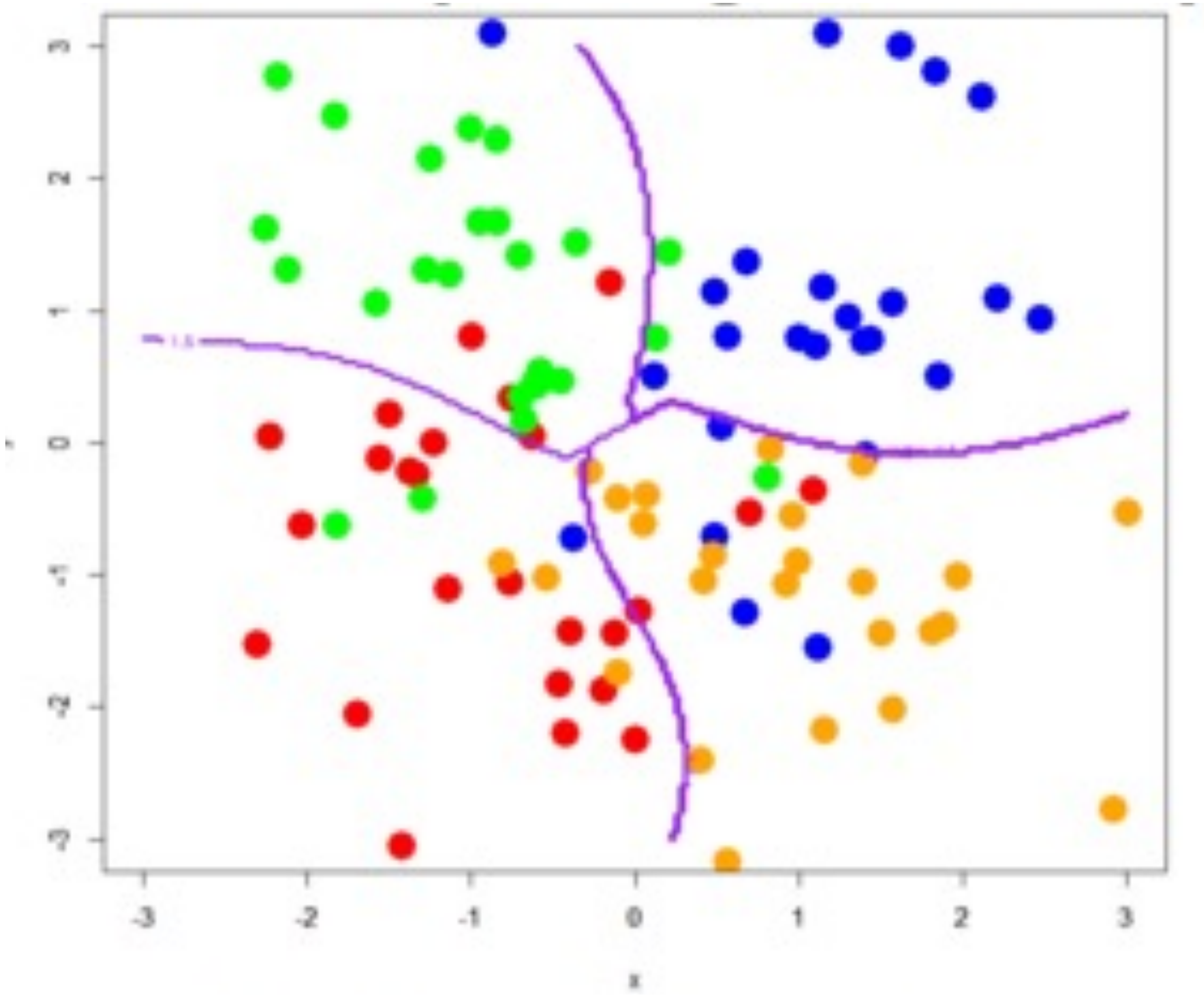
Input variables: x → Model: $y = f(ax + b)$ → Output: y

