Learning from Examples

Module – 6

Learning from Examples

- Forms of Learning
- Dimensionality reduction
- Regression
- Statistical Methods:
- Naïve Bayes,
- Nearest Neighbor
- Decision Trees
- Random Forest
- Clustering
- Ensemble Learning

What is Machine Learning?

"Learning is any process by which a system improves performance from experience."

- Herbert Simon

Definition by Tom Mitchell (1998):

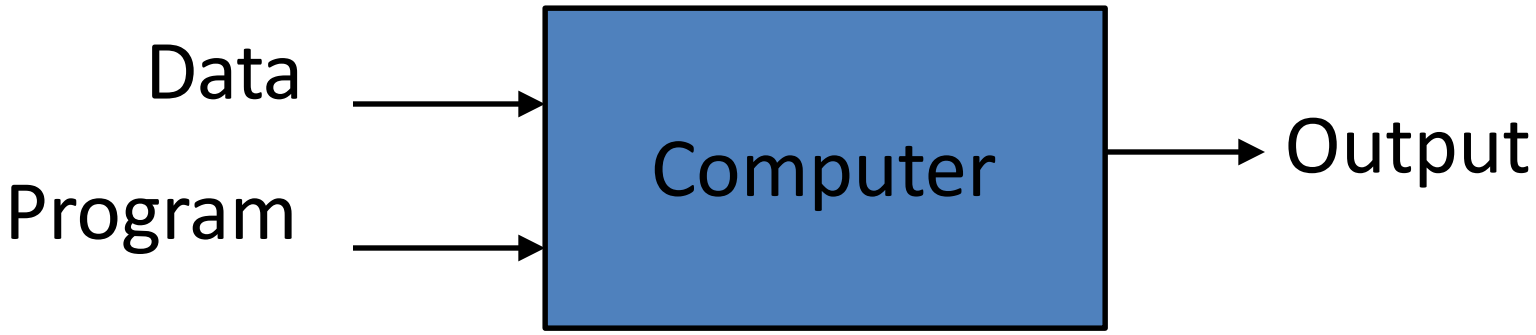
Machine Learning is the study of algorithms that

- improve their performance P
- at some task T
- with experience E.

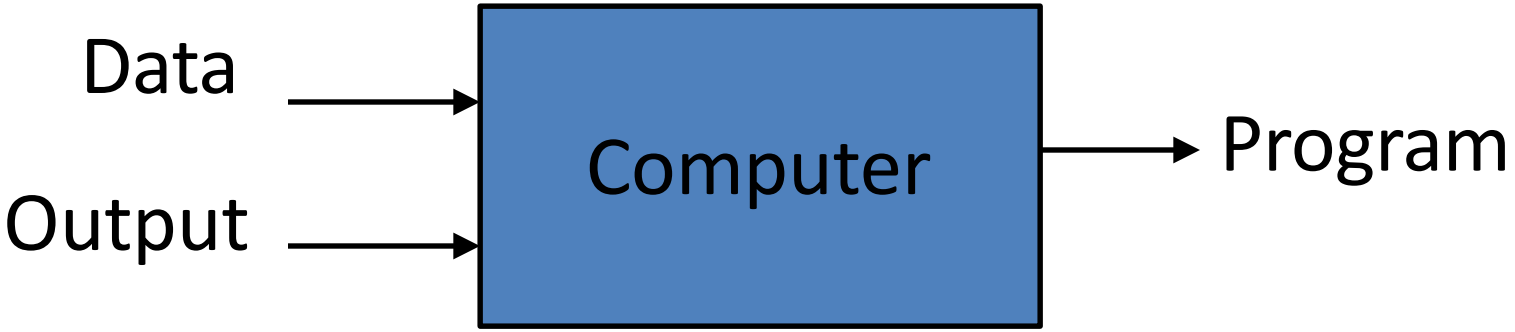
A well-defined learning task is given by <P, T, E>.

Machine Learning

Traditional Programming



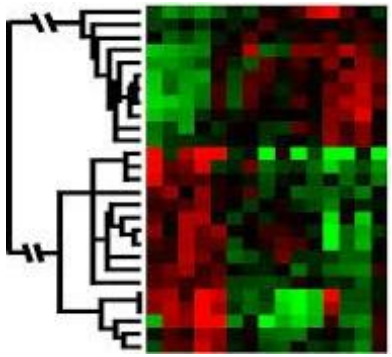
Machine Learning



When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



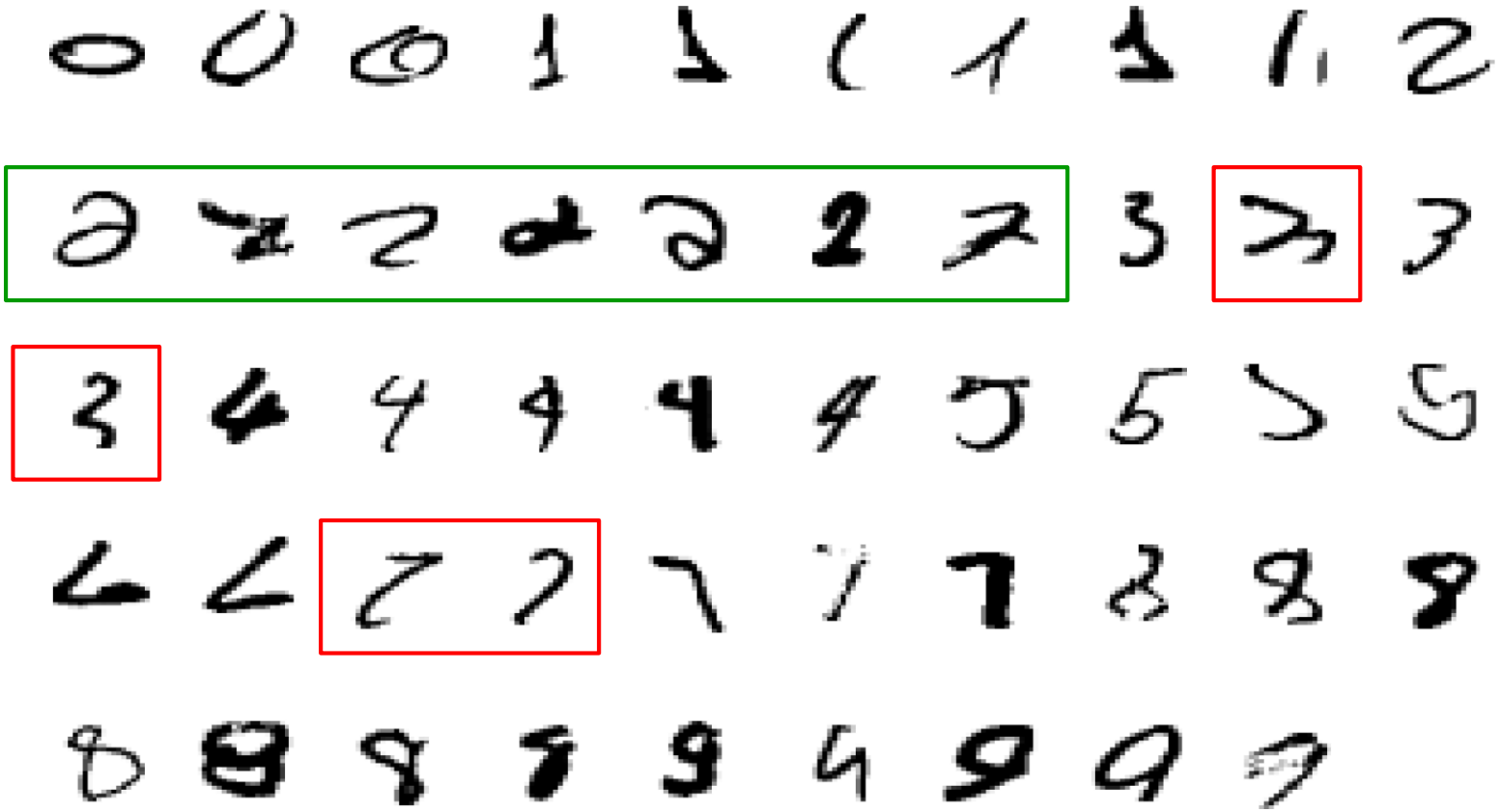
Learning isn't always useful:

