T.Y. BSC. (CS): Sem. V (Elective I)

Artificial Intelligence

Mumbai University Examination Paper Solution: Nov-22

Q.P. Code:13494

Q1. Attempt All

(a) Multiple choice Questions **[10]**

I. An AI system is composed of?
- **A.** Agent
- **B.** Environment
- **C.** Both A and B
- **D.** None

Ans: C. Both A and B

II. Which of the following is not a type of agents in artificial intelligence?
- A. Utility
- B. Model
- C. Simple
- D. Target

Ans: D. Target

III. Rationality of an agent does not depend on?
- A. Performance measure
- B. Percept sequence
- C. Reaction
- D. Action

Ans: C. Reaction

IV. In which ANN, loops are allowed?
- A. Feedforward ANN
- B. Feedback ANN
- C. Both A and B
- D. None

Ans: B. Feedback ANN

V. What is purpose of Axon?
- A. Receptors
- B. Transmission
- C. Transmitter
- D. None of the above

Ans: B. Transmission

VI. The _____ for an agent specifies the action taken by the agent in response to any percept sequence.

A. Agent function
B. Agent program
C. Agent Structure
D. None of the above

Ans: A. Agent function

VII. To pass the total Turing test the computer will need _____

A. Robotics
B. Computer Vision
C. Both A and B
D. None of the above

Ans: C. Both A and B

VIII. _____ expands shallowest unexpanded node first.

A. Breadth First Search
B. Depth First Search
C. IDA
D. A*

Ans: A. Breadth First Search

IX. The _____, which determines whether a given state is a goal state.

A. Goal test
B. Path test
C. Agent test
D. None of the above

Ans: A. Goal Test

X. How the decision tree reaches its decision?

A. Single test
B. Two test
C. Sequence of tests
D. No test

Ans: C. Sequence of tests

(b) Fill in the blanks (lm, Multiple Linear Model, Problem, Solution, Activation, Evidence [5]

1. A search algorithm takes Problem as an input and returns Solution as an output.
2. Function used for linear regression in python is lm
3. When there is more than one independent variable in the model, then the linear model is termed as Multiple Linear Model.
4. The process of adjusting the weight in Neural Network is known as Activation.
5. In Bayes Theorem, unconditional probability is called as Evidence.

2. Attempt any three of the following **[15]**

Q2(a) What are Uninformed Strategies? Explain any one in detail. **(5)**

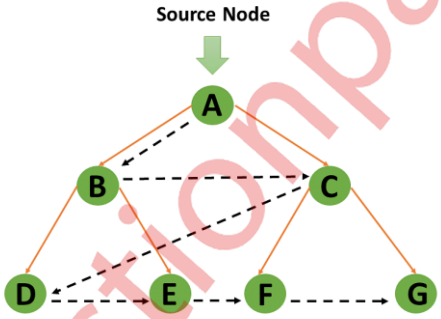
Ans. 1. Uninformed search, also known as blind search, is a search algorithm that explores a problem space without any specific knowledge or information about the problem other than the initial state and the possible actions to take.

2. These algorithms are typically less efficient than informed search algorithms but can be useful in certain scenarios or as a basis for more advanced search techniques.

3. The different types of uninformed search algorithms used in AI are as follows:

- Depth First Search
- Breadth-First Search
- Depth Limited Search

4. Breadth-first Search:



a) Breadth-first search is the most common search strategy for traversing a tree or graph. This algorithm searches breadthwise in a tree or graph, so it is called breadth-first search.

b) BFS algorithm starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level.

c) Breadth-first search implemented using FIFO queue data structure.

d) Advantages:

- BFS will provide a solution if any solution exists.
- If there are more than one solutions for a given problem, then BFS will provide the minimal solution which requires the least number of steps.

e) Disadvantages:

- It requires lots of memory since each level of the tree must be saved into memory to expand the next level.
- BFS needs lots of time if the solution is far away from the root node.

f) Space Complexity: Space complexity of BFS algorithm is given by the Memory size of frontier which is O(bd).

