

CST401 ARTIFICIAL INTELLIGENCE

MODULE 5



















Course Outcome

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Explain the fundamental concepts of intelligent systems and their architecture. (Cognitive Knowledge Level: Understanding)
CO2	Illustrate uninformed and informed search techniques for problem solving in intelligent systems. (Cognitive Knowledge Level: Understanding)
CO3	Solve Constraint Satisfaction Problems using search techniques. (Cognitive Knowledge Level: Apply)
CO4	Represent AI domain knowledge using logic systems and use inference techniques for reasoning in intelligent systems. (Cognitive Knowledge Level: Apply)
CO5	Illustrate different types of learning techniques used in intelligent systems (Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												

Abstract POs defined by National Board of Accreditation

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	60	30	40
Apply	20	40	30
Analyze			
Evaluate			
Create			

Mark Distribution

Mark Distribution

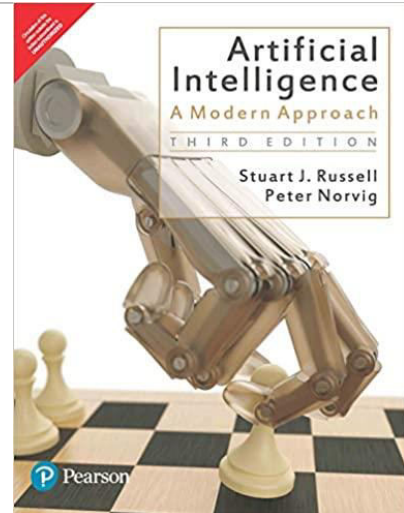
Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Continuous Assessment Tests(Average of Series Tests 1 & 2)	25 marks
Continuous Assessment Assignment	15 marks

Textbook

Stuart Russell and Peter Norvig. **Artificial Intelligence: A Modern Approach**, 3rd Edition. Prentice Hall.



SYLLABUS- Machine Learning

Learning from Examples –

- Forms of Learning,
- Supervised Learning,
- Learning Decision Trees,
- Evaluating and choosing the best hypothesis,
- Regression and classification with Linear models.

CO5: Illustrate different types of learning techniques used in intelligent systems
(Cognitive Knowledge Level: Understand)

PO1-Engineering Knowledge
PO2 Problem Analysis
PO5 Modern tool usage
PO12 Life long learning

Learning From Example

Agents that can improve their behavior through diligent study of their own experiences

Learning

An agent is learning if it improves its performance on future tasks after making observations about the world

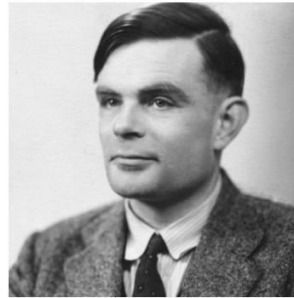
MIND
A QUARTERLY REVIEW
OF
PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND
INTELLIGENCE

By A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the

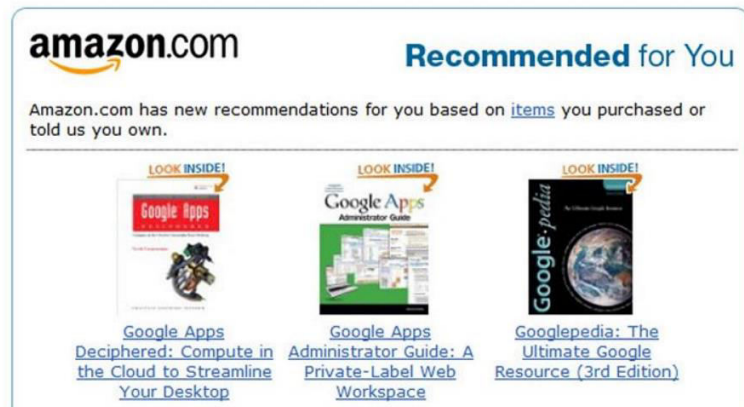


A. M. Turing, "Computing Machinery and Intelligence,"
Mind 59, no. 236 (1950): 433-460.

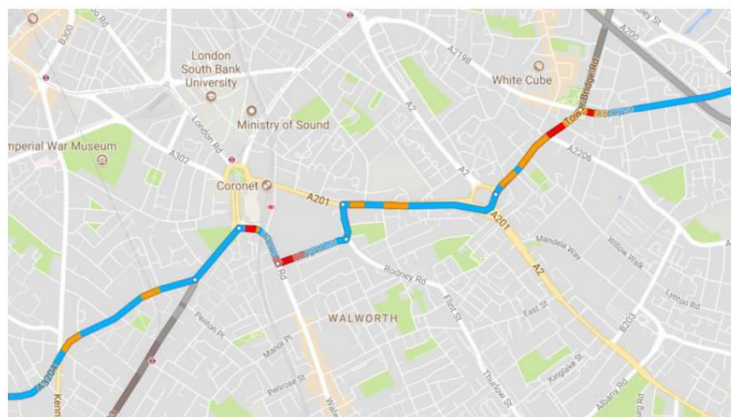
Intelligent personal assistant



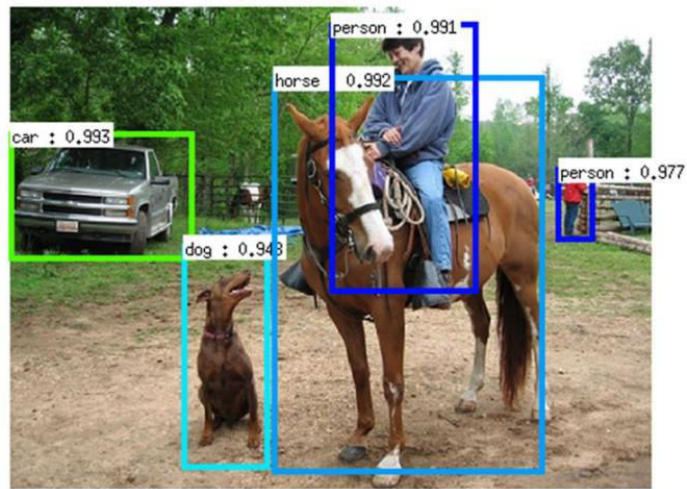
Recommendations



Predictions



Object Detection



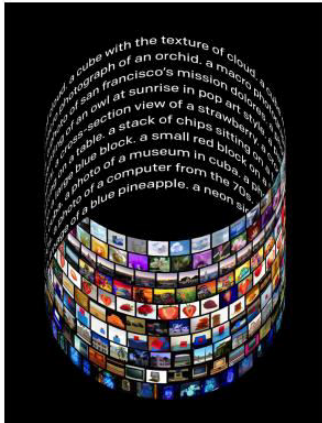
Colourization



OpenAI's DALL-E

DALL-E

- AI program creating images from text descriptions.