- There is no need to "learn" to calculate payroll

Machine Learning

A classic example of a task that requires machine learning:

It is very hard to say what makes a 2



Some more examples of tasks that are best solved by using a learning algorithm

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - Future stock prices or currency exchange rates

Types of Learning

- **Supervised (inductive) learning**
  - Given: training data + desired outputs (labels)
- **Unsupervised learning**
  - Given: training data (without desired outputs)
- **Semi-supervised learning**
  - Given: training data + a few desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions

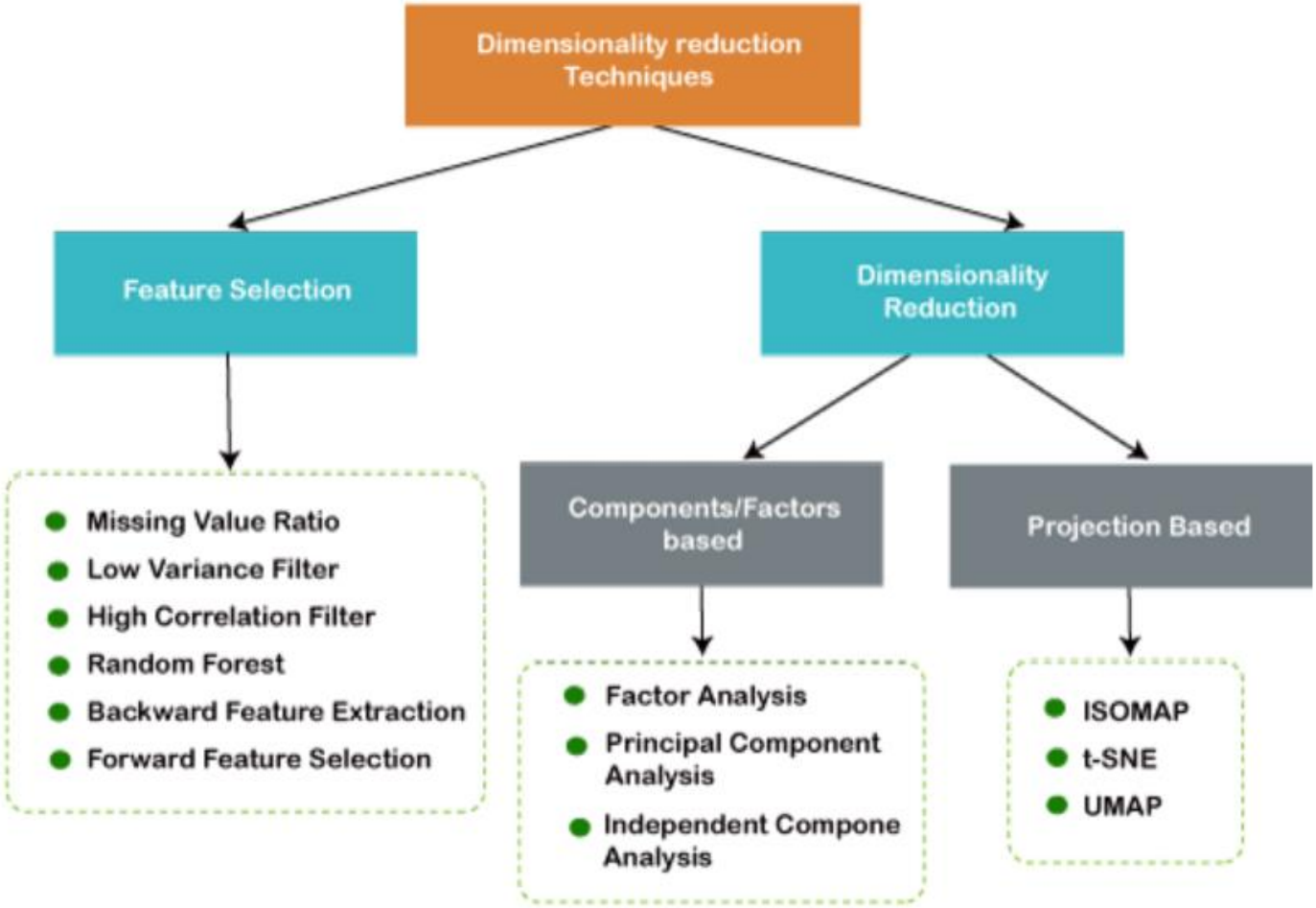
Dimensionality Reduction

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

What is Dimensionality Reduction?

- The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.
- A dataset contains a huge number of input features in various cases, which makes the predictive modeling task more complicated
- Dimensionality reduction technique ; ***"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information."***

Dimensionality Reduction



The Curse of Dimensionality

- Handling the high-dimensional data is very difficult in practice, commonly known as the *curse of dimensionality.*
- If the dimensionality of the input dataset increases, any machine learning algorithm and model becomes more complex.
- As the number of features increases, the number of samples also gets increased proportionally, and the chance of overfitting also increases.
- If the machine learning model is trained on high-dimensional data, it becomes overfitted and results in poor performance.

Regression

- Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables.
- More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed.
- It predicts continuous/real values such as **temperature, age, salary, price,** etc.

Regression Analysis

- **Example:** Suppose there is a marketing company A, who does various advertisement every year and get sales on that. The below list shows the advertisement made βy the company in the last 5 years and the corresponding sales:

Advertisement	Sales
$90	$1000
$120	$1300
$150	$1800
$100	$1200
$130	$1380
$200	??

Now, the company wants to do the advertisement of $200 in the year 2024 **and wants to know the prediction about the sales for this year**

Regression Analysis

- Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables.
- It is mainly used for **prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables**.
- In Regression, we plot a graph between the variables which best fits the given datapoints
- ***"Regression shows a line or curve that passes through all the datapoints on target-predictor graph in such a way that the vertical distance between the datapoints and the regression line is minimum."***

Regression Analysis

Examples:

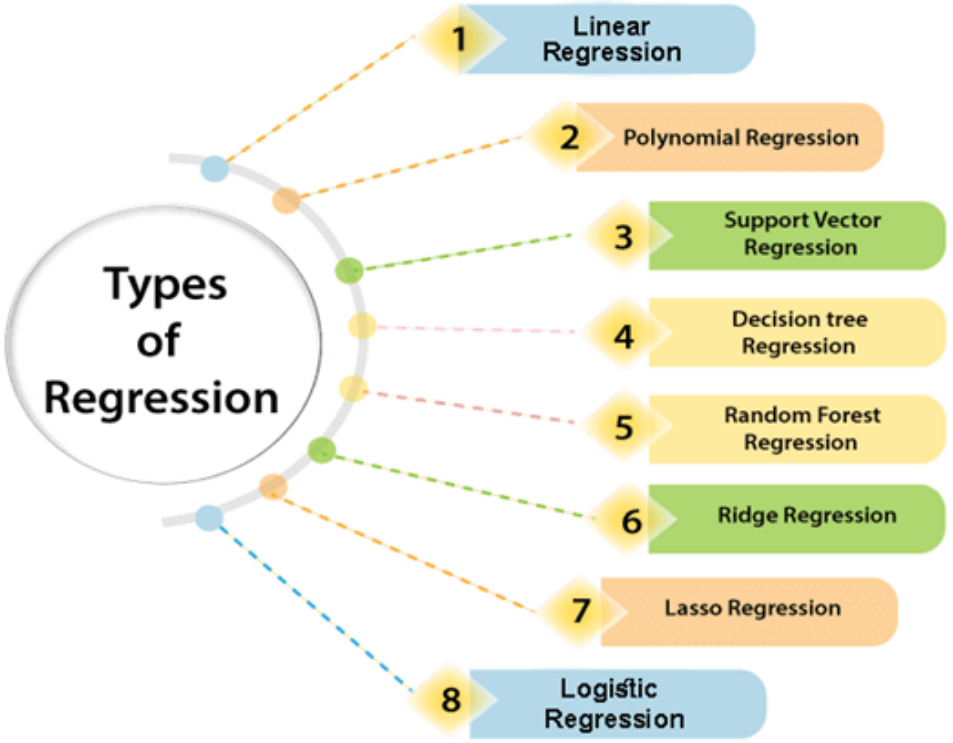
- Prediction of rain using temperature and other factors
- Determining Market trends
- Prediction of road accidents due to rash driving.

