

T86

Page: 15
April: 2024

ScoringIndicators

COURSENAME:FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

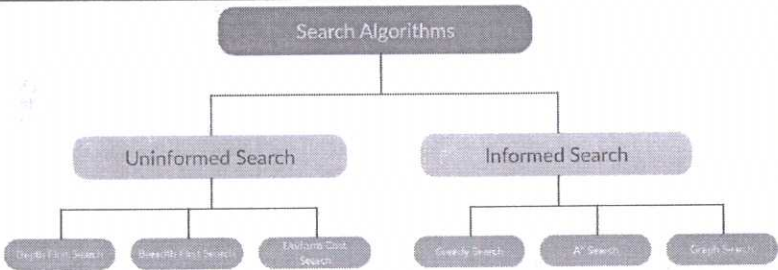
COURSECODE:5133C

QID:2109230304A

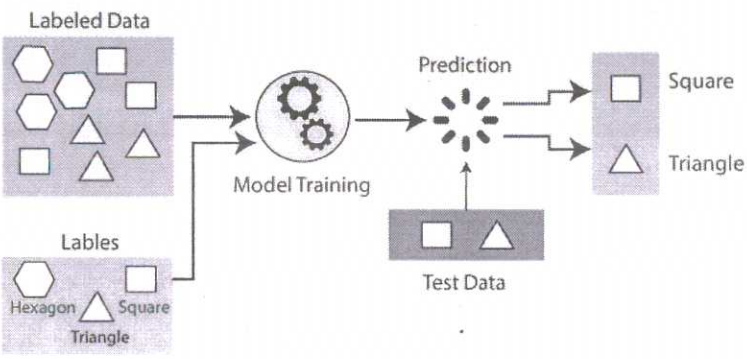
QNo	ScoringIndicators	Split score	Sub Total	Total score
	PARTA			9
I.1	Machine learning is a subset of AI, which enables the machine to automatically learn from data,improve performance from past experiences and make predictions..	1	1	
I.2	TensorFlow,PyTorch	1/2×2	1	
I.3	Sequence Types	1	1	
I.4	0 1 2 3	1	1	
I.5	def keyword	1	1	
I.6	Clustering and association	1/2×2	1	
I.7	Number of iterations	1	1	
I.8	Min/Max algorithm	1	1	
I.9	Competitive environment	1	1	
	PARTB			24
II.1	An AI model is a program or algorithm that relies on training data to recognize patterns and make predictions or decisions. Common AI Models <ul style="list-style-type: none">• Deep neural networks• Linear regression• Logistic regression• Decision trees• Random forest Deep neural networks The deep neural network is one of the most popular AI/ML models. The design for this deep learning model was inspired	1	3	

	<p>by the human brain and its neural network. This AI model uses layers of artificial neurons to combine multiple inputs and provide a single output value. Hence the name, deep learning.</p> <p>Linear Regression</p> <p>This AI model is very popular with data scientists working in statistics. Linear regression is based on a supervised learning model. These AI models are tasked with identifying the relationship between input and output variables.</p> <p>(Any two)</p>	1+1		
II.2	<p>1. Artificial intelligence and machine learning is a versatile discipline</p> <p>2. Artificial intelligence and machine learning is the skill of the century</p> <p>3. Artificial intelligence and machine learning is capable of ingesting a huge amount of data</p> <p>(Any Three)</p>	1+1+1	3	
II.3	<p>Python features</p> <ol style="list-style-type: none"> 1) Easy to Learn and Use 2) Expressive Language 3) Interpreted Language 4) Cross-platform Language 5) Free and Open Source 6) Object-Oriented Language <p>(Any Six)</p>	1/2×6	3	
II.4	<p>Fibonacci series between 0 to 50</p> <pre>x=0 y=1 while y<50: print(y) x=y y = x+y</pre>	3	3	
II.5	<p>The search() Function</p> <p>The search() function searches the string for a match, and</p>	1+1+1	3	

	<p>returns a Match object if there is a match.If there is more than one match, only the first occurrence of the match will be returned:</p> <p>Match Object</p> <p>A Match Object is an object containing information about the search and the result.</p> <p>The findall() Function</p> <p>The findall() function returns a list containing all matches.</p> <p>(Explain with example)</p>			
II.6	<p>There are two useful methods in Pandas:</p> <ul style="list-style-type: none"> • IsNull() and dropna() will help to find the columns/rows with missing data and drop them • Fillna() will replace the wrong values with a placeholder value 	1.5+1.5	3	
II.7	<p>The classifier is called 'naive' because it makes assumptions that may or may not turn out to be correct.</p> <p>The algorithm assumes that the presence of one feature of a class is not related to the presence of any other feature (absolute independence of features), given the class variable.</p> <p>For instance, a fruit may be considered to be a cherry if it is red in color and round in shape, regardless of other features. This assumption may or may not be right (as an apple also matches the description).</p>	3	3	
II.8	<p>Classification is used when your target is categorical, while regression is used when your target variable is continuous. Both classification and regression belong to the category of supervised machine learning algorithms.</p>	1.5+1.5	3	