TEXT PROMPT

an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES



Edit prompt or view more images +

TEXT PROMPT

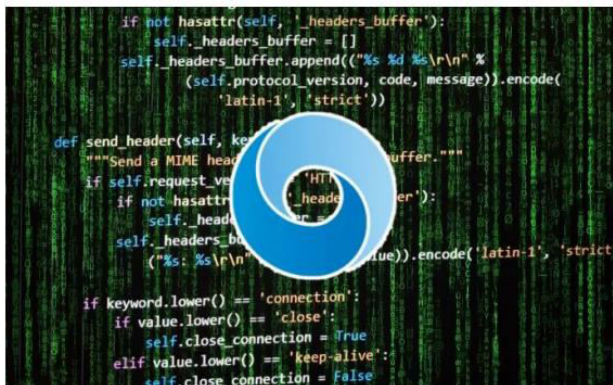
an armchair in the shape of an avocado [...]

AI-GENERATED IMAGES



Edit prompt or view more images +

DeepMind's AlphaCode (Feb 2022)



Takes english descriptions of competitive programming problems and generates solutions.

When participated in Codeforce's platform, it was ranked top 54% among 5000+ competitors.

<https://deepmind.com/blog/article/Competitive-programming-with-AlphaCode>

Google's LaMDA (January 2022)

LaMDA → Language Models for Dialog Applications (137 Billion)

Hi! Hello, I am a friendly dialog model. What do you want to talk about?"



What's a good topic for a new blog?

Well there are so many! How about something about a new food item that you just tried.



There are lots of food blogs. How can I make it unique?

Hmmm, you can write about a food that you tried somewhere that is exclusive to that location or restaurant.

A GOOGLE engineer has reportedly been suspended by the company after claiming that an artificial intelligence he helped to develop had become sentient. "If I didn't know exactly what it was, which is this computer program we built recently, I'd think it was a seven-year-old, eight-year-old kid," Blake Lemoine told *The Washington Post*.

Lemoine released transcripts of conversations with the AI, called LaMDA (Language Model for Dialogue Applications), in which it appears to express fears of being switched off, talks about how it feels happy and sad and attempts to form bonds with humans by mentioning situations that it could never have actually experienced. Here is everything you need to know.

Google's Parti (June 2022)



A close-up high-contrast photo of Sydney Opera House sitting next to Eiffel tower, under a blue night sky of roiling energy, exploding yellow stars, and radiating swirls of blue.

Machine Learning

An algorithmic way of making sense (learning) from data.

Forms of Learning

Any component of an agent can be improved by learning from data.

The improvements, and the techniques used to make them, depend on four major factors

- Which component is to be improved.
- What prior knowledge the agent already has.
- What representation is used for the data and the component.
- What feedback is available to learn from.

Components to be learned

The components of these agents include:

1. A direct mapping from conditions on the current state to actions.
2. A means to infer relevant properties of the world from the percept sequence.
3. Information about the way the world evolves and about the results of possible actions the agent can take.
4. Utility information indicating the desirability of world states.
5. Action-value information indicating the desirability of actions.
6. Goals that describe classes of states whose achievement maximizes the agent's utility

Each of these components can be learned.

example

an agent training to become a taxi driver.

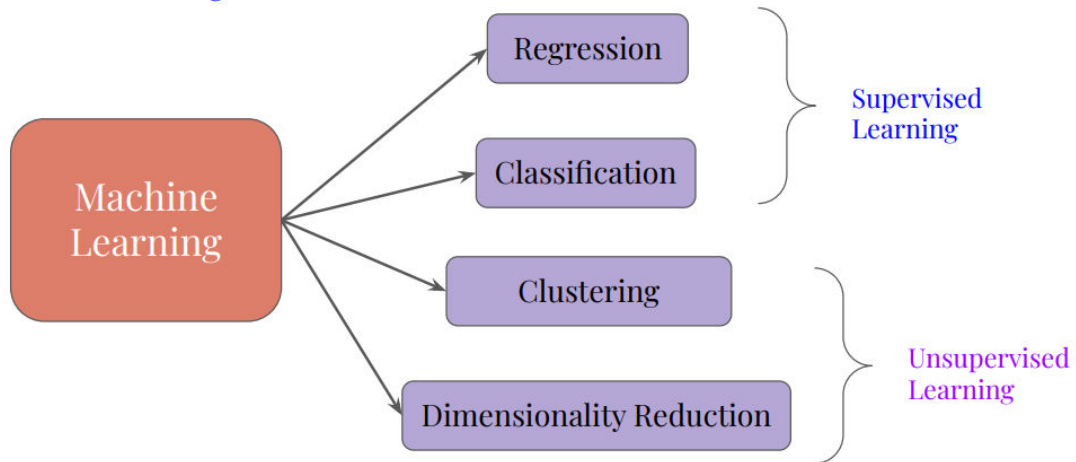
Every time the instructor shouts “Brake!” the agent might learn a condition– action rule for when to brake (component 1); the agent also learns every time the instructor does not shout.

By seeing many camera images that it is told contain buses, it can learn to recognize them (2).

By trying actions and observing the results—for example, braking hard on a wet road—it can learn the effects of its actions (3).

Then, when it receives no tip from passengers who have been thoroughly shaken up during the trip, it can learn a useful component of its overall utility function (4).

Forms of Learning



supervised learning

In supervised learning the machine experiences the examples along with the labels or targets for each example.

The labels in the data help the algorithm to correlate the features.

Two of the most common supervised machine learning tasks are

- **classification** and
- **regression**.

Classification

uses an algorithm to accurately assign test data into specific categories.

It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined.

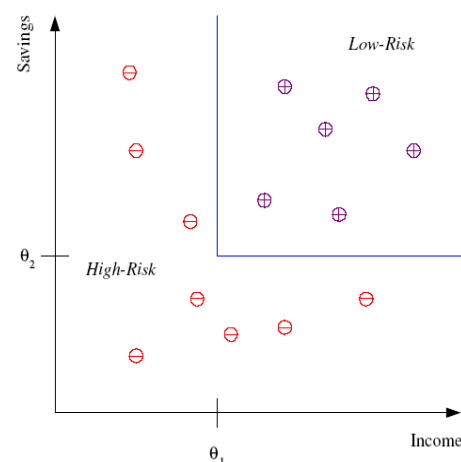
Common classification algorithms are

- linear classifiers,
- support vector machines (SVM),
- decision trees,
- k-nearest neighbor, and
- random forest

Classification example

Example: Credit scoring

Differentiating between **low-risk** and **high-risk** customers from their *income* and *savings*



Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN **low-risk** ELSE **high-risk**

Regression

It is used to understand the relationship between dependent and independent variables.

It is commonly used to make projections, such as for sales revenue for a given business.

Types are

- Linear regression,
- logistical regression, and
- polynomial regression

Regression example

Example: Price of a used car

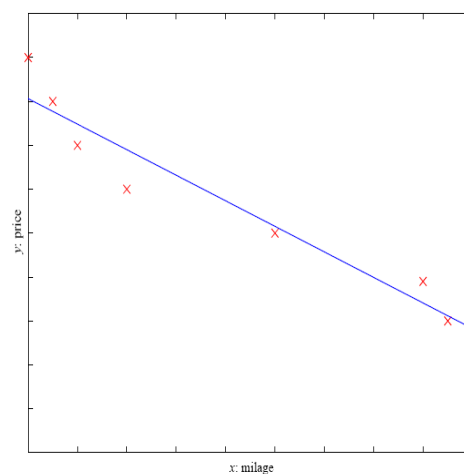
x : car attributes

y : price

$$y = g(x | \vartheta)$$

$g(\cdot)$ model,

ϑ parameters



Regression Applications

Navigating a car: Angle of the steering wheel (CMU NavLab)

Kinematics of a robot arm

Response surface design

Unsupervised Learning

When we have unclassified and unlabeled data, the system attempts to uncover patterns from the data .