Regression Analysis- Technologies

- **Dependent Variable:** The main factor is we want to predict or understand is called the dependent variable. It is also called **target variable**.
- **Independent Variable:** The factors which affect the dependent variables, also called as a **predictor**.
- **Outliers:** Outlier is an observation which contains either very low value or very high value in comparison to other oβserved values. An outlier may hamper the result, so it should βe avoided.
- **Multicollinearity:** If the independent variables are highly correlated with each other than other variables, then such condition is called Multicollinearity. It should not be present in the dataset
- **Underfitting and Overfitting:** If our algorithm works well with the training dataset but not well with test dataset, then such problem is called **Overfitting**. And if our algorithm does not perform well even with training dataset, then such problem is called **underfitting**.

Types of Regression

- Linear Regression
- Logistic Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression
- Ridge Regression
- Lasso Regression:



Linear Regression

- Linear regression is a statistical regression method which is used for predictive analysis.
- It is one of the very simple and easy algorithms which works on regression and shows the relationship between the continuous variables.
- It is used for solving the regression problem in machine learning.
- Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), hence called linear regression.
- If there is only one input variable (x), then such linear regression is called **simple linear regression**. And if there is more than one input variable, then such linear regression is called **multiple linear regression**.

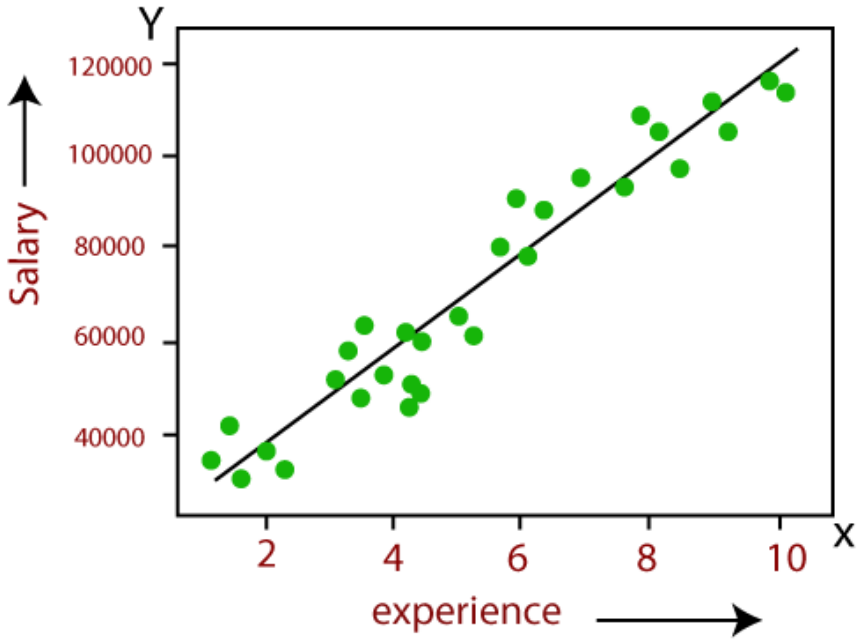
Linear Regression

- Example: predicting the salary of an employee on the basis of **the year of experience**.

- Mathematical equation for Linear regression:

  $Y = a + bX$

- Here, Y = dependent variables (target variables),
  X= Independent variables (predictor variables),
  a and b are the linear coefficients, a is intercept and b is coefficient of X.
- a-----β_0 and b-----β_1



Finding a Linear Regression Line

- Example: let say we want to predict 'y' from 'x' given in following table and let's assume that our regression equation will look like "$y= \beta_0 + \beta_1 *x$"

x	y	Predicted 'y'
1	2	$\beta_0+\beta_1*1$
2	1	$\beta_0+\beta_1*2$
3	3	$\beta_0+\beta_1*3$
4	6	$\beta_0+\beta_1*4$
5	9	$\beta_0+\beta_1*5$
6	11	$\beta_0+\beta_1*6$
7	13	$\beta_0+\beta_1*7$
8	15	$\beta_0+\beta_1*8$
9	17	$\beta_0+\beta_1*9$
10	20	$\beta_0+\beta_1*10$

Where,

Std. Dev. of x	3.02765
Std. Dev. of y	6.617317
Mean of x	5.5
Mean of y	9.7
Correlation between x & y	.989938

Correlation

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Finding a Linear Regression Line

- If we differentiate the Residual Sum of Square (RSS) wrt. β_0 & β_1 and equate the results to zero, we get the following equations as a result:
- β_1 = *Correlation * (Std. Dev. of y/ Std. Dev. of x)*
- β_0 = *Mean(Y) –* β_1 ** Mean(X)*
- Putting values from table 1 into the above equations,
- β_1 = 2.64
- β_0 = -2.2
- Hence, the least regression equation will become –
- **Y = -2.2 + 2.64*x**

X	Y
1	2
2	1
3	3
4	6
5	9

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

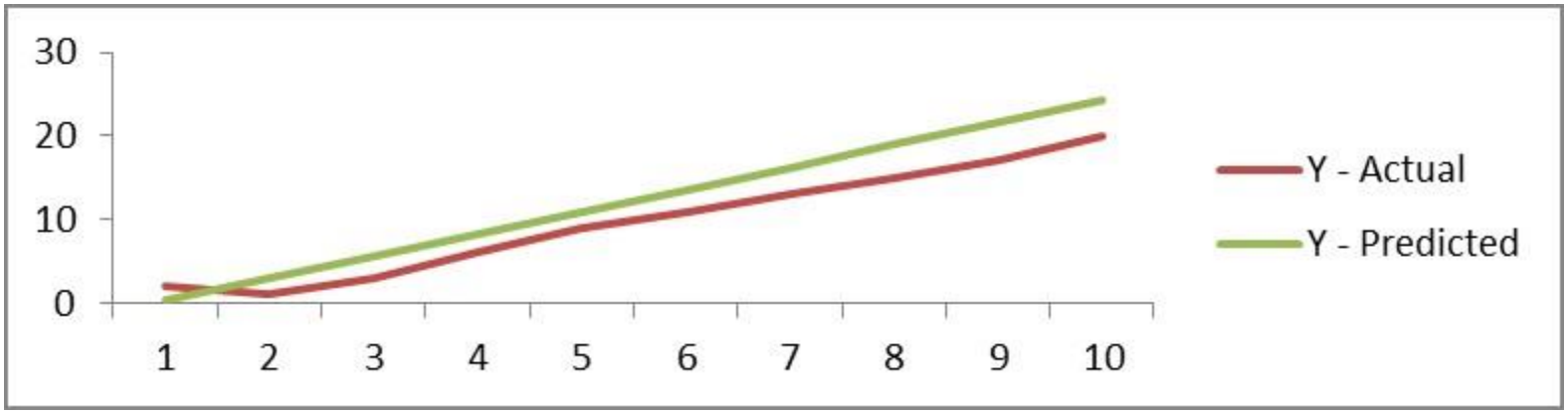
Finding a Linear Regression Line

Let see, how our predictions are looking like using this equation

x	Y - Actual	Y - Predicted
1	2	0.44
2	1	3.08
3	3	5.72
4	6	8.36
5	9	11
6	11	13.64
7	13	16.28
8	15	18.92
9	17	21.56
10	20	24.2

Finding a Linear Regression Line

- Given only 10 data points to fit a line our predictions are not pretty accurate but if we see the correlation between ‘Y-Actual’ & ‘Y – Predicted’ it will turn out to be very high; hence both the series are moving together and here is the graph for visualizing our prediction values:



Logistic Regression

- Logistic regression is another supervised learning algorithm which is used to solve the classification problems.
  In **classification problems**, we have dependent variables in a binary or discrete format such as 0 or 1.
- Logistic regression algorithm works with the categorical variable such as 0 or 1, Yes or No, True or False, Spam or not spam, etc.
- It is a predictive analysis algorithm which works on the concept of probability.
- Logistic regression is a type of regression, but it is different from the linear regression algorithm in the term how they are used.