function mapping input to output

Example of a Model

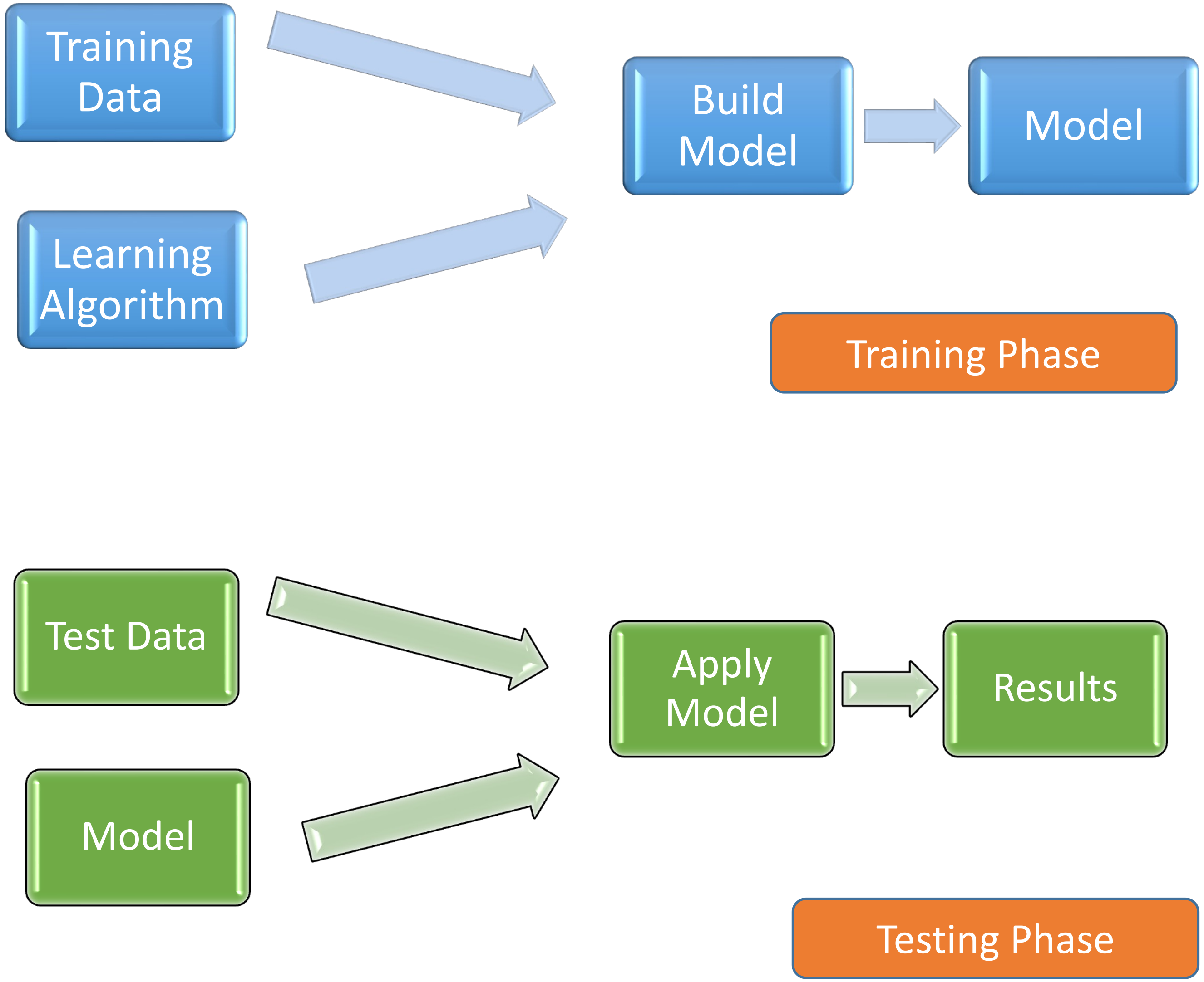


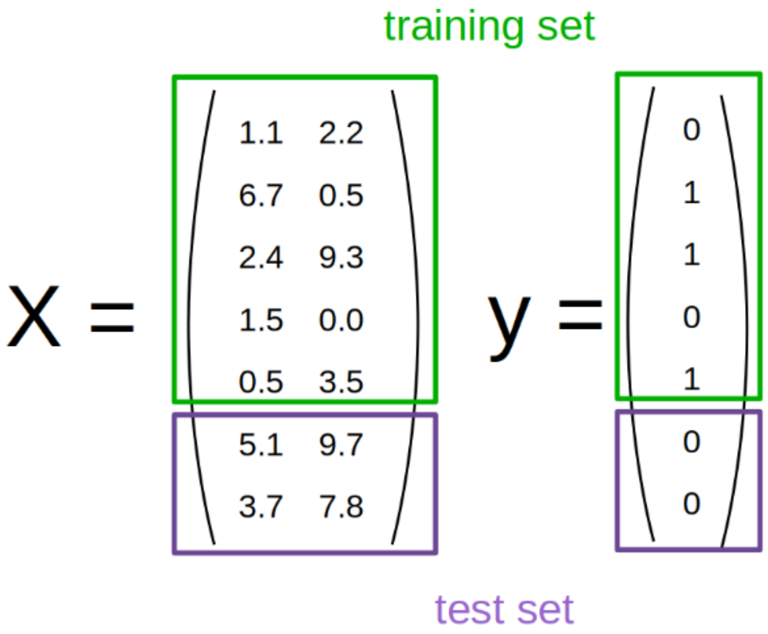
Building Classification Model



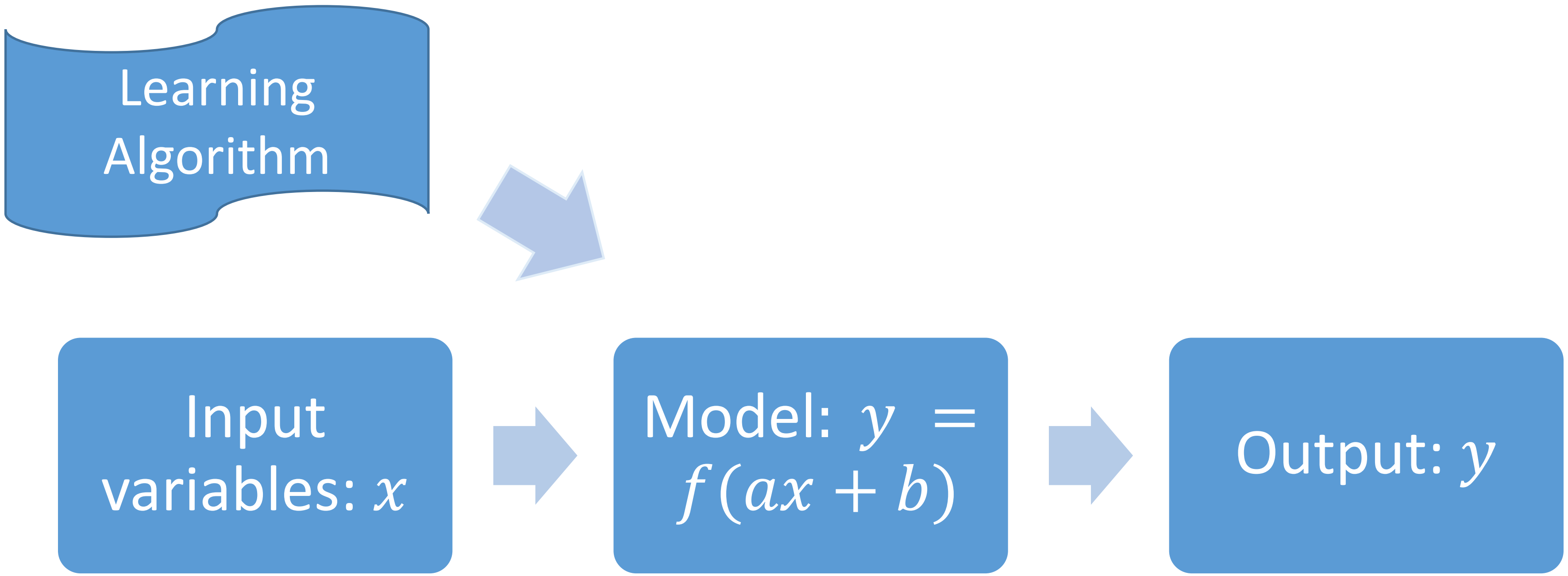
Building vs. Applying Model

- Training Phase
  - Adjust model parameters
  - Use training data