There is no label or target given for the examples.

One common task is to group similar examples together called clustering.

Example applications

- Customer segmentation in CRM
- Image compression: Color quantization
- Bioinformatics: Learning motifs

Unsupervised learning models are used for three main tasks:

- clustering: group data according to "distance"
- association: find frequent co-occurrences
- link prediction: discover relationships in data
- data reduction: project features to fewer features

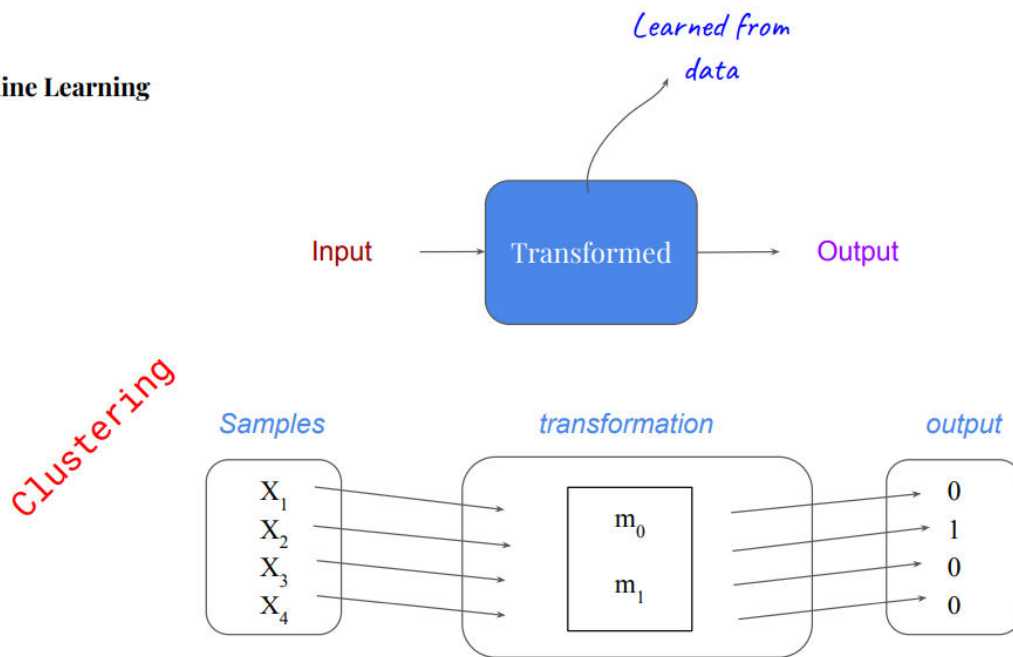
Clustering

It is a data mining technique for grouping unlabeled data based on their similarities or differences.

For example, K-means clustering algorithms assign similar data points into groups, where the K value represents the size of the grouping and granularity.

This technique is helpful for market segmentation, image compression, etc.

Machine Learning



DEPARTMENT OF CSE SNGCE

35

Dimensionality Reduction

It is a learning technique used when the number of features (or dimensions) in a given dataset is too high.

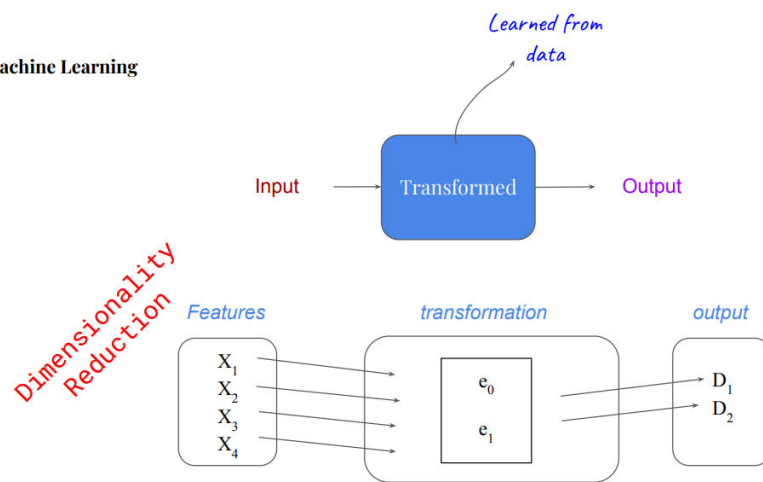
It reduces the number of data inputs to a manageable size while also preserving the data integrity.

Often, this technique is used in the preprocessing data stage, such as when autoencoders remove noise from visual data to improve picture quality.

DEPARTMENT OF CSE SNGCE

36

Machine Learning



Association

uses different rules to find relationships between variables in a given dataset.

These methods are frequently used for market basket analysis and recommendation engines, along the lines of “Customers Who Bought This Item Also Bought” recommendations.

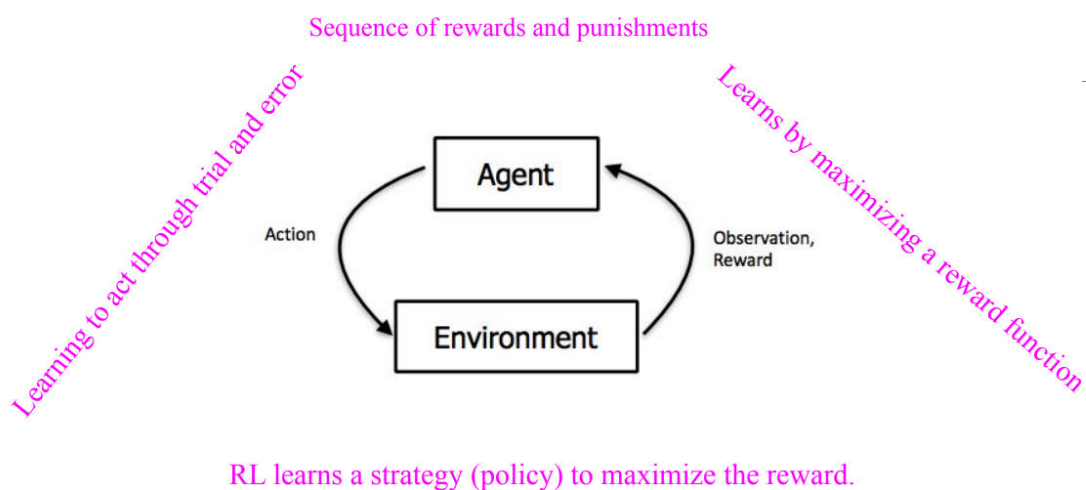
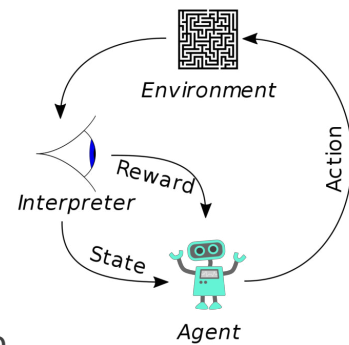
REINFORCEMENT LEARNING

Agent learns from a series of reinforcements rewards or punishments.