	<p>Examples of classification problems include:</p> <ul style="list-style-type: none"> • Predicting yes or no • Breed of an animal • Type of color <p>Examples of regression problems include:</p> <ul style="list-style-type: none"> • Estimating sales and price of a product • Predicting the score of a team • Predicting the amount of rainfall 			
<p>II.9</p>	 <pre> graph TD SA[Search Algorithms] --> US[Uninformed Search] SA --> IS[Informed Search] US --> DFS[Depth-First Search] US --> BFS[Breadth-First Search] US --> ECS[Exhaustive Cost Search] IS --> GS[Greedy Search] IS --> A*[A* Search] IS --> G[Graph Search] </pre> <p>DFS Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking. It uses last in- first-out strategy and hence it is implemented using a stack.</p> <p>BFS Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level. It is implemented using a queue.</p> <p>(Any two)</p>	<p>1.5+1.5</p>	<p>3</p>	

II.10	<p>Properties of Mini-Max algorithm:</p> <ul style="list-style-type: none"> ○ Complete- Min-Max algorithm is Complete. It will definitely find a solution (if exist), in the finite search tree. ○ Optimal- Min-Max algorithm is optimal if both opponents are playing optimally. ○ Time complexity- As it performs DFS for the game-tree, so the time complexity of Min-Max algorithm is $O(b^m)$, where b is branching factor of the game-tree, and m is the maximum depth of the tree. ○ Space Complexity- Space complexity of Mini-max algorithm is also similar to DFS which is $O(bm)$. <p>(Any three)</p>	1+1+1	3	
PART C				42
III	<p>Supervised machine learning</p> <ul style="list-style-type: none"> • Supervised machine learning is based on supervision. • It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. • Here, the labelled data specifies that some of the inputs are already mapped to the output. • More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset. <p>Example</p> <ul style="list-style-type: none"> • Suppose we have an input dataset of cats and dog images. • So, first, we will provide the training to the machine to understand the images, such as the shape & size of the tail of cat and dog, Shape of eyes, colour, height (dogs are taller, cats are smaller), etc. • After completion of training, we input the picture of a cat and ask the machine to identify the object and predict the output. • Now, the machine is well trained, so it will check all the features of the object, such as height, shape, colour, eyes, ears, tail, etc., and find that it's a cat. • So, it will put it in the Cat category. • This is the process of how the machine identifies the objects in Supervised Learning. 	5+2	7	7

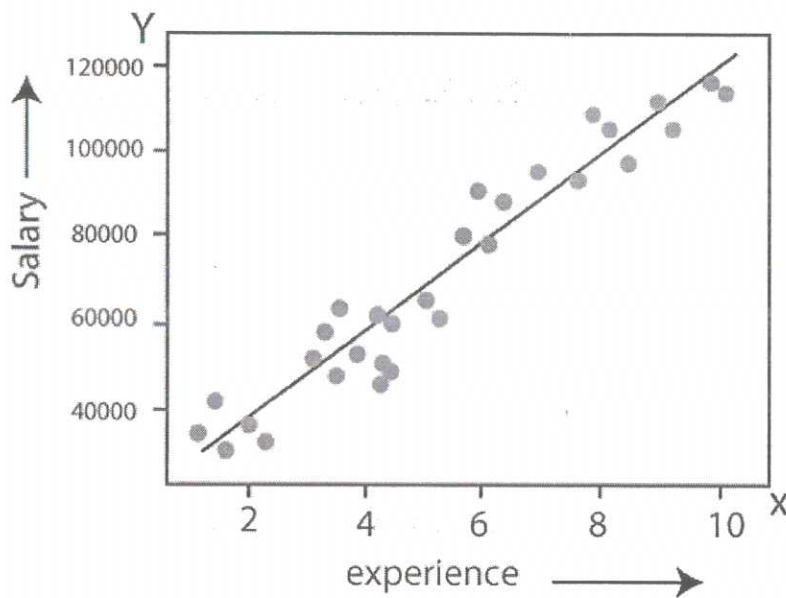
	 <p>Some real-world applications of supervised learning are Risk Assessment, Fraud Detection, Spam filtering, etc (Explanation-5, Figure-2)</p>			
IV	<ul style="list-style-type: none"> Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next. After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect. It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error. <p>Example:</p> <p>The problem is as follows: We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The goal of the robot is to get the reward that is the diamond and avoid the hurdles that are fired. The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles. Each right step will give the robot a reward and each wrong step will subtract the reward of the robot. The total reward will be calculated when it reaches the final reward that is the diamond.</p> <p>(Explanation-4, Example-3)</p>	4+3	7	7
V	<p>a) <code>get()</code> The <code>get()</code> method returns the value of the item with the specified key.</p> <p>Example: <code>car = {</code> <code> "brand": "Ford",</code></p>	2+2+2+1	7	