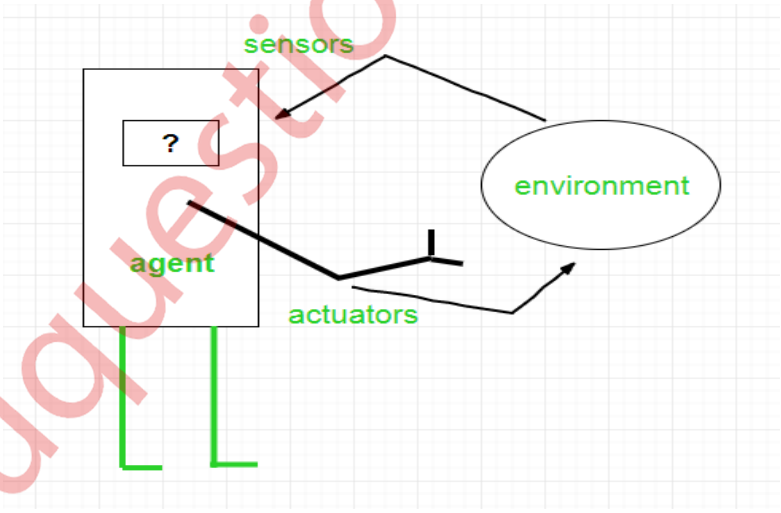
g) Completeness: BFS is complete, which means if the shallowest goal node is at some finite depth, then BFS will find a solution.

h) Optimality: BFS is optimal if path cost is a non-decreasing function of the depth of the node.

Q2(b) Explain Agent Structure in detail. **(5)**

Ans. 1. In artificial intelligence, an agent is a computer program or system that is designed to perceive its environment, make decisions and take actions to achieve a specific goal or set of goals. The agent operates autonomously, meaning it is not directly controlled by a human operator.

2. Agents can be classified into different types based on their characteristics, such as whether they are reactive or proactive, whether they have a fixed or dynamic environment, and whether they are single or multi-agent systems.



3. Artificial intelligence is defined as the study of rational agents. A rational agent could be anything that makes decisions, such as a person, firm, machine, or software. It carries out an action with the best outcome after considering past and current percepts (agent's perceptual inputs at a given instance).

4. Structure of an AI Agent

Agent = Architecture + Agent Program

5. There are many examples of agents in artificial intelligence. Here are a few:

- Intelligent personal assistants: These are agents that are designed to help users with various tasks, such as scheduling appointments, sending messages, and setting reminders. Examples of intelligent personal assistants include Siri, Alexa, and Google Assistant.
- Autonomous robots: These are agents that are designed to operate autonomously in the physical world. They can perform tasks such as cleaning, sorting, and delivering goods. Examples of autonomous robots include the Roomba vacuum cleaner and the Amazon delivery robot.

6. Types of Agents:

- Simple Reflex Agents
- Model-Based Reflex Agents
- Goal-Based Agents
- Utility-Based Agents
- Learning Agent

Q2(c) What is well defined problem and solution explain for Romanian map? (5)

Ans. 1. A well-defined problem is a problem that has a clear and unambiguous statement, a defined goal, a specified set of constraints, and a well-understood problem domain. It is characterized by having all the necessary information and criteria required to find a solution.

2. In the context of a Romanian map, let's consider the following well-defined problem and its solution:

- Problem: Find the shortest route between two cities on the Romanian map.
- Solution: The solution to this problem involves using a pathfinding algorithm, such as Dijkstra's algorithm or A* search algorithm, to determine the shortest route between the given pair of cities.

3. Here's a step-by-step explanation of the solution process:

a) Representation: The Romanian map needs to be represented as a graph, where cities

are represented as nodes, and the connections between cities (roads) are represented as edges. Each edge has a weight or cost associated with it, representing the distance or travel time between the connected cities.

b) Input: Identify the starting city and the destination city for which you want to find the shortest route.

c) Initialization: Initialize the algorithm by assigning a distance value of 0 to the starting city and infinity to all other cities. Set the starting city as the current city.

d) Exploration: Iterate through the cities based on the selected pathfinding algorithm, considering the neighboring cities connected by edges. Update the distance values of the neighboring cities if a shorter path is found. Keep track of the path taken to reach each city.

e) Goal Test: Once the destination city is reached, the algorithm terminates, as the shortest route has been found.

f) Traceback: From the destination city, trace back the path taken by following the cities' predecessors recorded during the exploration phase. This will give you the sequence of cities representing the shortest route.

g) Output: The output is the shortest route, which includes the sequence of cities to be visited from the starting city to the destination city, along with the total distance or cost associated with the route.

4. By applying a suitable pathfinding algorithm to the well-defined problem of finding the shortest route between two cities on the Romanian map, you can obtain an optimal solution that minimizes travel distance or time.