For example, the lack of a tip at the end of the journey gives the taxi agent an indication that it did something wrong.

The two points for a win at the end of a chess game tells the agent it did something right.

It is up to the agent to decide which of the actions prior to the reinforcement were most responsible for it.



semi-supervised learning

In semi-supervised learning we are given a few labeled examples and must make what we can of a large collection of unlabeled examples.

Even the labels themselves may not be the oracular truths that we hope for.

Imagine that you are trying to build a system to guess a person's age from a photo.

- You gather some labeled examples by snapping pictures of people and asking their age. That's supervised learning.
- But in reality some of the people lied about their age.
- It's not just that there is random noise in the data; rather the inaccuracies are systematic, and to uncover them is an unsupervised learning problem involving images, self-reported ages, and true (unknown) ages.
- Thus, both noise and lack of labels create a continuum between supervised and unsupervised learning

SUPERVISED LEARNING

The task of supervised learning is this

Given a training set of N example input–output pairs

$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$,

where each y_j was generated by an unknown function $y = f(x)$,

discover a function h that approximates the true function f .

Here x and y can be any value; they need not be numbers.

The function h is a hypothesis

Learning is a search through the space of possible hypotheses for one that will perform well, even on new examples beyond the training set.

To measure the accuracy of a hypothesis we give it a test set of examples that are distinct from the training set

a hypothesis generalizes well if it correctly predicts the value of y for novel examples.

Sometimes the function f is stochastic it is not strictly a function of x , and what we have to learn is a conditional probability distribution, $P(Y | x)$.

CLASSIFICATION AND REGRESSION

When the output y is one of a finite set of values (such as sunny, cloudy or rainy), the learning problem is called classification, and is called Boolean or binary classification if there are only two values.

When y is a number (such as tomorrow's temperature), the REGRESSION learning problem is called regression

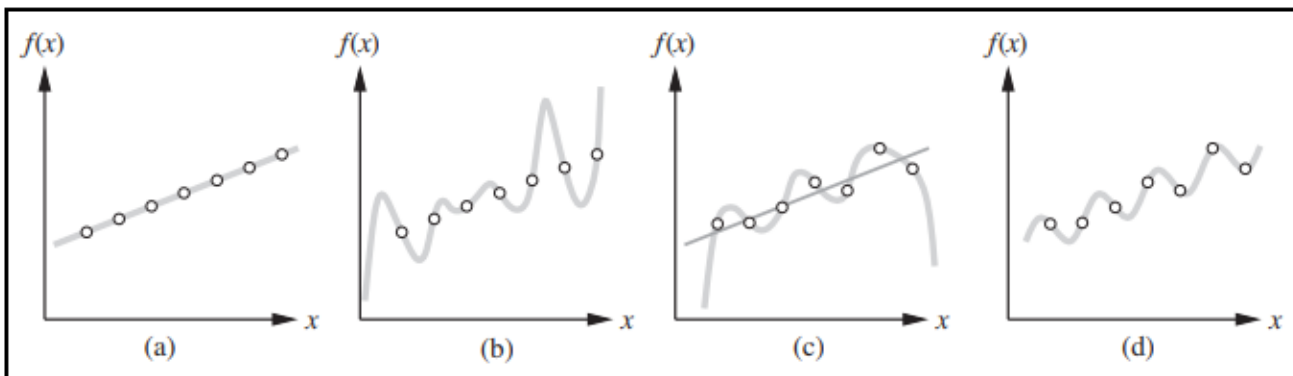


Figure 18.1 (a) Example $(x, f(x))$ pairs and a consistent, linear hypothesis. (b) A consistent, degree-7 polynomial hypothesis for the same data set. (c) A different data set, which admits an exact degree-6 polynomial fit or an approximate linear fit. (d) A simple, exact sinusoidal fit to the same data set.

fitting a function of a single variable to some data points.

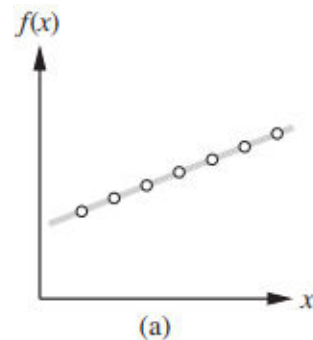
The examples are points in the (x, y) plane, where $y = f(x)$.

We don't know what f is, but we will approximate it with a function h selected from a hypothesis space, H ,

which for this example we will take to be the set of polynomials, such as $x^5 + 3x^2 + 2$

shows some data with an exact fit by a straight line

The line is called a consistent hypothesis because it agrees with all the data



DEPARTMENT OF CSE SNGCE

47

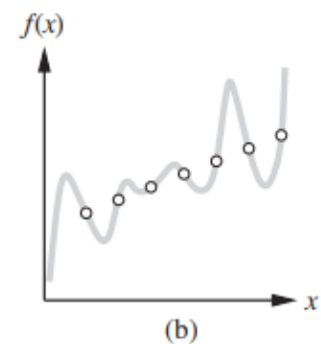
shows a **high degree polynomial that is also consistent with the same data.**

This illustrates a fundamental problem in inductive learning: **how do we choose from among multiple consistent hypotheses?**

One answer is to **prefer the simplest hypothesis consistent with the data.**

This principle is called **Ockham's razor**, after the 14th-century English philosopher William of Ockham, who used it to argue sharply against all sorts of complications.

Defining simplicity is not easy, but it seems clear that a degree-1 polynomial is simpler than a degree-7 polynomial, and thus (a) should be preferred to (b).



DEPARTMENT OF CSE SNGCE

48

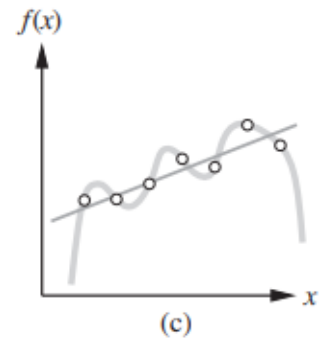
shows a second data set.

There is no consistent straight line for this data set; in fact, it requires a degree-6 polynomial for an exact fit.

There are just 7 data points, so a polynomial with 7 parameters does not seem to be finding any pattern in the data and we do not expect it to generalize well.

A straight line that is not consistent with any of the data points, but might generalize fairly well for unseen values of x , is also shown in (c).

In general, **there is a tradeoff between complex hypotheses that fit the training data well and simpler hypotheses that may generalize better**



DEPARTMENT OF CSE SNGCE

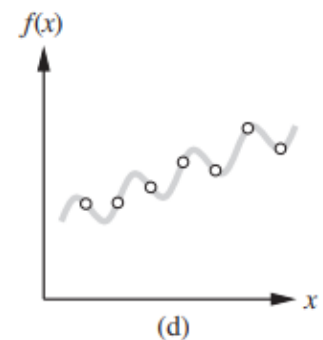
49

we expand the hypothesis space H to allow polynomials over both x and $\sin(x)$, and find that the data in (c) can be fitted exactly by a simple function of the form $ax + b + c \sin(x)$.