- Testing Phase
  - Apply learned model
  - Use new data


Training Data
Learning Algorithm
Build Model
Model
Training Phase
Test Data
Model
Apply Model
Results
Testing Phase

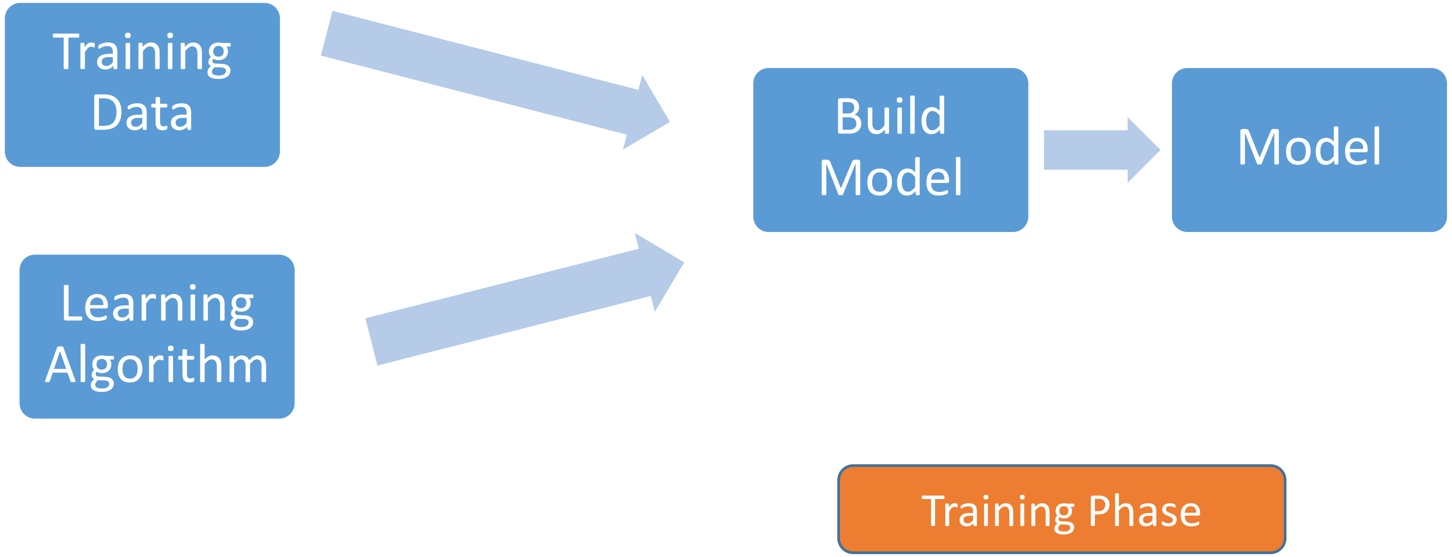

training set
X =
1.1 2.2
6.7 0.5
2.4 9.3
1.5 0.0
0.5 3.5
5.1 9.7
3.7 7.8
y =
0
1
1
0
1
0
0
test set