Q2(d) What is PEAS? Explain with two suitable examples. (5)

Ans. 1. PEAS stands for Performance measure, Environment, Actuators, and Sensors. It is a framework used in artificial intelligence to analyze and understand the behavior and characteristics of intelligent agents

2. Performance − If the agent's performance is being evaluated by an objective function. Things that we can use to measure an agent's performance.

3.Environment − The environment refers to the agent's immediate surroundings at the time the agent is working in that environment. Depending on the mobility of the agent, it might be static or dynamic.

4.Actuators − Agents rely on actuators to function in their surroundings. Display boards,

object-picking arms, track-changing devices, etc. are examples of actuators.

5.Sensors − By providing agents with a comprehensive collection of Inputs, sensors enable them to comprehend their surroundings. Agent behavior is influenced by their recent past and their present input set. Various sensing devices, such as cameras, GPS, odometers, and others, are examples of sensors.

a.) PEAS Descriptor of Automated Car Driver

- Performance

  Safety − The automated system needs to be able to operate the vehicle securely without rushing.

  Optimized Speed − Depending on the environment, automated systems should be able to maintain the ideal speed.

- Environment

  Roads − Automated automobile drivers ought to be able to go on any type of route, from local streets to interstates.

  Traffic Conditions − For various types of roadways, there are various traffic conditions to be found.

- Actuators

  Steering wheel − to point an automobile in the appropriate direction.

  Gears and accelerators − adjusting the car's speed up or down.

- Sensors

In-car driving tools like cameras, sonar systems. are used to collect environmental data.

a) PEAS Descriptor of Medical Diagnosis System:

- Performance:

  Healthy Patients, minimize cost, lawsuits.

- Environment:

  Patients, Hospital Staff

- Actuators

  Display questions, test, diagnosis, treatments, referrals.

- Sensors

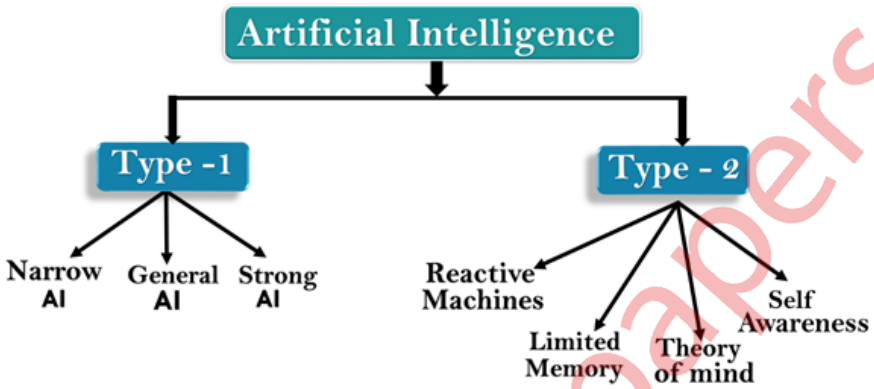
  Keyboard entry of symptoms, patient answers.

Q2(e) List and explain the categories of definitions of AI. **(5)**

Ans. 1. John McCarthy in 1955, defined It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence.

2. Types of Artificial Intelligence:

Artificial Intelligence can be divided in various types, there are mainly two types of main categorization which are based on capabilities and based on functionally of AI. Following is flow diagram which explain the types of AI.



AI type-1: Based on Capabilities

1. Weak AI or Narrow AI:
   - Narrow AI is a type of AI which is able to perform a dedicated task with intelligence. The most common and currently available AI is Narrow AI in the world of Artificial Intelligence.
   - Some Examples of Narrow AI are playing chess, purchasing suggestions on e-commerce site, self-driving cars, speech recognition, and image recognition.

2. General AI:
   - General AI is a type of intelligence which could perform any intellectual task with efficiency like a human.
   - Currently, there is no such system exist which could come under general AI and can perform any task as perfect as a human.

3. Super AI:
   - Super AI is a level of Intelligence of Systems at which machines could surpass human intelligence, and can perform any task better than human with cognitive properties. It is an outcome of general AI.
   - Super AI is still a hypothetical concept of Artificial Intelligence. Development of such systems in real is still world changing task.

Artificial Intelligence type-2: Based on functionality

1. Reactive Machines

- Purely reactive machines are the most basic types of Artificial Intelligence.
- Such AI systems do not store memories or past experiences for future actions.
- Google's AlphaGo is also an example of reactive machines.