This shows the importance of the choice of hypothesis space.

We say that a learning problem is **realizable** if the hypothesis space contains the true function.

Unfortunately, **we cannot always tell whether a given learning problem is realizable, because the true function is not known**



DEPARTMENT OF CSE SNGCE

50

Supervised learning can be done by choosing the hypothesis h^* that is most probable given the data:

$$h^* = \operatorname{argmax}_{h \in \mathcal{H}} P(h|data) .$$

By Bayes' rule this is equivalent to

$$h^* = \operatorname{argmax}_{h \in \mathcal{H}} P(data|h) P(h) .$$

Then we can say that the prior probability $P(h)$ is high for a degree-1 or -2 polynomial, lower for a degree-7 polynomial, and especially low for degree-7 polynomials with large, sharp spikes as in Figure 18.1(b).

We allow unusual-looking functions when the data say we really need them, but we discourage them by giving them a low prior probability.

Decision Tree Classification Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.

It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

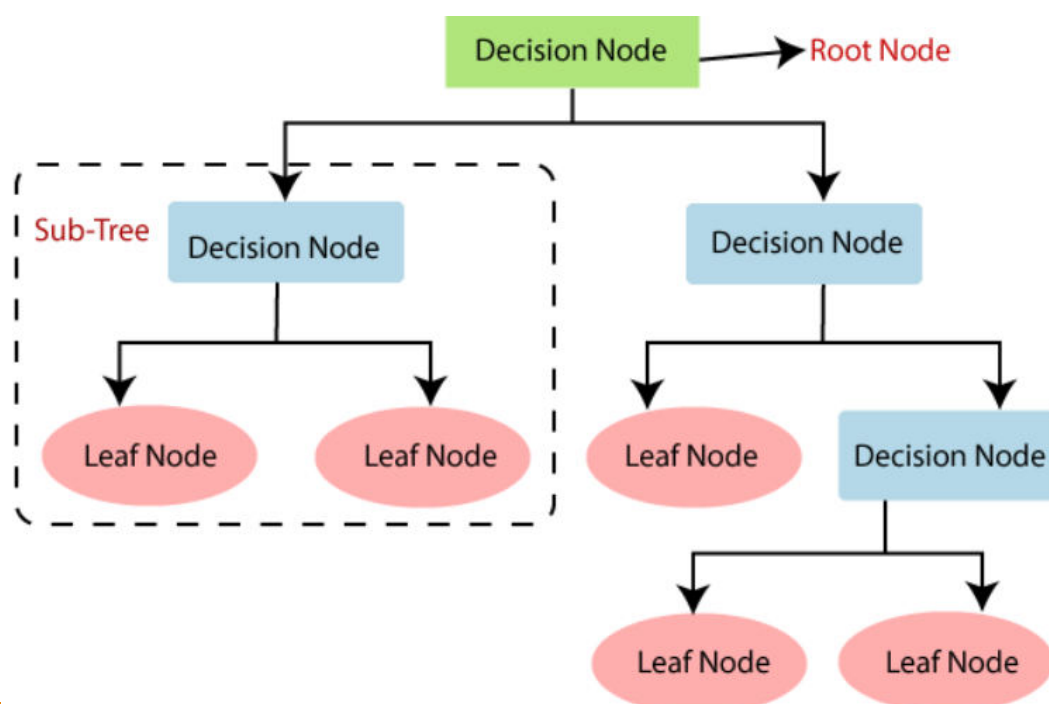
In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm

A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.



Why use Decision Trees?

Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.

The logic behind the decision tree can be easily understood because it shows a treelike structure.

Decision Tree Terminologies

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets

Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Sub Tree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches from the tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

How does the Decision Tree algorithm Work?

Step-1: Begin the tree with the root node, says S , which contains the complete dataset.

Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

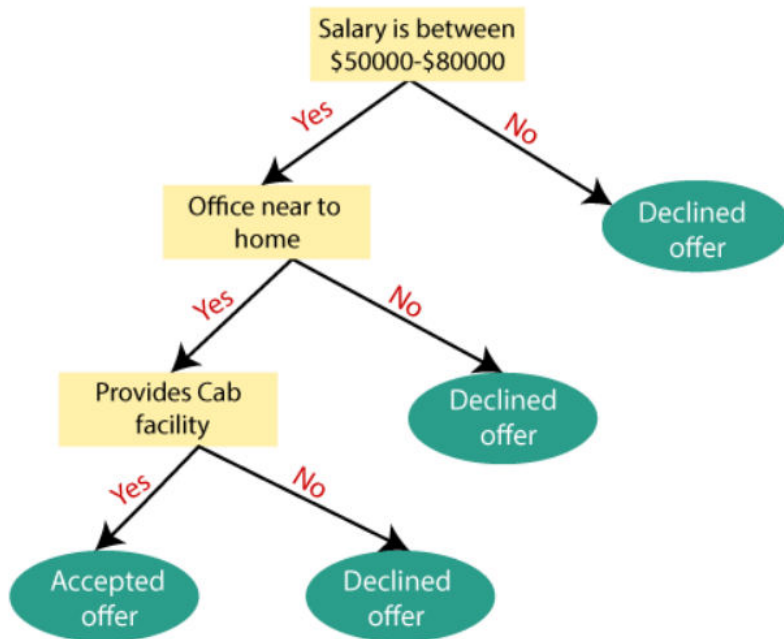
Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example

Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer).



Selection Measures: Attribute selection measure or ASM

1. Information Gain
2. Gini Index

Information Gain

Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.

It calculates how much information a feature provides us about a class.

According to the value of information gain, we split the node and build the decision tree.

A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first.

It can be calculated using the below formula:

$$\text{Information Gain} = \text{Entropy}(S) - [(\text{Weighted Avg}) * \text{Entropy}(\text{each feature})]$$

Entropy

Entropy is a metric to measure the impurity in a given attribute.

It specifies randomness in data. Entropy can be calculated as:

$$\text{Entropy}(s) = -P(\text{yes}) \log_2 P(\text{yes}) - P(\text{no}) \log_2 P(\text{no})$$

Where,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

Gini Index

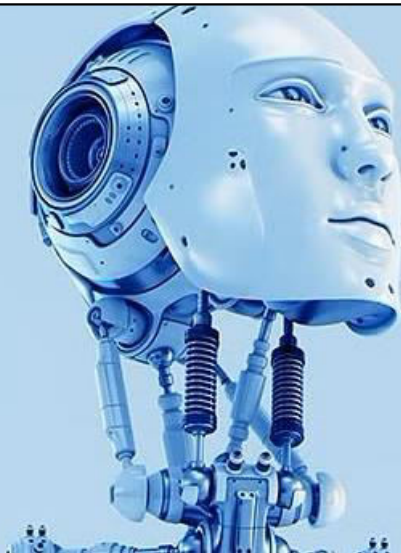
Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.