Building Classification Algorithm



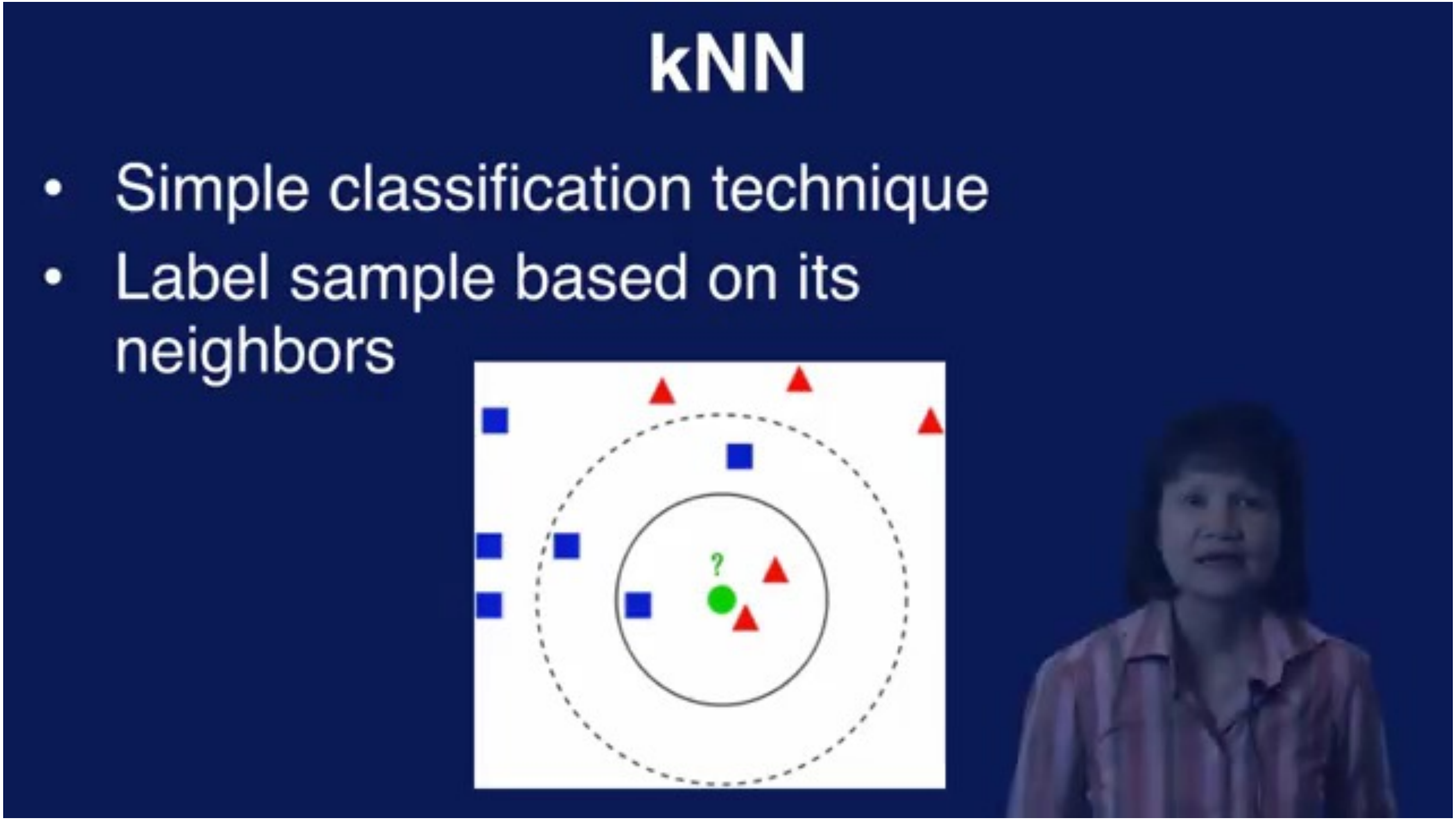
Classification Algorithms

- **Task:** Predict category from input variables
- **Goal:** Match model outputs to targets (desired outputs)
- Learning algorithm used to adjust model's parameters



- Common Algorithms
  - kNN
  - Decision tree
  - Naïve bayes
  - Logistic regression
  - SVM