2. Limited Memory

- Limited memory machines can store past experiences or some data for a short period of time.
- Self-driving cars are one of the best examples of Limited Memory systems. These cars can store recent speed of nearby cars, the distance of other cars, speed limit, and other information to navigate the road.

3.Theory of Mind

- Theory of Mind AI should understand the human emotions, people, beliefs, and be able to interact socially like humans.
- This type of AI machines are still not developed, but researchers are making lots of efforts and improvement for developing such AI machines.

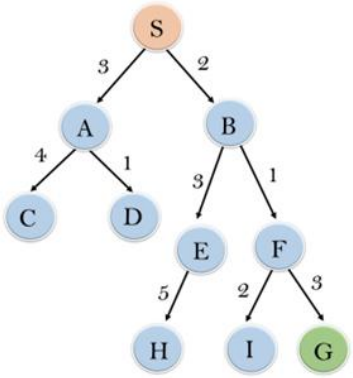
4. Self-Awareness

- Self-awareness AI is the future of Artificial Intelligence. These machines will be super intelligent, and will have their own consciousness, sentiments, and self-awareness.
- Self-Awareness AI does not exist in reality still and it is a hypothetical concept.

Q2(f) Write a Short note on Greedy Best First Search. (5)

Ans. 1. Greedy Best-First Search is an AI search algorithm that attempts to find the most promising path from a given starting point to a goal. It prioritizes paths that appear to be the most promising, regardless of whether or not they are actually the shortest path.

2.The algorithm works by evaluating the cost of each possible path and then expanding the path with the lowest cost. This process is repeated until the goal is reached.



node	H (n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

Advantages of Greedy Best-First Search:

- Simple and Easy to Implement: Greedy Best-First Search is a relatively straightforward algorithm, making it easy to implement.
- Fast and Efficient: Greedy Best-First Search is a very fast algorithm, making it ideal for applications where speed is essential.
- Low Memory Requirements: Greedy Best-First Search requires only a small amount of memory, making it suitable for applications with limited memory.
- Flexible: Greedy Best-First Search can be adapted to different types of problems and can be easily extended to more complex problems.

Disadvantages of Greedy Best-First Search:

- Inaccurate Results: Greedy Best-First Search is not always guaranteed to find the optimal solution, as it is only concerned with finding the most promising path.
- Local Optima: Greedy Best-First Search can get stuck in local optima, meaning that the path chosen may not be the best possible path.
- Lack of Completeness: Greedy Best-First Search is not a complete algorithm, meaning it may not always find a solution if one is exists. This can happen if the algorithm gets stuck in a cycle or if the search space is a too much complex.

Applications of Greedy Best-First Search:

- Pathfinding: Greedy Best-First Search is used to find the shortest path between two points in a graph. It is used in many applications such as video games, robotics, and navigation systems.
- Machine Learning: Greedy Best-First Search can be used in machine learning

algorithms to find the most promising path through a search space.

- Optimization: Greedy Best-First Search can be used to optimize the parameters of a system in order to achieve the desired result.

Q3. Attempt any three of the following **[15]**

Q3(a) Write a short note on overfitting in decision tree. **(5)**

Ans. 1. Overfitting is a phenomenon that can occur when constructing decision trees, and it refers to a situation where the tree model becomes excessively complex and specific to the training data, resulting in poor performance on new, unseen data.

2. Here are some key points about overfitting in decision trees:

- Training Data Memorization: Decision trees are prone to overfitting when they try to memorize the training data instead of capturing general patterns and relationships. The tree may create branches and splits that perfectly match the training instances, including noise and outliers, leading to a high level of specificity that does not generalize well to new data.
- Excessive Complexity: Overfitting often arises when decision trees grow too deep and include many levels of splits. This complexity allows the tree to capture even the smallest variations or peculiarities in the training data, which are unlikely to occur in future data. As a result, the decision tree becomes too specific and fails to generalize to unseen examples.
- Lack of Pruning: Pruning is a technique used to counter overfitting in decision trees. It involves removing unnecessary branches and nodes from the tree to simplify its structure and increase generalization. When pruning is not applied or is insufficient, overfitting can occur as the tree grows without constraint.
- Impact on Performance: An overfit decision tree may exhibit excellent performance on the training data, achieving high accuracy or low error rates. However, when tested on new, unseen data, its performance can significantly degrade. This discrepancy between training and test performance is an indication of overfitting.
- Cross-Validation and Regularization: Cross-validation is a valuable technique for assessing the performance and identifying potential overfitting in decision trees. Regularization methods, such as adjusting the complexity parameter (e.g., maximum

depth) or using regularization terms in the tree building process (e.g., cost complexity), can also help combat overfitting.