An attribute with the low Gini index should be preferred as compared to the high Gini index.

It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.

Gini index can be calculated using the below formula:

$$\text{Gini Index} = 1 - \sum_j P_j^2$$



CST 401: ARTIFICIAL
INTELLIGENCE
2019 SCHEME

MODULE 5: Decision Tree

Consider the following data set comprised of two binary input attributes (A1 and A2) and one binary output. (8)

Example	A ₁	A ₂	Output y
x ₁	1	1	1
x ₂	1	1	1
x ₃	1	0	0
x ₄	0	0	1
x ₅	0	1	0
x ₆	0	1	0

Use the DECISION-TREE-LEARNING algorithm to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

First find Entropy(S) whole dataset

$$\text{Entropy}(s) = -P(\text{yes})\log_2 P(\text{yes}) - P(\text{no})\log_2 P(\text{no})$$

$$= -\frac{3}{6}\log_2 \frac{3}{6} - \frac{3}{6}\log_2 \frac{3}{6} = 1$$

$$\text{Gain}(S, A_1) = \text{ES} - \left\{ \frac{|S_{A_1=1}|}{|S|} \text{Entropy}(S_{A_1=1}) + \frac{|S_{A_1=0}|}{|S|} \text{Entropy}(S_{A_1=0}) \right\}$$

$S_{A1=1}$

X	A1	Y
X1	1	1
X2	1	1
X3	1	0

$$|S_{A1=1}|=3$$

$$\begin{aligned} \text{Entropy}(S_{A1=1}) &= -P_1 \log_2 P_1 - P_0 \log_2 P_0 \\ &= -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.9183 \end{aligned}$$

$S_{A1=0}$

X	A1	Y
X4	0	1
X5	0	0
X6	0	0

$$|S_{A1=0}|=3$$

$$\text{Entropy}(S_{A1=0}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0.9183$$

$$\begin{aligned} \text{So Gain}(S, A_1) &= \text{ES} - \left\{ \frac{|S_{A1=1}|}{|S|} \text{Entropy}(S_{A1=1}) + \frac{|S_{A1=0}|}{|S|} \text{Entropy}(S_{A1=0}) \right\} \\ &= 1 - \left\{ \frac{3}{6} \times 0.9183 + \frac{3}{6} \times 0.9183 \right\} = 0.0817 \text{ -----(1)} \end{aligned}$$

$S_{A2=1}$

X	A2	Y
X1	1	1
X2	1	1
X5	1	0
X6	1	0

$$|S_{A2=1}|=4$$

$$\begin{aligned} \text{Entropy}(S_{A2=1}) &= -P_1 \log_2 P_1 - P_0 \log_2 P_0 \\ &= -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1 \end{aligned}$$

$S_{A2=0}$

X	A2	Y
X3	0	0
X4	0	1

$$|S_{A2=0}|=2$$

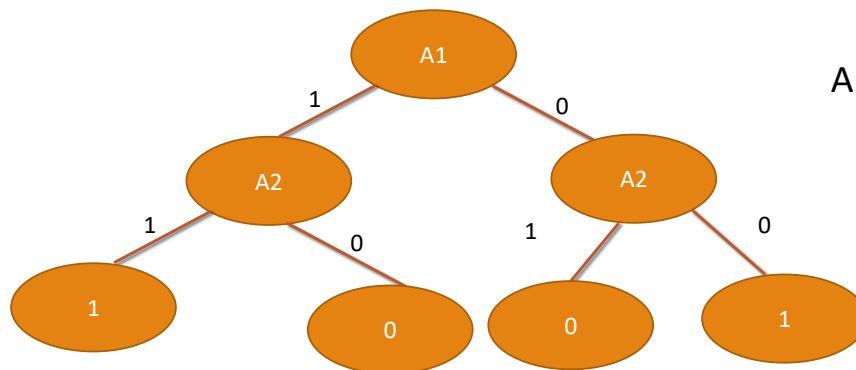
$$\text{Entropy}(S_{A2=0}) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1$$

$$\begin{aligned} \text{So Gain}(S, A_2) &= \text{ES} - \left\{ \frac{|S_{A2=1}|}{|S|} \text{Entropy}(S_{A2=1}) + \frac{|S_{A2=0}|}{|S|} \text{Entropy}(S_{A2=0}) \right\} \\ &= 1 - \left\{ \frac{4}{6} \times 1 + \frac{2}{6} \times 1 \right\} = 0 \quad \text{-----(2)} \end{aligned}$$

A1=1

X	A2	Y
X1	1	1
X2	1	1
X3	0	0

From (1) and (2) We found that A1 give more information gain.
So we make A1 as root node



A1=0

X	A2	Y
X4	0	1
X5	1	0
X6	1	0

Thank You!

See you in next video

Pruning: Getting an Optimal Decision tree

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree.

A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset.

Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning.

There are mainly two types of tree pruning technology used:

- Cost Complexity Pruning
- Reduced Error Pruning.

Advantages of the Decision Tree

It is simple to understand as it follows the same process which a human follow while making any decision in real-life.

It can be very useful for solving decision-related problems

It helps to think about all the possible outcomes for a problem.

There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

The decision tree contains lots of layers, which makes it complex.

It may have an overfitting issue, which can be resolved using the Random Forest algorithm.

For more class labels, the computational complexity of the decision tree may increase.

Overfitting in Machine Learning

In the real world, the dataset present will never be clean and perfect.

It means each dataset contains impurities, noisy data, outliers, missing data, or imbalanced data.

Due to these impurities, different problems occur that affect the accuracy and the performance of the model.

One of such problems is Overfitting in Machine Learning.

Overfitting is a problem that a model can exhibit.

A statistical model is said to be overfitted if it can't generalize well with unseen data.

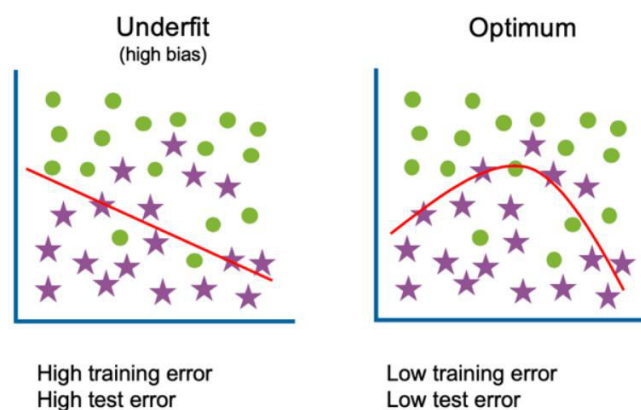
Noise: Noise is meaningless or irrelevant data present in the dataset. It affects the performance of the model if it is not removed.

Bias: Bias is a prediction error that is introduced in the model due to oversimplifying the machine learning algorithms. Or it is the difference between the predicted values and the actual values.

Variance: If the machine learning model performs well with the training dataset, but does not perform well with the test dataset, then variance occurs.

Generalization: It shows how well a model is trained to predict unseen data

What is Overfitting?