K-NN Algorithm

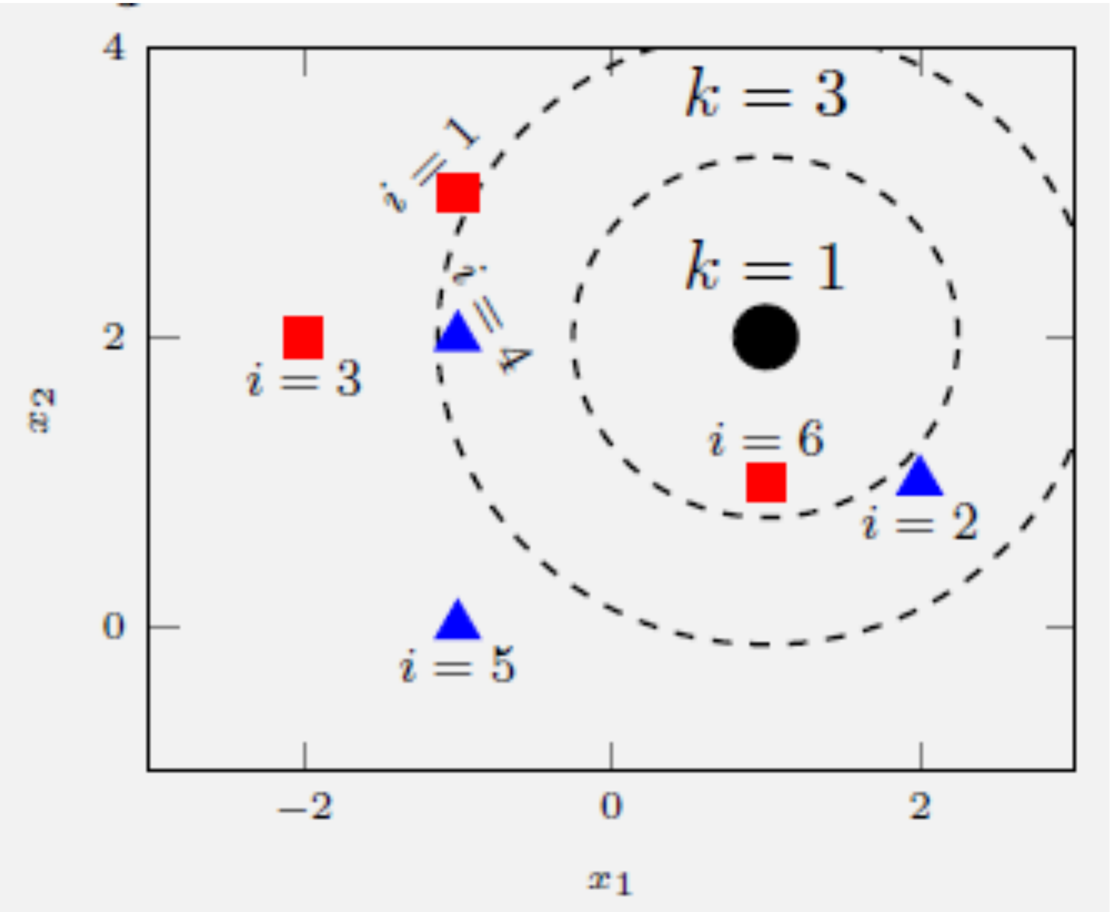

kNN
• Simple classification technique
• Label sample based on its neighbors
?