3.Overfitting in decision trees is a common challenge when building models. It is essential to strike a balance between capturing meaningful patterns and avoiding excessive complexity. Through careful model selection, feature engineering, regularization, and pruning techniques, it is possible to mitigate overfitting and create decision trees that generalize well to new data.

Q3(b) What is entropy? How we calculate it? (5)

Ans. 1. In the context of Artificial Intelligence (AI) and machine learning, entropy is a measure of impurity or uncertainty in a dataset. It is commonly used in decision tree algorithms, such as ID3 or C4.5, to assess the quality of a split and make informed decisions during the learning process.

3. In the context of AI, entropy is calculated as follows:
   Calculate the probability of each class label in the dataset.
   Compute the entropy for each class label using the formula:

   $$\text{Entropy} = -\Sigma P(c) \log_2(P(c))$$

   Where:

   P(c) represents the probability of a specific class label c.

   Σ denotes the sum over all possible class labels.

4. The resulting entropy value provides a measure of the disorder or uncertainty within the dataset. A lower entropy indicates a more pure or homogeneous distribution of class labels, while a higher entropy suggests greater diversity or randomness.
5. The calculation of entropy helps in decision tree algorithms to determine the most informative feature or attribute for splitting the data. The goal is to minimize entropy by selecting the feature that maximally reduces uncertainty and improves the separation of different classes.
6. By comparing the entropy before and after a split, the information gain can be calculated, which quantifies the reduction in entropy achieved by the split. High information gain implies a more significant reduction in uncertainty and a more effective split for decision making.

7. Entropy, along with information gain, forms the basis for feature selection and decision-making processes in various machine learning algorithms. It enables the identification of the most valuable features that contribute to the predictive power and accuracy of AI models.

Q3(c) Explain Artificial Neural Network. (5)

Ans. 1. An artificial neural network (ANN) is a computational model inspired by the structure and functionality of the human brain. It is a powerful machine learning technique used for solving complex problems, such as pattern recognition, regression, classification, and optimization.

2. Here's an explanation of the key components and workings of an artificial neural network:

- Neurons (Nodes):
  An artificial neural network consists of interconnected computational units called neurons or nodes. These nodes simulate the behavior of biological neurons and perform computations on input data.
- Layers:
  Neurons are organized into layers within an ANN. The three main types of layers are:
  a. Input Layer: This layer receives and represents the input data to the network. Each node in the input layer corresponds to a specific feature or attribute of the input data.
  b. Hidden Layers: These intermediate layers perform computations and transformations on the input data. They extract relevant features and patterns from the data, allowing the network to learn complex relationships.
  c. Output Layer: The final layer produces the output or prediction of the network. The number of nodes in the output layer depends on the type of problem being solved (e.g., binary classification, multi-class classification, regression).
- Feedforward and Backpropagation:
  In the feedforward phase, data flows through the network from the input layer to the output layer, with computations and activations occurring at each neuron. The output is then compared to the desired output, and an error or loss value is computed.
- Training and Learning:

ANNs are trained using labeled training data. The network iteratively adjusts its weights and biases to minimize the error between its predictions and the true labels. This learning process is known as supervised learning, where the network learns from examples and gradually improves its performance.

3. Artificial neural networks are widely used in various domains, including image recognition, natural language processing, speech recognition, recommendation systems, and many more. Their ability to model complex non-linear relationships and learn from data makes them a powerful tool in the field of artificial intelligence and machine learning.

Q3(d) What is Logistic Regression explain with the help of an example. (5)

Ans. 1. Logistic regression is a statistical learning algorithm used for binary classification problems, where the goal is to predict the probability of an input belonging to one of two classes. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm.

2. Here's an explanation of logistic regression using an example:

Example: Predicting Loan Approval

Suppose you work for a bank and want to predict whether a loan applicant will be approved or denied based on certain factors such as income, credit score, and debt-to-income ratio. You have a dataset containing historical loan applications along with their approval outcomes.