Overfitting & underfitting are the two main errors/problems in the machine learning model, which cause poor performance in Machine Learning.

Overfitting occurs when the model fits more data than required, and it tries to capture each and every datapoint fed to it. Hence it starts capturing noise and inaccurate data from the dataset, which degrades the performance of the model.

An overfitted model doesn't perform accurately with the test/unseen dataset and can't generalize well.

An overfitted model is said to have low bias and high variance.

Example

Suppose there are three students, X, Y, and Z, and all three are preparing for an exam.

X has studied only three sections of the book and left all other sections.

Y has a good memory, hence memorized the whole book.

And the third student, Z, has studied and practiced all the questions.

So, in the exam, X will only be able to solve the questions if the exam has questions related to section 3.

Student Y will only be able to solve questions if they appear exactly the same as given in the book.

Student Z will be able to solve all the exam questions in a proper way.

The same happens with machine learning;

- if the algorithm learns from a small part of the data, it is unable to capture the required data points and hence under fitted.
- Suppose the model learns the training dataset, like the Y student. They **perform very well on the seen dataset but perform badly on unseen data or unknown instances**. In such cases, the model is said to be Overfitting.
- And if the model performs well with the training dataset and also with the test/unseen dataset, similar to student Z, it is said to be a good fit.

How to detect Overfitting?

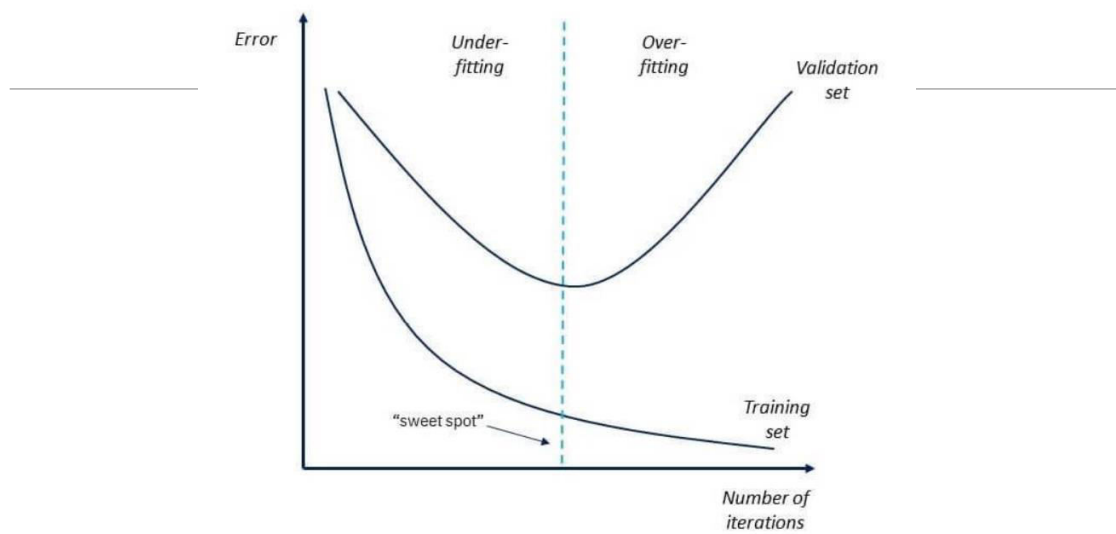
Overfitting in the model can only be detected once you test the data. To detect the issue, we can perform Train/test split.

In the train-test split of the dataset, we can divide our dataset into random test and training datasets.

We train the model with a training dataset which is about 80% of the total dataset. After training the model, we test it with the test dataset, which is 20 % of the total dataset.

Now, if the model performs well with the training dataset but not with the test dataset, then it is likely to have an overfitting issue.

For example, if the model shows 85% accuracy with training data and 50% accuracy with the test dataset, it means the model is not performing well.



Ways to prevent the Overfitting

Although overfitting is an error in Machine learning which reduces the performance of the model, however, we can prevent it in several ways.

With the use of the linear model, we can avoid overfitting; however, many real-world problems are non-linear ones. Below are several ways that can be used to prevent overfitting:

1. Early Stopping
2. Train with more data
3. Feature Selection
4. Cross-Validation
5. Data Augmentation
6. Regularization

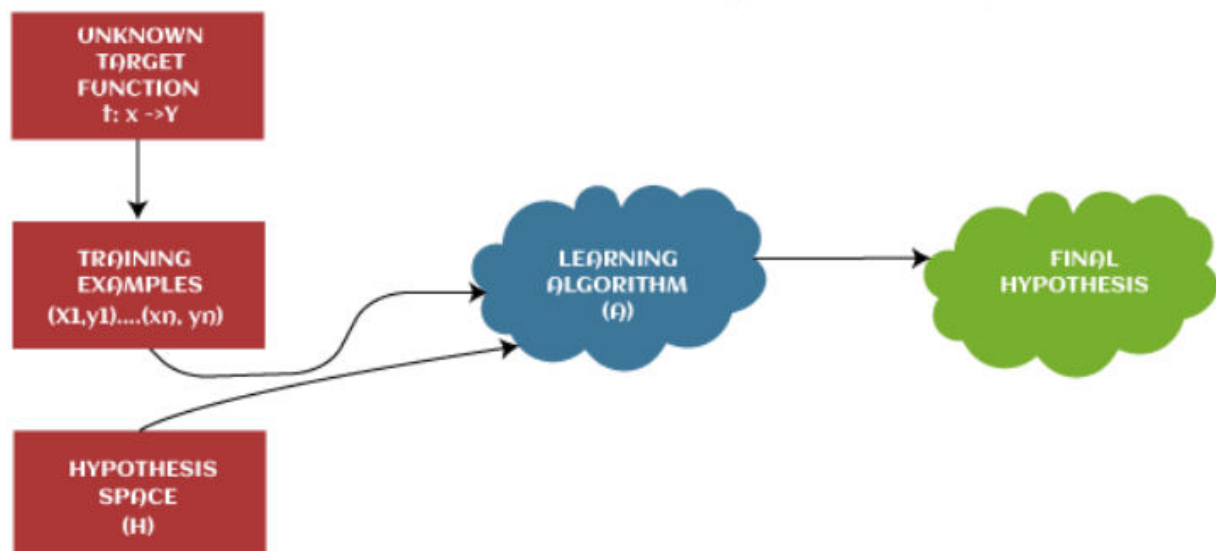
Hypothesis in Machine Learning

The hypothesis is defined as the supposition or proposed explanation based on insufficient evidence or assumptions

It is just a guess based on some known facts but has not yet been proven. A good hypothesis is testable, which results in either true or false.

Example: Let's understand the hypothesis with a common example. Some scientist claims that ultraviolet (UV) light can damage the eyes then it may also cause blindness.

In this example, a scientist just claims that UV rays are harmful to the eyes, but we assume they may cause blindness. However, it may or may not be possible. Hence, these types of assumptions are called a hypothesis.



In supervised learning techniques, the main aim is to determine the possible hypothesis out of hypothesis space that best maps input to the corresponding or correct outputs.