Example: Predicting colors with k-NN

i	x_1	x_2	y
1	-1	3	Red
2	2	1	Blue
3	-2	2	Red
4	-1	2	Blue
5	-1	0	Blue
6	1	1	Red

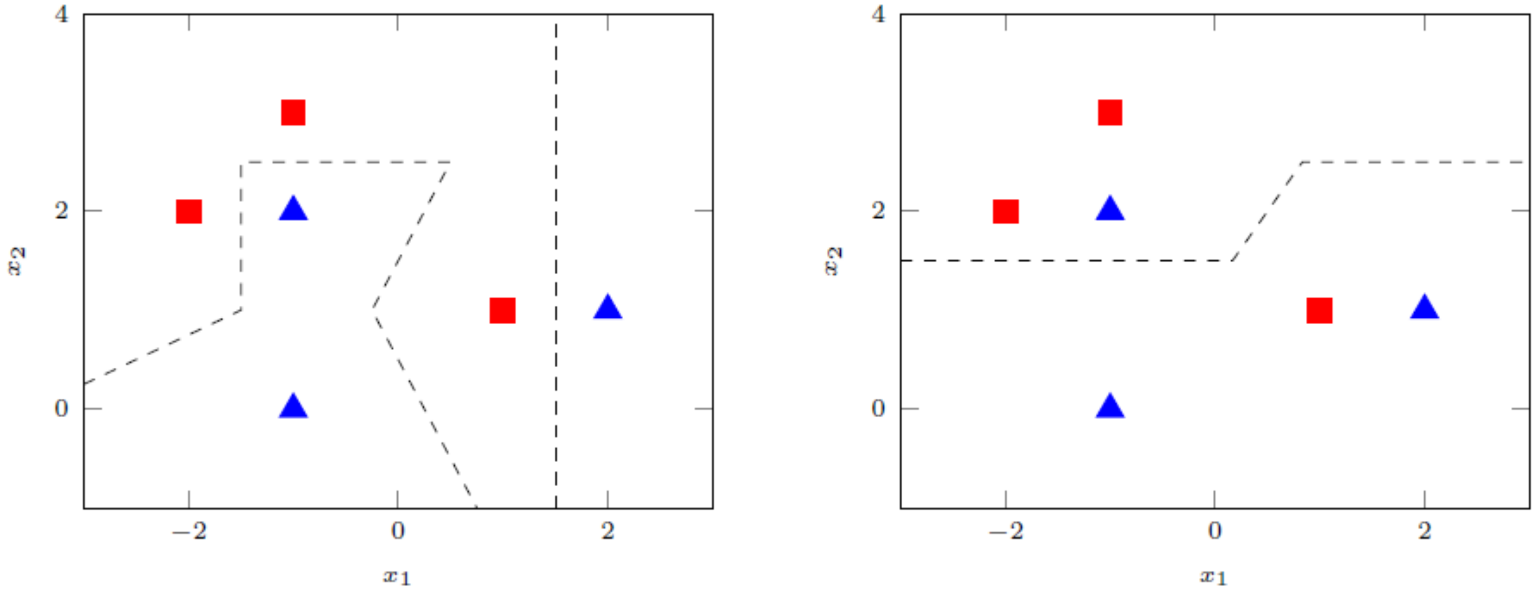
Example: Predicting colors with k-NN

i	x_1	x_2	y
1	-1	3	Red
2	2	1	Blue
3	-2	2	Red
4	-1	2	Blue
5	-1	0	Blue
6	1	1	Red



Example: Predicting colors with k-NN

i	x_1	x_2	y
1	-1	3	Red
2	2	1	Blue
3	-2	2	Red
4	-1	2	Blue
5	-1	0	Blue
6	1	1	Red



(a) $k = 1$ (b) $k = 3$

Decision boundaries

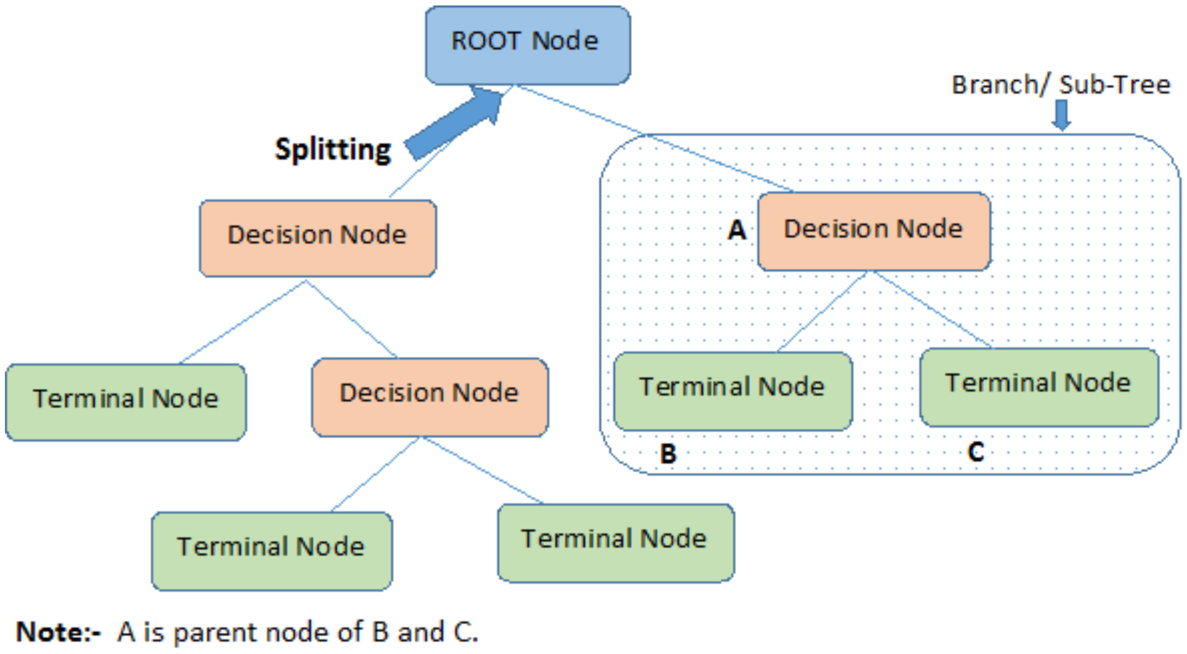
Decision Trees

Overview

- Tree-based learning algorithms are a broad and popular family of related **nonparametric**, **supervised** methods for both **classification and regression**.
- Tree-based methods **divides the feature space into different regions**.
- The **rules to divide** the feature space can be **summarized in a tree**, and hence these methods are known as decision trees.
- Trees can be used for **both regression and classification** problems.