Logistic Regression

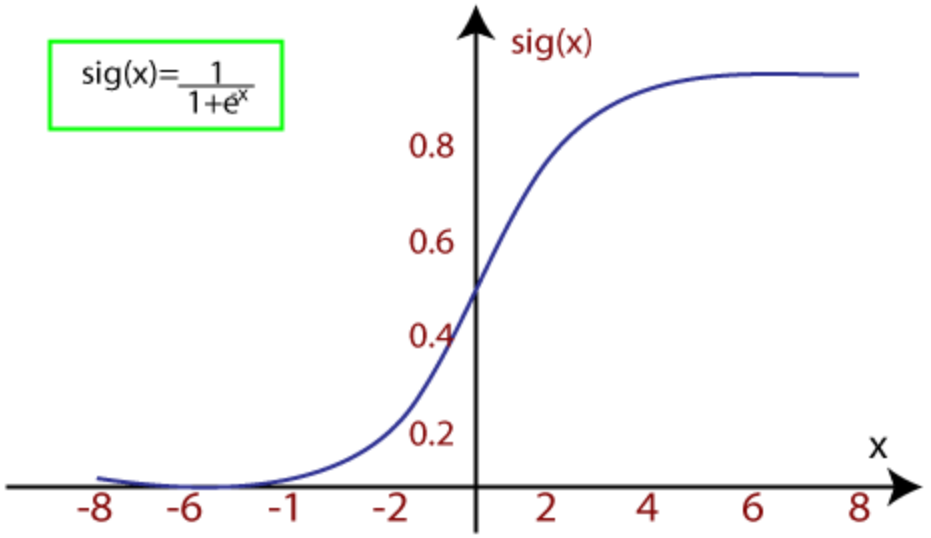
- Logistic regression uses **sigmoid function** or logistic function which is a complex cost function.
- This sigmoid function is used to model the data in logistic regression.
- The function can be represented as:

$$f(x)= \frac{1}{1+e^{-x}}$$

- f(x)= Output between the 0 and 1 value.
- x= input to the function
- e= base of natural logarithm.

Logistic Regression

- When we provide the input values (data) to the function, it gives the S-curve as follows:



- It uses the concept of threshold levels, values above the threshold level are rounded up to 1, and values below the threshold level are rounded up to 0.

Logistic Regression

There are three types of logistic regression:

- **binary(0/1, pass/fail)**
- **Multi(cats, dogs, lions)**
- **Ordinal(low, medium, high)**

Logistic Regression

- Formula of logistic function

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

- Best Fit Equation in Linear Regression

$$y = \beta_0 + \beta_1 x$$

- Let's say instead of y we are taking probabilities (P). But there is an issue here, the value of (P) will exceed 1 or go below 0 and we know that range of Probability is (0-1). To overcome this issue we take ***"odds"*** of P:

$$P = \beta_0 + \beta_1 x$$

$$\frac{P}{1-P} = \beta_0 + \beta_1 x$$

- Odds are nothing but the ratio of the probability of success and probability of failure.

Logistic Regression

- We know that odds can always be positive which means the range will always be (0,+∞).
- The problem here is that the range is restricted and we don't want a restricted range because if we do so then our correlation will decrease.
- To control this we take the ***log of odds*** which has a range from (-∞,+∞).

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x$$

Logistic Regression

*Take **exponent** on both sides*

$$\exp[\log(\frac{p}{1-p})] = \exp(\beta_0 + \beta_1 x)$$

$$e^{\ln[\frac{p}{1-p}]} = e^{(\beta_0 + \beta_1 x)}$$

$$\frac{p}{1-p} = e^{\left(\beta_0 + \beta_1 x\right)}$$

$$p = e^{\left(\beta_0 + \beta_1 x\right)} - pe^{\left(\beta_0 + \beta_1 x\right)}$$

$$p[1 + e^{\left(\beta_0 + \beta_1 x\right)}] = e^{\left(\beta_0 + \beta_1 x\right)}$$

$$p = \frac{e^{\left(\beta_0 + \beta_1 x\right)}}{1 + e^{\left(\beta_0 + \beta_1 x\right)}}$$

Now dividing by $e^{\left(\beta_0 + \beta_1 x\right)}$, *we will get*

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$ *This is our sigmoid function.*

Decision Tree

- The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses *Entropy* and *Information Gain* to construct a decision tree.
- **Entropy-** A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.
  - To build a decision tree, we need to calculate two types of entropy using frequency tables as follows:
  - Entropy using the frequency table of one attribute:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

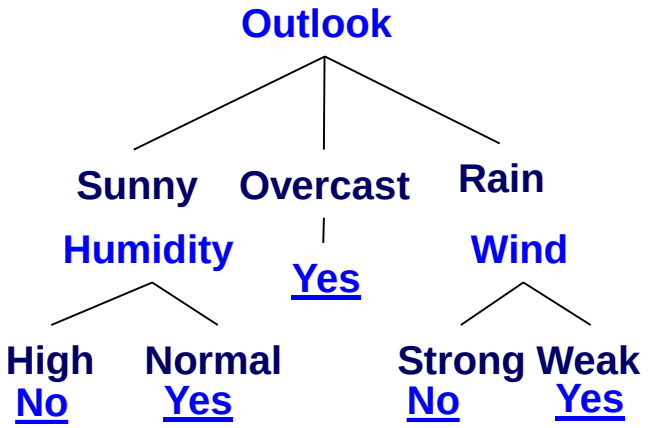
Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

Decision Tree

- Can represent any Boolean Function
- Can be viewed as a way to compactly represent a lot of data.
- Natural representation
- The evaluation of the Decision Tree Classifier is easy

- Clearly, given data, there are many ways to represent it as a decision tree.
- Learning a good representation from data is the challenge.



Will I play tennis today?

- **Features**
  - Outlook: {Sun, Overcast, Rain}
  - Temperature: {Hot, Mild, Cool}
  - Humidity: {High, Normal, Low}
  - Wind: {Strong, Weak}

- **Labels**
  - Binary classification task: Y = {+, -}

Decision Tree

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Outlook: S(unny), O(vercast), R(ainy)

Temperature: H(ot), M(edium), C(ool)

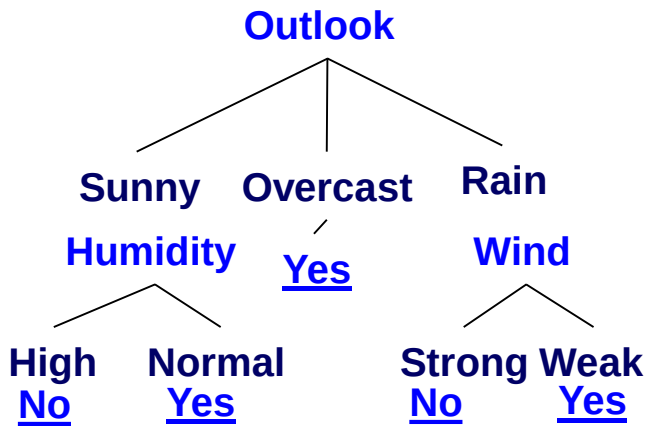
Humidity: H(igh), N(ormal), L(ow)

Wind: S(trong), W(eak)

Decision Tree

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

- Data is processed in Batch (i.e. all the data available)
- Recursively build a decision tree top down.



Basic Decision Tree Algorithm

- Let S be the set of Examples
  - Label is the target attribute (the prediction)
  - Attributes is the set of measured attributes
- ID3(*S*, Attributes, Label)

If all examples are labeled the same return a single node tree with Label

Otherwise Begin

A = attribute in Attributes that ***best*** classifies S (Create a Root node for tree)

for each possible value v of A

Add a new tree branch corresponding to A=v

Let *Sv* be the subset of examples in S with A=v

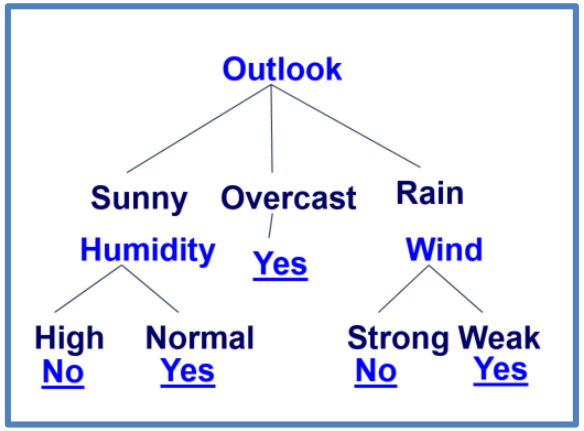
if *Sv* is empty: add leaf node with the common value of Label in S

Else: below this branch add the subtree

ID3(*Sv*, Attributes - {a}, Label)

End

Return Root



Picking the Root Attribute

- The goal is to have the resulting decision tree as small as possible (Occam's Razor)
  - The main decision in the algorithm is the selection of the next attribute to condition on.
- We want attributes that split the examples to sets that are relatively pure in one label; this way we are closer to a leaf node.
  - The most popular heuristics is based on information gain, originated with the ID3 system of Quinlan.