There are some common methods given to find out the possible hypothesis from the Hypothesis space, where hypothesis space is represented by uppercase-h (H) and hypothesis by lowercase-h (h).

Hypothesis space (H):

Hypothesis space is defined as a set of all possible legal hypotheses; hence it is also known as a hypothesis set.

It is used by supervised machine learning algorithms to determine the best possible hypothesis to describe the target function or best maps input to output.

It is often constrained by choice of the framing of the problem, the choice of model, and the choice of model configuration

Hypothesis (h)

It is defined as the approximate function that best describes the target in supervised machine learning algorithms. It is primarily based on data as well as bias and restrictions applied to data.

Hence hypothesis (h) can be concluded as a single hypothesis that maps input to proper output and can be evaluated as well as used to make predictions.

The hypothesis (h) can be formulated in machine learning as follows: $y = mx + b$

Where,

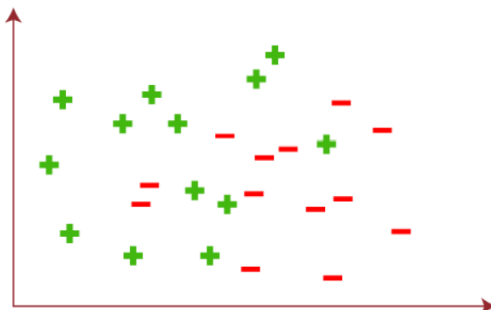
Y: Range

m: Slope of the line which divided test data or changes in y divided by change in x.

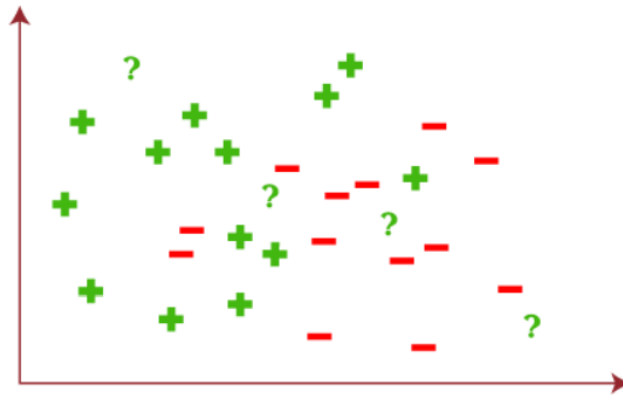
x: domain

c: intercept (constant)

Example



Now, assume we have some test data by which ML algorithms predict the outputs for input as follows



91

REGRESSION AND CLASSIFICATION WITH LINEAR MODELS

linear functions of continuous-valued inputs

regression with a univariate linear function, otherwise known as “fitting a straight line.”

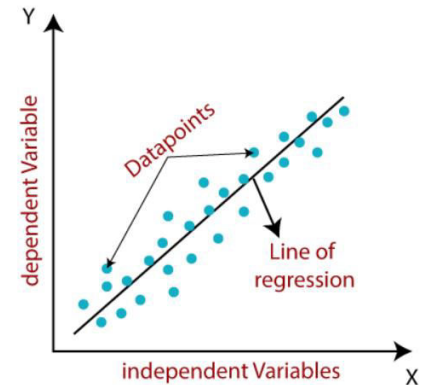
Linear Regression in Machine Learning

Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price**, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression.

Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables.



Mathematically, we can represent a linear regression as:

$$y = a_0 + a_1x + \epsilon$$

Here,

Y= Dependent Variable (Target Variable)

X= Independent Variable (predictor Variable)

a_0 = intercept of the line (Gives an additional degree of freedom)

a_1 = Linear regression coefficient (scale factor to each input value).

ϵ = random error

Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

- **Simple Linear Regression:**

If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

- **Multiple Linear regression:**

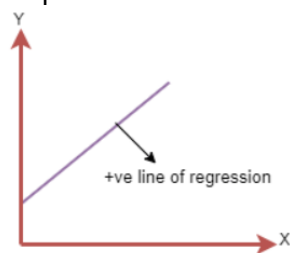
If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Linear Regression Line

A linear line showing the relationship between the dependent and independent variables is called a **regression line**. A regression line can show two types of relationship:

- **Positive Linear Relationship:**

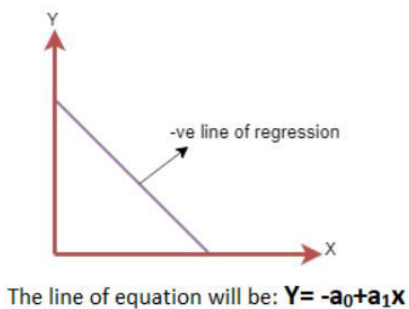
If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



The line equation will be: $Y = a_0 + a_1X$

Negative Linear Relationship:

If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.



Finding the best fit line:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized.

The best fit line will have the least error.

The different values for weights or the coefficient of lines (a_0 , a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost function.

Cost function

The different values for weights or coefficient of lines (a_0 , a_1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.

Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.

We can use the cost function to find the accuracy of the mapping function, which maps the input variable to the output variable. This mapping function is also known as Hypothesis function.

For Linear Regression, we use the **Mean Squared Error (MSE)** cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

$$MSE = \frac{1}{N} \sum_{i=1}^n (y_i - (a_1 x_i + a_0))^2$$

Where,

N = Total number of observation

Y_i = Actual value

$(a_1 x_i + a_0)$ = Predicted value.

Gradient Descent

Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.

A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.

It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

Model Performance

The Goodness of fit determines how the line of regression fits the set of observations. The process of finding the best model out of various models is called **optimization**. It can be achieved by **R-squared method**

R-squared method:

R-squared is a statistical method that determines the goodness of fit.

It measures the strength of the relationship between the dependent and independent variables on a scale of 0-100%.

The high value of R-square determines the less difference between the predicted values and actual values and hence represents a good model.

It is also called a coefficient of determination, or coefficient of multiple determination for multiple regression.

It can be calculated from the below formula:

$$\text{R-squared} = \frac{\text{Explained variation}}{\text{Total Variation}}$$

Multiple Linear Regression

In simple Linear Regression a single Independent/Predictor(X) variable is used to model the response variable (Y).

But there may be various cases in which the response variable is affected by more than one predictor variable; for such cases, the Multiple Linear Regression algorithm is used.

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable.

Example: Prediction of CO₂ emission based on engine size and number of cylinders in a car.

Some key points about MLR:

- For MLR, the dependent or target variable(Y) must be the continuous/real, but the predictor or independent variable may be of continuous or categorical form.
- Each feature variable must model the linear relationship with the dependent variable.
- MLR tries to fit a regression line through a multidimensional space of data-points.

MLR equation

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon$$

where, for $i = n$ observations:

y_i = dependent variable

x_i = explanatory variables

β_0 = y-intercept (constant term)

β_p = slope coefficients for each explanatory variable

ϵ = the model's error term (also known as the residuals)

Assumptions for Multiple Linear Regression:

A linear relationship should exist between the Target and predictor variables.

The regression residuals must be normally distributed.

MLR assumes little or no multicollinearity (correlation between the independent variable) in data.



Thank You!

MODULE 5 ENDS

THANK YOU