Important Terminology related to Decision Trees

- **Root Node**: It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting**: It is a process of dividing a node into two or more sub-nodes.
- **Decision Node**: When a node splits into further sub-nodes, then it is called decision node.
- **Leaf/ Terminal Node**: Nodes do not split is called Leaf or Terminal node.
- **Pruning**: When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
- **Branch / Sub-Tree**: A sub section of entire tree is called branch or sub-tree.
- **Parent and Child Node**: A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.



- The result looks vaguely like an upside-down tree, with the first decision rule at the top and subsequent decision rules spreading out below.

Types of Decision Trees

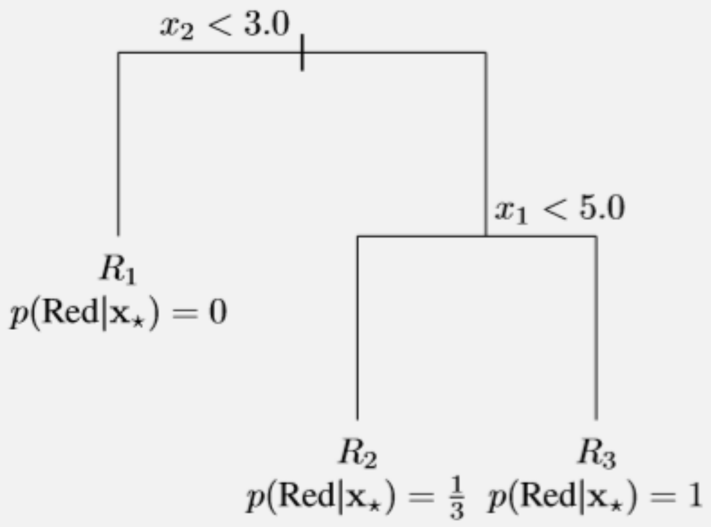
- **Categorical Variable Decision Tree**: Decision Tree which has categorical target variable then it is called as Categorical Variable Decision Tree.
- **Continuous Variable Decision Tree**: Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

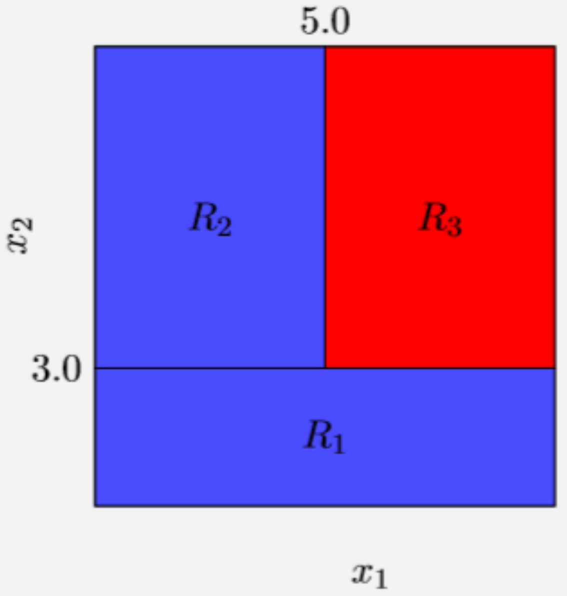
Training a Decision Tree Classifier

- In a classification tree the function $p(y|x)$ is modeled with a series of rules on the input variables $x_1, \dots, x_m$
- These rules can be represented by a binary tree.
- This tree effectively divides the input space into multiple regions and in each region a constant value for the predicted class probability $p(y|x)$ is assigned.

Example 4.2: Predicting colors with a classification tree

Consider a problem with two input variables x_1 and x_2 and one quantitative output y, the color red or blue. A classification tree for this problem can look like the one below. To use this tree to classify a new point $\mathbf{x}_\star = [x_{\star 1}, x_{\star 2}]^\mathsf{T}$ we will start at the top and work the way down until we reach the end of a branch. Each such final branch corresponds to a constant predicted class probability $p(\text{Red}|\mathbf{x}_\star)$.



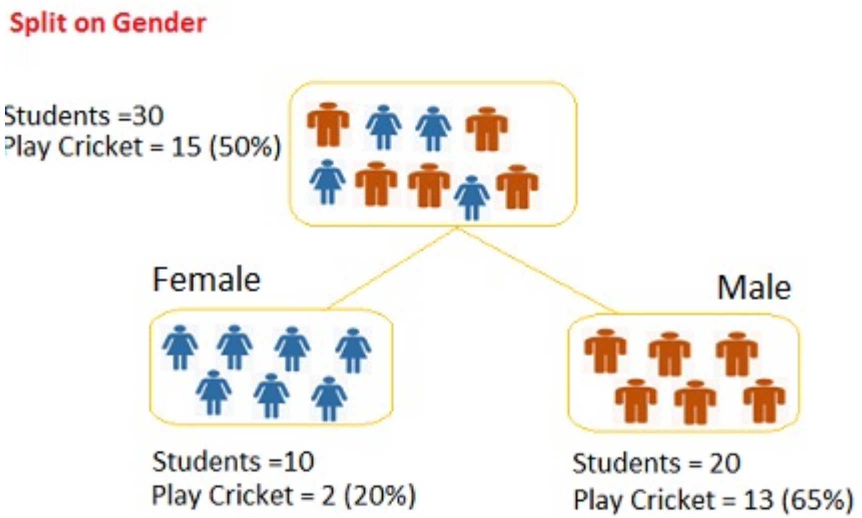


A pseudo code for classifying a test input

```
if x_2 < 3.0 then
    return p(Red|x)=0
else
    if x_1 < 5.0 then
       return p(Red|x)=1/3
    else
        return p(Red|x)=1
    end
end
```

Deciding the split

- Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes.
- The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that purity of the node increases with respect to the target variable.