	<pre> "model": "Mustang", "year": 1964 } x = car.get("price", 15000) print(x) b) pop() The pop() method removes the item with the specified key name: Example thisdict = { "brand": "Ford", "model": "Mustang", "year": 1964} thisdict.pop("model") print(thisdict) c) update() The update() method will update the dictionary with the items from the given argument. The argument must be a dictionary, or an iterable object with key:value pairs. thisdict = { "brand": "Ford", "model": "Mustang", "year": 1964 } thisdict.update({"year": 2020}) d) values() This method is used to return values of a dictionary: for x in thisdict.values(): print(x) (Explain with example) </pre>			
<p>VI</p>	<p>Package in Python is a folder that contains various modules as files.</p> <p>Creating Package: create a package in Python named mypkg that will contain two modules mod1 and mod2. To create this module follow the below steps:</p>	<p>1</p> <p>2</p>	<p>7</p>	<p>7</p>

	<ul style="list-style-type: none"> • Create a folder named mypkg. • Inside this folder create an empty Python file i.e. <code>__init__.py</code> • Then create two modules mod1 and mod2 in this folder <p>Download a Package</p> <p>Downloading a package is very easy. Open the command line interface and tell PIP to download the package you want. Navigate your command line to the location of Python's script directory, and type the following:</p> <pre>>pip install camelcase</pre> <p>Using a Package</p> <p>Once the package is installed, it is ready to use. import the "camelcase" package into your project. Import and use "camelcase":</p> <pre>import camelCase c = camelcase.CamelCase() txt = "hello world" print(c.hump(txt))</pre> <p>Find Packages</p> <p>Find more packages at https://pypi.org/.</p> <p>List Packages</p> <p>Use the list command to list all the packages installed on your system:</p> <p>Example: List installed packages:</p> <pre>C:\>pip list</pre>	1		
VII	<p>In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming. The main concept of OOPs is to bind the data and the functions that work on that together as a single unit so</p>	1	7	7

	<p>that no other part of the code can access this data.</p> <p>OOPs Concepts in Python</p> <ul style="list-style-type: none"> • Class • Objects • Polymorphism • Encapsulation • Inheritance • Data Abstraction <p>(Explain each one)</p>	1×6		
VIII	<pre>def string_reverse(str1): rstr1 = '' index = len(str1) while index > 0: rstr1 += str1[index - 1] index = index - 1 return rstr1 print(string_reverse('hello'))</pre> <p>(function def-2, loop-3, function call-2)</p>	2+3+2	7	7
IX	<p>Step 1: getting datasets</p> <p>Step 2: importing libraries</p> <p>Step 3: Importing packages</p> <p>Step 4: Missing value elimination</p> <p>Step 5: Feature Scaling</p> <p>Step 6: Splitting the data</p> <p>(Explain each one)</p>	7	7	7
X	<p>Clustering</p> <p>It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.</p> <p>Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.</p> <p>Example: 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</p>	5+2	7	7

	<p>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. The clustering technique also works in the same way.</p> <p>Some most common uses of this technique are:</p> <ul style="list-style-type: none"> ○ Market Segmentation ○ Statistical data analysis ○ Social network analysis ○ Image segmentation ○ Anomaly detection, etc. <p>(Explanation-5,Example-2)</p>			
<p>XI</p>	<p>Linear Regression:</p> <ul style="list-style-type: none"> ○ 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. ○ The relationship between variables in the linear regression model can be explained using the below image. Here we are predicting the salary of an employee on the basis of the year of experience. 	<p>5+2</p>	<p>7</p>	<p>7</p>



- Below is the mathematical equation for Linear regression:

- $Y = aX + b$

Here, Y = dependent variables (target variables),
 X = Independent variables (predictor variables),
 a and b are the linear coefficients

Some popular applications of linear regression are:

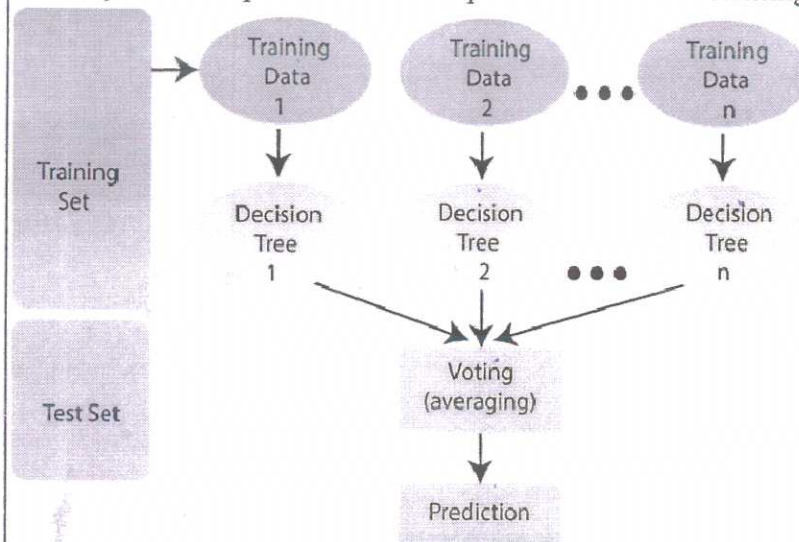
- Analyzing trends and sales estimates
- Salary forecasting
- Real estate prediction
- Arriving at ETAs in traffic.

(Explanation-5, Figure-2)

XII	<p>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.</p> <p>"Random Forest is a classifier that contains a number of</p>	5+2	7	7
-----	---	-----	---	---

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.



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.

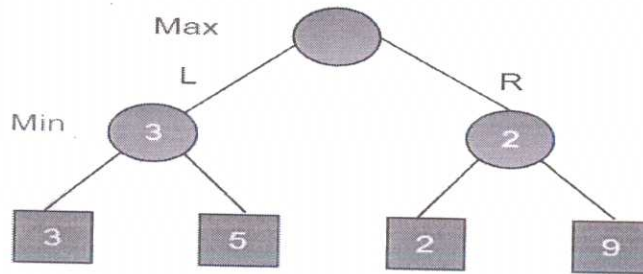
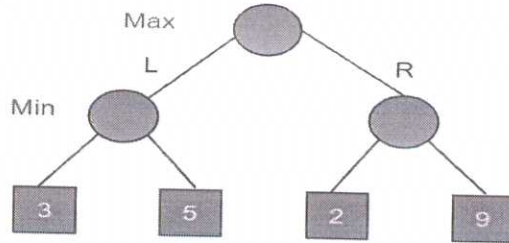
Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

1. Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
2. Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.
3. Land Use: We can identify the areas of similar land use

	<p>by this algorithm.</p> <p>4. Marketing: Marketing trends can be identified using this algorithm.</p> <p>(Explanation-5,Figure-2)</p>			
XIII	<ul style="list-style-type: none"> • Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory. It provides an optimal move for the player assuming that opponent is also playing optimally. • Mini-Max algorithm uses recursion to search through the game-tree. • Min-Max algorithm is mostly used for game playing in AI. Such as Chess, Checkers, tic-tac-toe, go, and various tow-players game. This Algorithm computes the minimax decision for the current state. • In this algorithm two players play the game, one is called MAX and other is called MIN. • Both the players fight it as the opponent player gets the minimum benefit while they get the maximum benefit. • Both Players of the game are opponent of each other, where MAX will select the maximized value and MIN will select the minimized value. • The minimax algorithm performs a depth-first search algorithm for the exploration of the complete game tree. • The minimax algorithm proceeds all the way down to the terminal node of the tree, then backtrack the tree as the recursion. • The working of the minimax algorithm can be easily described using an example. Below we have taken an example of game-tree which is representing the two-player game. • In this example, there are two players one is called Maximizer and other is called Minimizer. • Maximizer will try to get the Maximum possible score, and Minimizer will try to get the minimum possible score. • This algorithm applies DFS, so in this game-tree, we have to go all the way through the leaves to reach the terminal nodes. 	4+3	7	7

- At the terminal node, the terminal values are given so we will compare those value and backtrack the tree until the initial state occurs.



Being the maximizer you would choose the larger value that is 3. Hence the optimal move for the maximizer is to go LEFT and the optimal value is 3.
(Explanation-4,Example-3)

XIV

```
# Python3 program to find winner of game
# if player can pick 1, x, y coins
# To find winner of game
def findWinner(x, y, n):
# To store results
dp = [0 for i in range(n + 1)]
# Initial values
dp[0] = False
dp[1] = True
# Computing other values.
for i in range(2, n + 1):
# If A losses any of i-1 or i-x
```

7

7

7

```
# or i-y game then he will
# definitely win game i
if (i - 1 >= 0 and not dp[i - 1]):
    dp[i] = True
elif (i - x >= 0 and not dp[i - x]):
    dp[i] = True
elif (i - y >= 0 and not dp[i - y]):
    dp[i] = True
# Else A loses game.
else:
    dp[i] = False
# If dp[n] is true then A will
# game otherwise he losses
return dp[n]
# Driver Code
x = 3; y = 4; n = 100
if (findWinner(x, y, n)):
    print('A wins')
else:
    print('B wins')
```