Entropy

- Entropy (impurity, disorder) of a set of examples, S, relative to a binary classification is:

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

- p_+ is the proportion of positive examples in S and
- p_- is the proportion of negatives examples in S
  - If all the examples belong to the same category: Entropy = 0
  - If all the examples are equally mixed (0.5, 0.5): Entropy = 1
  - Entropy = Level of uncertainty.
- In general, when p_i is the fraction of examples labeled i:

$$Entropy(S[p_1, p_2, \ldots, p_k]) = -\sum_1^k p_i \log(p_i)$$

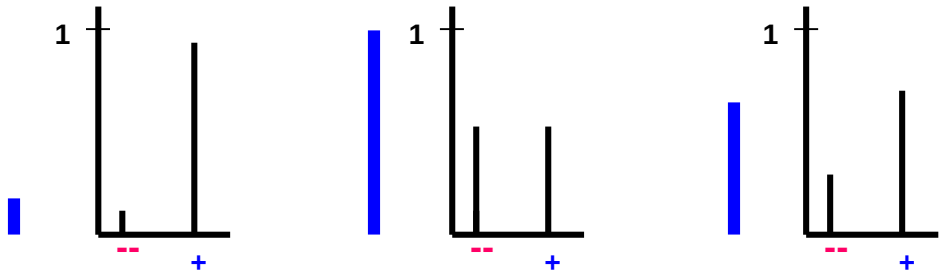
- Entropy can be viewed as the number of bits required, on average, to encode the class of labels. If the probability for + is 0.5, a single bit is required for each example; if it is 0.8 – can use less then 1 bit.

Entropy

- Entropy (impurity, disorder) of a set of examples, S, relative to a binary classification is:

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

- p_+ is the proportion of positive examples in S and
- p_- is the proportion of negatives examples in S
  - If all the examples belong to the same category: Entropy = 0
  - If all the examples are equally mixed (0.5, 0.5): Entropy = 1
  - Entropy = Level of uncertainty.

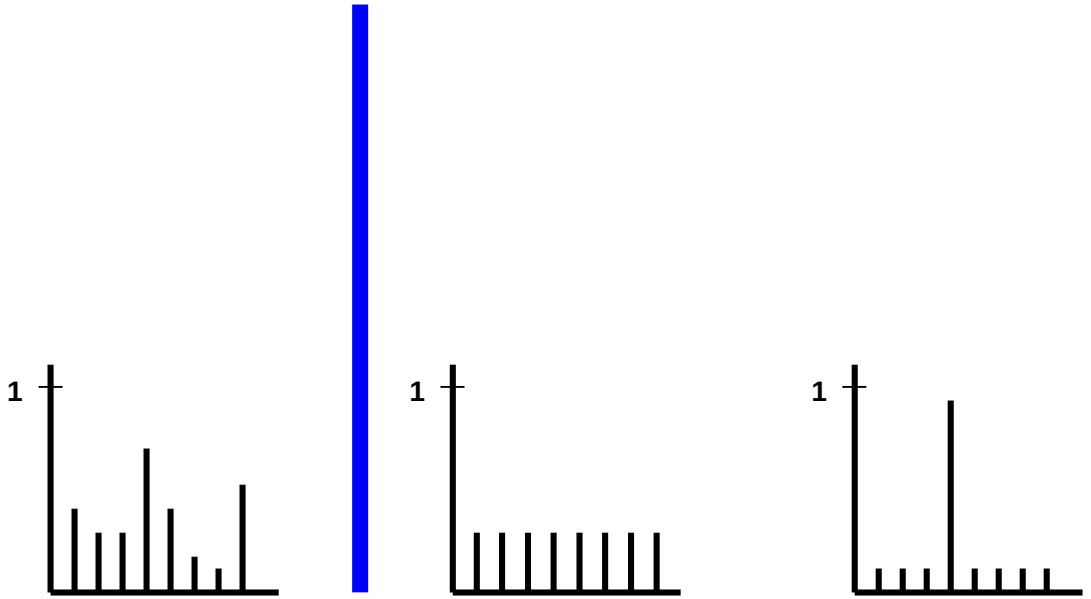


Entropy

(Convince yourself that the max value would be $\log(k)$)

(Also note that the base of the log only introduce a constant factor; therefore, we'll think about base 2)

$$Entropy(S[p_1, p_2, \ldots, p_k]) = -\sum_1^k p_i \log(p_i)$$



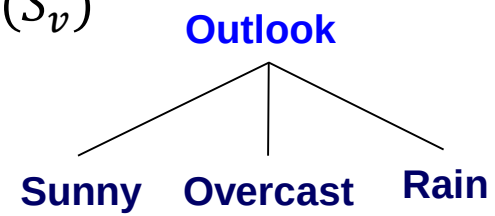
Information Gain

High Entropy – High level of Uncertainty

Low Entropy – No Uncertainty.

- The information gain of an attribute a is the expected reduction in entropy caused by partitioning on this attribute

$$Gain(S, a) = Entropy(S) - \sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$$



- Where:
  - S_v is the subset of S for which attribute a has value v, and
  - the entropy of partitioning the data is calculated by **weighing the entropy of each partition** by its size relative to the original set

- Partitions of low entropy (imbalanced splits) lead to high gain
- Go back to check which of the A, B splits is better

Will I play tennis today?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Outlook: S(unny), O(vercast), R(ainy)

Temperature: H(ot), M(edium), C(ool)

Humidity: H(igh), N(ormal), L(ow)

Wind: S(trong), W(eak)

Will I play tennis today?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

calculate current entropy

- $p_+ = \frac{9}{14} \quad p_- = \frac{5}{14}$
- $Entropy(Play) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$

$$= -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14}$$

$$\approx 0.94$$

Information Gain: Outlook

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

$$Gain(S,a) = Entropy(S) - \sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Outlook = sunny:

$p_+ = 2/5 \quad p_- = 3/5$ **Entropy(O = S) = 0.971**

Outlook = overcast:

$p_+ = 4/4 \quad p_- = 0$ **Entropy(O = O) = 0**

Outlook = rainy:

$p_+ = 3/5 \quad p_- = 2/5$ **Entropy(O = R) = 0.971**

Expected entropy

= $\sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$

= (5/14)×0.971 + (4/14)×0 + (5/14)×0.971 = **0.694**

Information gain = 0.940 – 0.694 **= 0.246**

Information Gain: Humidity

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

$$Gain(S,a) = Entropy(S) - \sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Humidity = high:

$p_+ = 3/7 \quad p_- = 4/7$ **Entropy(H = H) = 0.985**

Humidity = Normal:

$p_+ = 6/7 \quad p_- = 1/7$ **Entropy(H = N) = 0.592**

Expected entropy

= $\sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$

= (7/14)×0.985 + (7/14)×0.592 = **0.7785**

Information gain = 0.940 – 0.7785 **= 0.1515**

Which feature to split on?