- Data Preparation:
  Each loan application in the dataset is represented by a set of features, such as income, credit score, and debt-to-income ratio. Let's assume we have two features: income (numeric) and credit score (numeric).The target variable is the loan approval outcome, which can take two values: approved (1) or denied (0).
- Hypothesis and Model:
  In logistic regression, we assume a linear relationship between the features and the log-odds of the loan approval. The log-odds are transformed using the logistic function (sigmoid function) to obtain the predicted probability. The logistic regression model can be represented as:

  Probability(Approval) = sigmoid(W_0 + W_1 * income + W_2 * credit_score)

  where W_0, W_1, and W_2 are the parameters (weights) to be learned from the data.

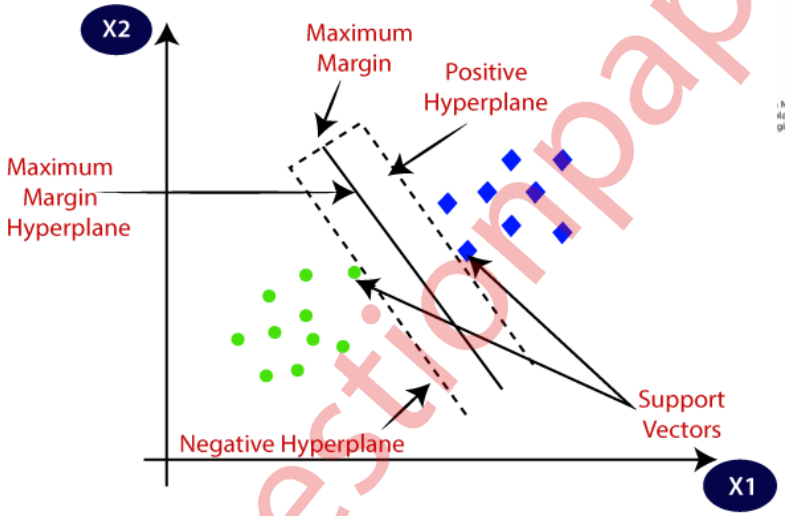
- Learning and Training:

  The goal is to learn the optimal values of the parameters that best fit the data. This is achieved through an optimization algorithm, such as maximum likelihood estimation or gradient descent. The algorithm iteratively adjusts the weights to minimize the difference between the predicted probabilities and the actual loan approval outcomes in the training data.
- Decision Boundary and Prediction:

  Once the model is trained, it can be used to predict the loan approval outcome for new loan applications.

3. Logistic regression is a popular and widely used algorithm in binary classification problems due to its simplicity, interpretability, and ability to handle both linear and non-linear relationships.

Q3(e) Explain Support Vector Machine and its properties.



Ans.

1. Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. It finds an optimal hyperplane or decision boundary that maximally separates different classes while maximizing the margin between them.

2. Here's an explanation of SVM and its properties:

- Linear Separability:

  SVM works on the principle of finding a hyperplane that can linearly separate the input data into different classes. Linear separability means that the classes can be separated by a straight line or hyperplane in the feature space.
- Margin Maximization:

  SVM aims to find the hyperplane that maximizes the margin between the classes.

The margin is the distance between the hyperplane and the closest data points from each class. By maximizing the margin, SVM seeks to achieve better generalization and robustness to new data.

- Support Vectors:
  Support vectors are the data points that lie closest to the decision boundary or hyperplane. These points have the most influence on determining the hyperplane's position and are used to define the decision boundary. SVM focuses only on these support vectors, making it memory-efficient and suitable for handling large datasets.
- Kernel Functions:
  SVM can handle non-linearly separable data by using kernel functions. Kernel functions transform the input data into a higher-dimensional feature space, where it becomes linearly separable. This transformation allows SVM to find a hyperplane in the transformed feature space, which corresponds to a non-linear decision boundary in the original space.
- Interpretability:
  SVM provides interpretability in terms of support vectors and their associated weights. The support vectors represent the most critical instances for determining the decision boundary. Additionally, the weight assigned to each feature provides insights into the feature's importance in the classification process.

3. Support Vector Machines are widely used in various domains, including image classification, text classification, bioinformatics, and finance. They offer good generalization performance, especially when the data is well-separated or when kernel functions are used to handle non-linear relationships

Q3(f) What is ensemble learning? How bagging algorithm works. (5)

Ans. 1. Ensemble learning is a machine learning technique that combines multiple individual models (learners) to make more accurate and robust predictions.

2. It leverages the wisdom of the crowd by aggregating the predictions of multiple models to achieve better overall performance.