Splitting Criteria

$$\boldsymbol{minimize} \sum_{m=1}^{M} n_m Q_m(T),$$

where $Q_m(T) =$

$$-\sum_{k=1}^{K} \hat{\pi}_{mk} \log \hat{\pi}_{mk} \quad \textbf{(Entropy)}$$

$$\sum_{k=1}^{K} \hat{\pi}_{mk}(1 - \hat{\pi}_{mk}) \quad \textbf{(Gini Index)}$$

$$1 - \max_{k} \hat{\pi}_{mk} \quad \textbf{(Misclassification Rate)}$$

Where $\hat{\pi}_{mk} = \frac{1}{n_m} \sum_{i:x_i \in R_m} \mathbb{I}\{y_i = k\}$, M is the total number of regions (leaf nodes) in the tree and $\mathbb{I}\{y_i = k\} = 1$ if $x_i \in R_m$ and 0 otherwise. Since probabilities should sum up to one in each region, $\sum_{k=1}^{K} \hat{\pi}_{mk} = 1$.

- Finding the best tree T that minimizes the above equation is, unfortunately, a **combinatorial problem and hence computationally infeasible**.
- Instead, we choose a **greedy algorithm** known as ***recursive binary splitting***
  - **minimization for each node split** separately (instead of optimizing the entire tree at the same time).
  - This approach starts in the top of the tree and successively splits the input where each split divides one branch into two new branches.
- This approach is **greedy** since it builds the tree by introducing only one split at a time, without having the full tree 'in mind'.
  - For each decision node, we scan through the finite number of possible splits and pick the input variable x_j and the cut point s that provides the minimum.
  - After that, we repeat the process to create new splits by finding the best values (x_j, s) for each of the new branches.
  - We continue the process until some stopping criteria is reached, for example until no region contains more than five training data points.

Questions

1. Define the following terms:
   - Depth of a node
   - Depth of a decision tree
   - Size of a decision tree
2. Describe the steps of constructing a decision tree.
3. What is an induction algorithm? Why is it referred to as a greedy algorithm?
4. How does the decision tree decide the best split for a given node? What factors are considered in this process?
5. What are the stopping criterions used when splitting a node?
6. Discuss the advantages and disadvantages of using a decision tree for a classification task.

Questions Ctd..

7. The following dataset will be used to learn a decision tree for predicting whether the outcome $\boldsymbol{Y}$ is 'Yes' or 'No' based on three categorical predictor variables, $\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3}$.

•

$\boldsymbol{X_1}$	$\boldsymbol{X_2}$	$\boldsymbol{X_3}$	$\boldsymbol{Y}$
C	B	1	Yes
D	B	1	Yes
D	W	1	Yes
D	W	2	Yes
C	B	2	Yes
D	B	2	No
D	G	2	No
C	U	2	No
C	B	3	No
C	W	3	No
D	W	3	No

Based on the Entropy criteria, which attribute is the best for the root of the decision tree? Clearly show all the steps to find the root node.

Naïve Bayes

Overview

- It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors.
- In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- Is easy to build and particularly useful for very large data sets.
- Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Advantages of NB

- It is easy and fast to predict the class of test data set. It also performs well in multi-class prediction
- When the assumption of independence holds, a Naïve Bayes classifier performs better compared to other models like logistic regression and you need less training data.
- It performs well in the case of categorical input variables compared to the numerical variable(s). For numerical variables, normal distribution is assumed (bell curve, which is a strong assumption).

Disadvantages of NB

- Naive Bayes is also known as a bad estimator, so the probability outputs are not to be taken too seriously.
- In real life, it is almost impossible that we get a set of predictors which are completely independent

Applications of NB

- **Real-time Prediction:** Naive Bayes is an eager learning classifier and it is fast. Thus, it could be used for making predictions in real-time.
- **Multi-class Prediction:** This algorithm is also well known for multi-class prediction feature. Here we can predict the probability of multiple classes of the target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naïve Bayes classifiers mostly used in text classification (due to better results in multi-class problems and independence rule) have a higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naïve Bayes Classifier and Collaborative Filtering together build a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

Question

8 . Using the Bayes classifier, classify a Red, Domestic, SUV for the below dataset:

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Summary

- Building a classification model
- Classification Algorithms
  - K-NN
  - Decision Tree
  - Naïve Bayes