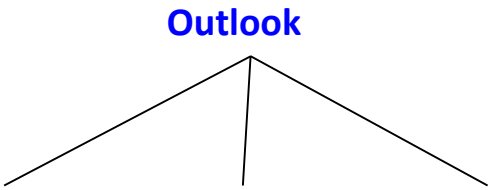
	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Information gain:

Outlook: 0.246
Humidity: 0.151
Wind: 0.048
Temperature: 0.029

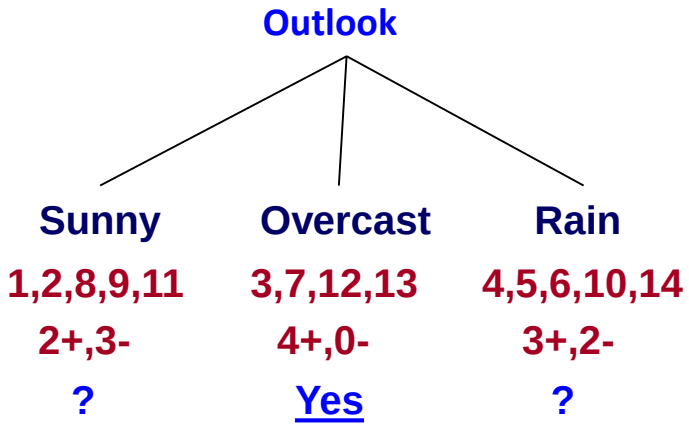
→ Split on Outlook

An Illustrative Example (III)



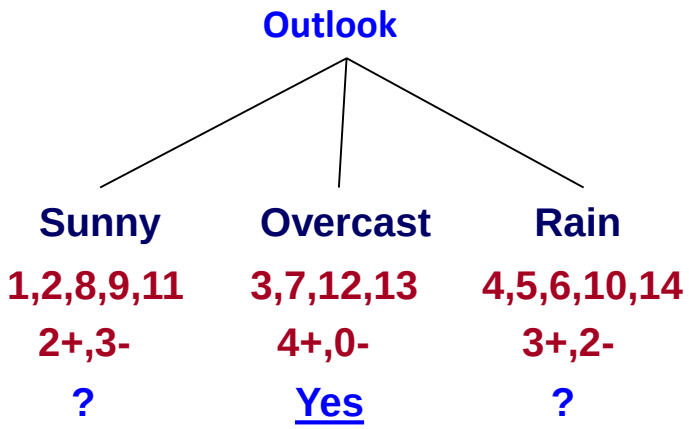
Gain(S,Humidity)=0.151
Gain(S,Wind) = 0.048
Gain(S,Temperature) = 0.029
Gain(S,Outlook) = 0.246

An Illustrative Example (III)



	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

An Illustrative Example (III)

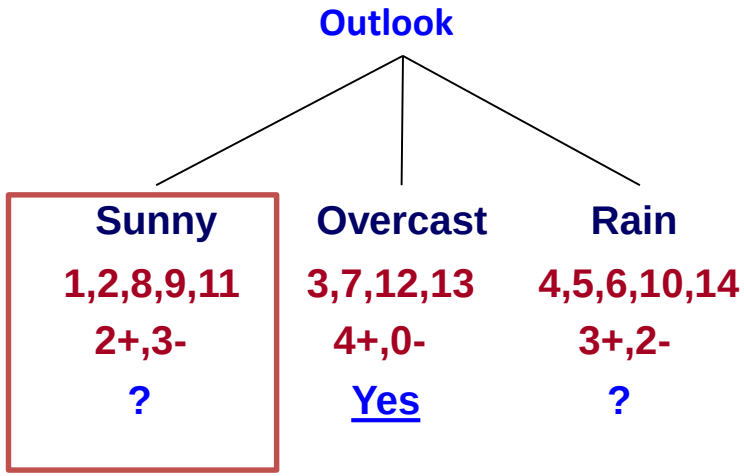


Continue until:

- Every attribute is included in path, or,
- All examples in the leaf have same label

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

An Illustrative Example (IV)



$Gain(S_{sunny}, Humidity) = .97-(3/5)\ 0-(2/5)\ 0 = .97$

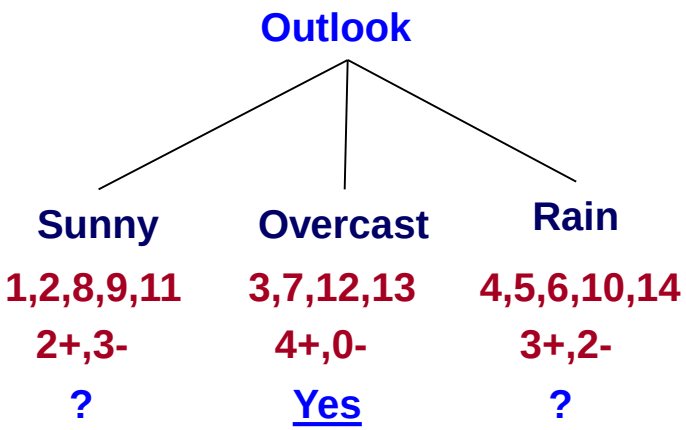
$Gain(S_{sunny}, Temp) = .97-\ 0-(2/5)\ 1 = .57$

$Gain(S_{sunny}, Wind) = .97-(2/5)\ 1 - (3/5)\ .92= .02$

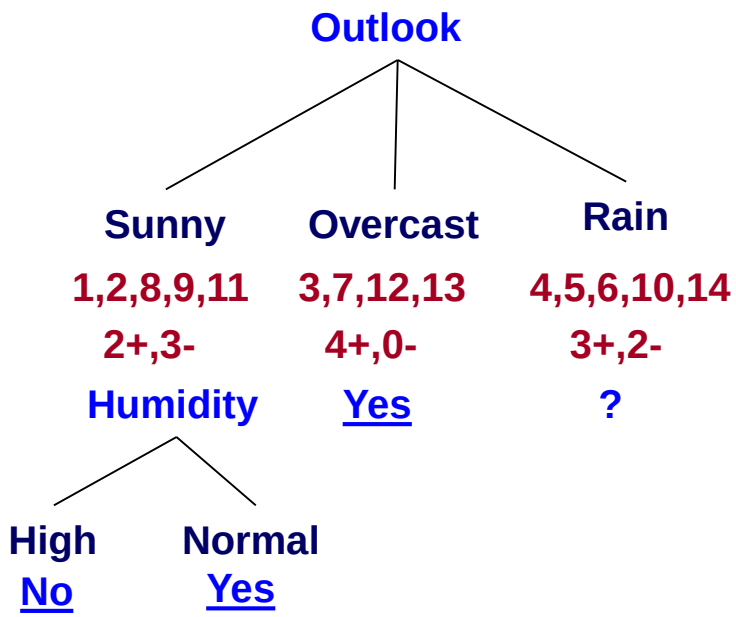
	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Split on Humidity

An Illustrative Example (V)



An Illustrative Example (V)



induceDecisionTree(S)

- 1. Does S uniquely define a class?

  if all s ∈ S have the same label y: **return** S;

- 2. Find the feature with the most information gain:

  $i = \text{argmax}_i \, \mathit{Gain}(S, X_i)$

- 3. Add children to S:

  for *k* **in** Values(X_i):

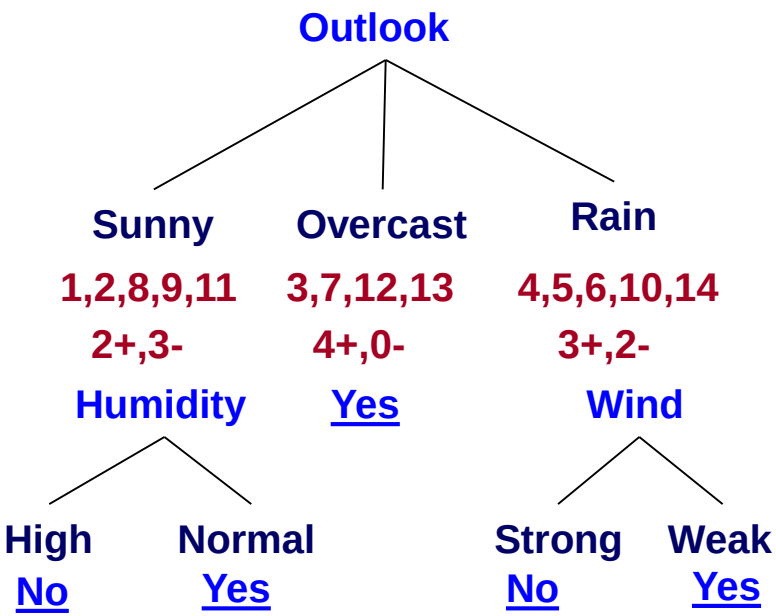
  $S_k = \{s \in S \mid x_i = k\}$

  addChild(S, S_k)

  induceDecisionTree(S_k)

  return S;

An Illustrative Example (VI)