3. Bagging (Bootstrap Aggregating) is one popular ensemble learning algorithm that works by creating multiple subsets of the original training data through random sampling with replacement. Each subset is used to train a separate base learner, typically using the same learning algorithm. The individual models are then combined or aggregated to make

predictions.

4. Here's an explanation of how the bagging algorithm works:

- Data Sampling:
  Bagging starts by creating multiple subsets of the original training data through random sampling with replacement. Each subset, known as a bootstrap sample, has the same size as the original dataset.
- Base Learners:
  A base learning algorithm, such as decision trees, is applied to each bootstrap sample independently. The same learning algorithm is typically used to ensure consistency and comparability among the base learners.
- Prediction Aggregation:
  Bagging combines the predictions of the individual models to make final predictions.
- Robustness and Generalization:
  Bagging aims to improve the overall performance of the ensemble by reducing variance and increasing stability.

5. The bagging algorithm, through its aggregation of multiple models, reduces overfitting, improves generalization, and increases prediction accuracy.

6. It is widely used in various machine learning tasks, including classification, regression, and anomaly detection. Examples of bagging-based algorithms include Random Forests, which combine bagging with decision trees, and AdaBoost, which adapts the weighting of training instances.

Q4. Attempt any three of the following **[15]**

Q4(a) Explain maximum likelihood parameter learning for continuous model. **(5)**

Ans. 1. Maximum Likelihood Estimation (MLE) is a widely used method for parameter learning in continuous models. It involves finding the parameter values that maximize the likelihood of observing the given data.

2. Explanation of Maximum Likelihood parameter learning for continuous models:

- Model Assumption: Continuous models assume that the data follows a specific probability distribution, such as the Gaussian distribution.
- Likelihood Function: The likelihood function measures the probability of observing

the given data under the assumed distribution. For continuous models, the likelihood function is the product of the probability density function (PDF) evaluated at each data point.

- Log-Likelihood Function: To simplify calculations and avoid numerical instability, it is common to work with the log-likelihood function, which is the logarithm of the likelihood function.
- Parameter Estimation: The goal is to find the parameter values that maximize the log-likelihood function or minimize the negative log-likelihood (NLL) function. Optimization algorithms, such as gradient descent or numerical optimization methods, are used to iteratively update the parameter values.
- Convergence: The optimization process continues until a convergence criterion is met, typically based on the change in log-likelihood or the gradient magnitude falling below a threshold.

3. Maximum Likelihood parameter learning provides optimal parameter values that maximize the likelihood of the observed data under the assumed distribution.

4. It is a fundamental technique for estimating parameters in continuous models, enabling better understanding and analysis of the underlying data distribution.

Q4(b) What is reinforcement learning? Explain in detail. (5)

Ans. 1. Reinforcement Learning (RL) is a branch of machine learning focused on enabling agents to learn and make decisions in an environment through interactions and feedback.

2. It involves an agent that learns to navigate and optimize its behavior by receiving rewards or penalties based on its actions.

3. explanation of reinforcement learning:

- Agent and Environment: In RL, an agent interacts with an environment, which can be a simulated or real-world setting. The agent takes actions, and the environment responds with new states and rewards.
- Learning through Feedback: The agent learns through a trial-and-error process. It doesn't receive explicit instructions but instead learns from the consequences of its actions.

  Feedback is provided in the form of rewards or penalties, which indicate the

desirability or undesirability of the agent's actions.

- Exploration and Exploitation: RL involves a trade-off between exploration and exploitation. Initially, the agent explores different actions and their outcomes to learn about the environment. Over time, it starts exploiting its knowledge to make optimal decisions.
- Markov Decision Process (MDP): RL problems are often formulated as Markov Decision Processes, which formalize the sequential decision-making process. MDPs consist of states, actions, transition probabilities, rewards, and a discount factor to balance immediate and future rewards.

4. Reinforcement Learning has applications in various domains, including robotics, game playing, recommendation systems, and autonomous vehicles. It enables agents to learn from experience, adapt to changing environments, and optimize their decision-making processes based on rewards and feedback.

Q4(c) What are beta distributions explain with the example. (5)

Ans. 1. The Beta distribution is a continuous probability distribution used in AI and machine learning to model and analyze random variables that have values within a specific range. It is commonly used when dealing with quantities that are constrained between 0 and 1, such as probabilities or proportions.

2. Example: Conversion Rate Optimization

Suppose you have an e-commerce website and want to optimize the conversion rate (the proportion of visitors who make a purchase).