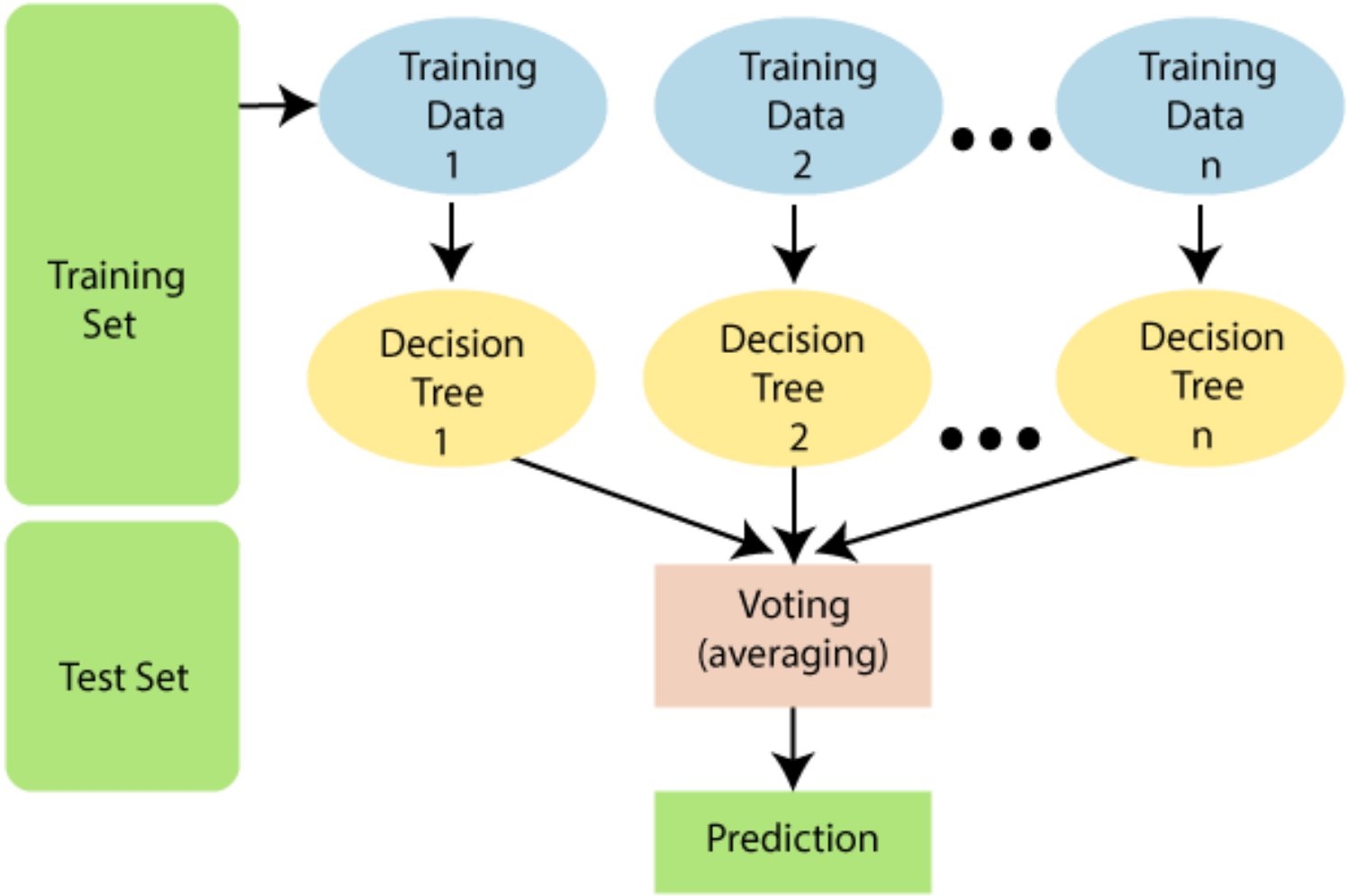
Random Forest

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.
- It can be used for both Classification and Regression problems in ML.
- It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

Random Forest

- ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."***
- Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
- **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

Random Forest



Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

1. There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
2. The predictions from each tree must have very low correlations.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

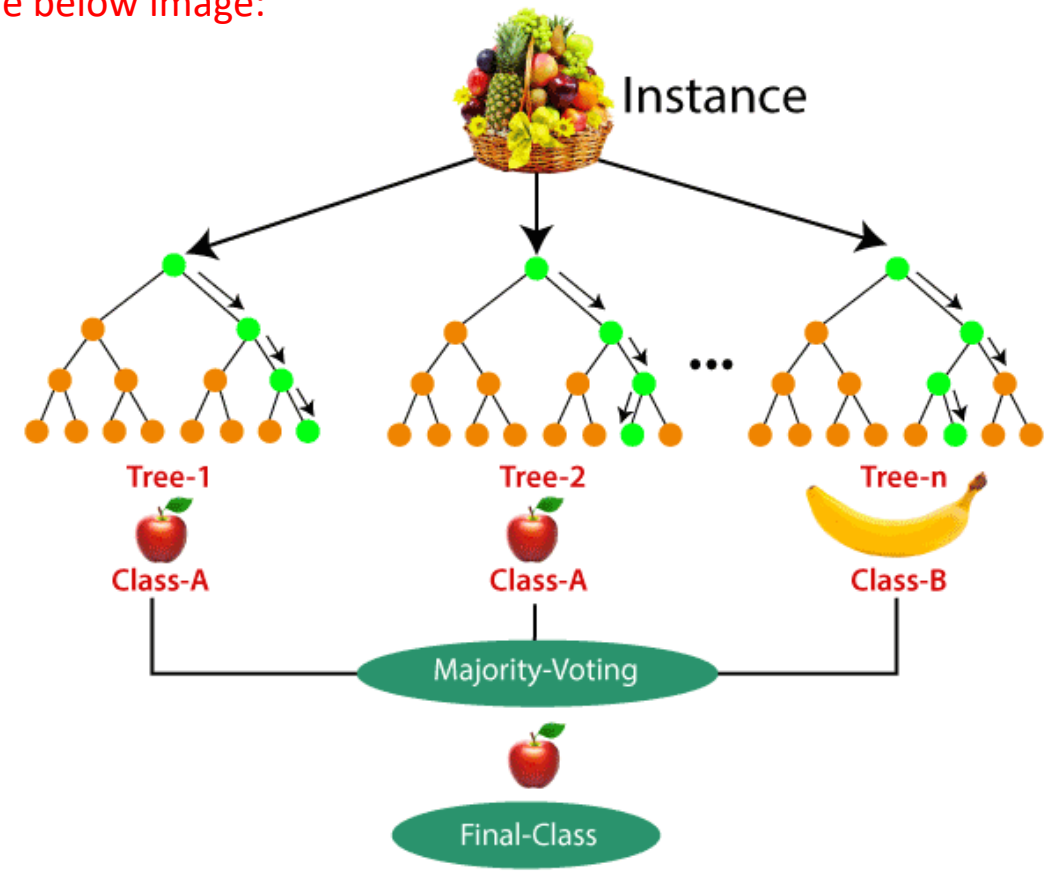
Random Forest algorithm

Working process can be explained in the below steps and diagram:

- **Step-1:** Select random K data points from the training set.
- **Step-2:** Build the decision trees associated with the selected data points (Subsets).
- **Step-3:** Choose the number N for decision trees that you want to build.
- **Step-4:** Repeat Step 1 & 2.
- **Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Random Forest algorithm

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



Advantages and Disadvantages of Random Forest

Advantages

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

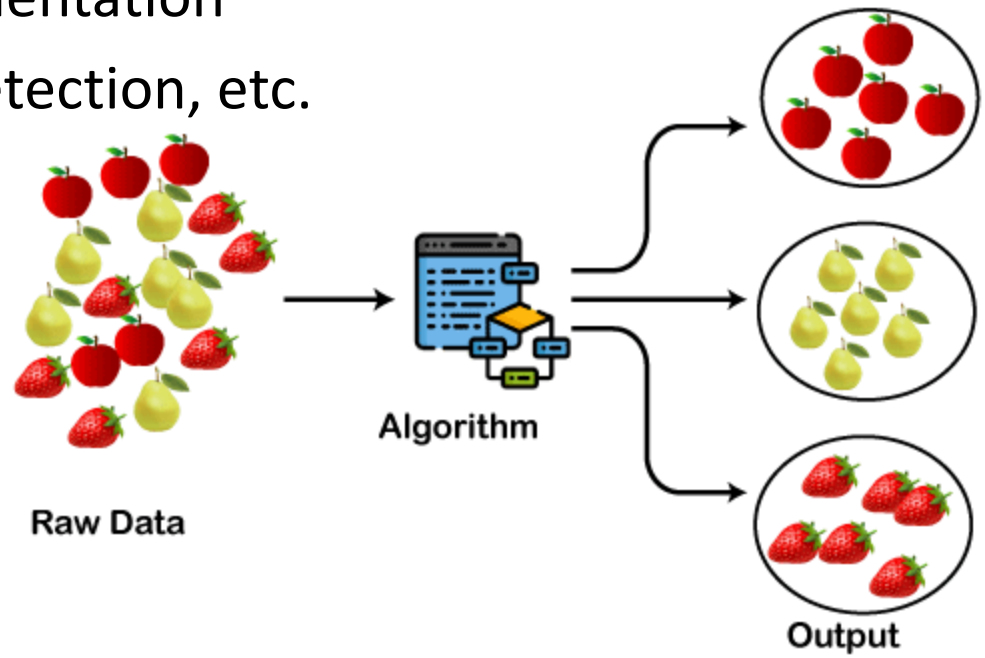
Clustering

- Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset.
- It can be defined as ***"A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."***
- It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.
- **Example**: Let's understand the clustering technique with the real-world example of Mall: When we visit any shopping mall, we can observe that the things with similar usage are grouped together. Such as the t-shirts are grouped in one section, and trousers are at other sections, similarly, at vegetable sections, apples, bananas, Mangoes, etc., are grouped in separate sections, so that we can easily find out the things.

Clustering

The clustering technique can be widely used in various tasks.

- Market Segmentation
- Statistical data analysis
- Social network analysis
- Image segmentation
- Anomaly detection, etc.



Types of Clustering Methods

1. **Partitioning Clustering**
2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

Clustering Algorithms

1. **K-Means algorithm:** The k-means algorithm is one of the most popular clustering algorithms. It classifies the dataset by dividing the samples into different clusters of equal variances. The number of clusters must be specified in this algorithm. It is fast with fewer computations required, with the linear complexity of **O(n).**
2. **Mean-shift algorithm:** Mean-shift algorithm tries to find the dense areas in the smooth density of data points. It is an example of a centroid-based model, that works on updating the candidates for centroid to be the center of the points within a given region.
3. **DBSCAN Algorithm:** It stands **for Density-Based Spatial Clustering of Applications with Noise**. It is an example of a density-based model similar to the mean-shift, but with some remarkable advantages. In this algorithm, the areas of high density are separated by the areas of low density. Because of this, the clusters can be found in any arbitrary shape.

Clustering Algorithms

4. **Expectation-Maximization Clustering using GMM:** This algorithm can be used as an alternative for the k-means algorithm or for those cases where K-means can be failed. In GMM, it is assumed that the data points are Gaussian distributed.
5. **Agglomerative Hierarchical algorithm:** The Agglomerative hierarchical algorithm performs the bottom-up hierarchical clustering. In this, each data point is treated as a single cluster at the outset and then successively merged. The cluster hierarchy can be represented as a tree-structure.
6. **Affinity Propagation:** It is different from other clustering algorithms as it does not require to specify the number of clusters. In this, each data point sends a message between the pair of data points until convergence. It has $O(N^2T)$ time complexity, which is the main drawback of this algorithm.

Exercise

For the given data, compute two clusters using K-means algorithm for clustering where initial cluster centers are (1.00, 1.00) and (5.00, 7.00). Execute for two iterations.

Record Number	A	B
R1	1.00	1.00
R2	1.50	2.00
R3	3.00	4.00
R4	5.00	7.00
R5	3.50	5.00
R6	4.50	5.50
R7	3.50	4.50

Solution

Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

- Initialization: Number of clusters (K) = 2, centroid for cluster1 (C1)= (1.0, 1.0) and centroid for cluster2 (C2) = (5.0, 7.0).
- We use Euclidean distance to find closest point to centroids.

- Iteration1:

Record Number	Close to C1(1.0, 1.0)	Close to C2(5.0, 7.0)	Assign to cluster
R1(1.0,1.0)	dist(R1, C1)=0.0	dist(R1, C2)=7.21	Cluster1
R2(1.5,2.0)	dist(R2, C1)=1.12	dist(R2, C2)=6.12	Cluster1
R3(3.0,4.0)	dist(R3, C1)=3.61	dist(R3, C2)=3.61	Cluster1
R4(5.0,7.0)	dist(R4, C1)=7.21	dist(R4, C2)=0.0	Cluster2
R5(3.5,5.0)	dist(R5, C1)=4.12	dist(R5, C2)=2.5	Cluster2
R6(4.5,5.0)	dist(R6, C1)= 5.31	dist(R6, C2)=2.06	Cluster2
R7(3.5,4.5)	dist(R7,C1)=4.30	dist(R7, C2)=2.92	Cluster2

- Thus, we obtain two clusters containing: Cluster1 {R1, R2, R3} and Cluster2 {R4, R5, R6, R7}.
- Their new centroids are:
- C1 = (1.0+1.5+3.0)/3, (1.0+2.0+4.0)/3
- = 5.5/3, 7.0/3 = 1.83, 2.33
- C2 = (5.0+3.5+4.5+3.5)/4, (7+5+5+4.5)/4
- = 16.5/4, 21.5/4 = 4.12, 5.37

- Iteration2:

Record Number	**Close to C1**(1.83, 2.33)	**Close to C2**(4.12, 5.37)	**Assign to cluster**
R1(1.0,1.0)	dist(R1, C1)=1.57	dist(R1, C2)=5.37	Cluster1
R2(1.5,2.0)	dist(R2, C1)=0.47	dist(R2, C2)=4.27	Cluster1
R3(3.0,4.0)	dist(R3, C1)=2.04	dist(R3, C2)=1.77	Cluster2
R4(5.0,7.0)	dist(R4, C1)=5.64	dist(R4, C2)=1.85	Cluster2
R5(3.5,5.0)	dist(R5, C1)=3.15	dist(R5, C2)=0.72	Cluster2
R6(4.5,5.0)	dist(R6, C1)=3.78	dist(R6, C2)=0.53	Cluster2
R7(3.5,4.5)	dist(R7,C1)=2.74	dist(R7, C2)=1.07	Cluster2

- Therefore, new clusters are:
- Cluster1 {R1, R2} and Cluster2 {R3, R4, R5, R6, R7}.
  - Their new centroids are:
  - C1 = (1.0+1.5)/2, (1.0+2.0)/2
  - = 2.50/2, 3.0/2 = 1.25, 1.5
  - C2 = (3.0+5.0+3.5+4.5+3.5)/5, (4+7+5+5+4.5)/5
  - = 19.5/5, 25.5/5 = 3.9, 5.1

Ensemble Learning

- When you want to purchase a new car, will you walk up to the first car shop and purchase one based on the advice of the dealer? It's highly unlikely.
- You would likely browser a few web portals where people have posted their reviews and compare different car models, checking for their features and prices. You will also probably ask your friends and colleagues for their opinion. In short, you wouldn't directly reach a conclusion, but will instead make a decision considering the opinions of other people as well.

Ensemble Learning

- Ensemble learning is a machine learning technique that enhances accuracy and resilience in forecasting by merging predictions from multiple models.
- It aims to mitigate errors or biases that may exist in individual models by leveraging the collective intelligence of the ensemble.

Simple Ensemble Techniques

1. Max Voting
2. Averaging
3. Weighted Averaging

Ensemble Learning

- Max Voting : The max voting method is generally used for classification problems. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a 'vote'. The predictions which we get from the majority of the models are used as the final prediction
- Averaging Similar to the max voting technique, multiple predictions are made for each data point in averaging. In this method, we take an average of predictions from all the models and use it to make the final prediction. Averaging can be used for making predictions in regression problems or while calculating probabilities for classification problems.
- Weighted Average: This is an extension of the averaging method. All models are assigned different weights defining the importance of each model for prediction. For instance, if two of your colleagues are critics, while others have no prior experience in this field, then the answers by these two friends are given more importance as compared to the other people.

Thank You!