You have collected data on 1000 visitors, where 400 made a purchase and 600 did not.

To model the conversion rate, you can use a Beta distribution with parameters $\alpha = 401$ (400 + 1) and $\beta = 601$ (600 + 1). With this Beta distribution, you can compute statistics such as the mean, variance, and percentiles to understand the distribution of conversion rates.

3. Bayesian Inference and Updating:

One of the strengths of the Beta distribution is its conjugacy in Bayesian inference. This means that if the prior distribution for a parameter is Beta, and the likelihood function is binomial, the posterior distribution will also be a Beta distribution. This property allows for

efficient updating of beliefs and incorporating new evidence as more data becomes available.

Applications:

- The Beta distribution finds applications in various AI tasks, including A/B testing, click-through rate modeling, sentiment analysis, and Bayesian modeling.
- It is particularly useful when dealing with proportions or probabilities constrained within a range, providing a flexible and interpretable distribution for modeling uncertainty.
- The Beta distribution is a valuable tool in AI and machine learning, providing a flexible framework for modeling random variables constrained within a specific range. Its ability to capture uncertainty and update beliefs in a Bayesian setting makes it suitable for a wide range of applications involving proportions and probabilities.

Q4(d)What is EM Algorithm? Explain its steps. (5)

Ans. 1. The Expectation-Maximization (EM) algorithm is an iterative optimization algorithm used to estimate the parameters of statistical models when there are missing or hidden variables in the data. It is widely employed in various fields, including machine learning, data analysis, and computer vision.

2. Here's an explanation of the EM algorithm and its steps:

Step 1: Initialization: Start by initializing the model's parameters. This could involve setting initial values randomly or based on prior knowledge.

Step 2: E-step (Expectation): In the E-step, the algorithm estimates the expected value of the hidden variables given the observed data and the current parameter estimates. It computes the posterior distribution of the hidden variables, which represents the probability distribution over the possible values of the hidden variables.

Step 3: M-step (Maximization): In the M-step, the algorithm updates the parameter estimates to maximize the likelihood of the observed data given the expected values of the hidden variables obtained from the E-step.This step involves finding the parameter values that maximize the expected log-likelihood, also known as the Q-function.

Step 4: Iteration: The E-step and M-step are repeated iteratively until convergence. In each iteration, the E-step computes the expected values of the hidden variables, and the M-step

updates the parameter estimates based on these expected values.

Step 5: Convergence: The algorithm continues iterating until a convergence criterion is met. This criterion is typically based on the change in the log-likelihood or the parameter values falling within a specified threshold.

The EM algorithm is widely used in various applications, including clustering, latent variable models, mixture models, and Gaussian mixture models. It provides a powerful and flexible framework for dealing with complex models involving missing data or unobserved variables.

Q4(e) Explain Naïve Bayes algorithm. (5)

Ans. 1. Naive Bayes is a simple yet effective probabilistic machine learning algorithm used for classification tasks. It is based on the principle of Bayes' theorem and assumes independence among the features, hence the term "naive."

2. Here's an explanation of the Naive Bayes algorithm:

Bayes' Theorem: Naive Bayes is based on Bayes' theorem, which calculates the posterior probability of a class given the observed features.

Bayes' theorem states that P(C|X) = (P(X|C) * P(C)) / P(X), where C represents the class label and X represents the set of observed features.

3.Independence Assumption: The Naive Bayes algorithm assumes that the features are conditionally independent of each other given the class label. This assumption simplifies the calculations and allows for efficient inference.

4. Parameter Estimation: Naive Bayes estimates the parameters required for classification. For categorical features, it estimates the class prior probabilities P(C) and the conditional probabilities P(X|C) for each feature given each class.

5. Classification: To classify a new instance with Naive Bayes, the algorithm calculates the posterior probabilities for each class given the observed features using Bayes' theorem.

6. Naive Bayes is known for its simplicity, speed, and scalability. It works well on large datasets and can handle high-dimensional feature spaces.

7.Although it assumes feature independence, Naive Bayes often performs surprisingly well in practice, especially for text classification, spam filtering, sentiment analysis, and other similar tasks.

8.It is a popular choice for baseline models and is widely used in various applications due to its efficiency and reasonable performance.

Q4(f) Write a short note on Hidden Markov Model. (5)

Ans. 1. A Hidden Markov Model (HMM) is a statistical model used to model sequential data with hidden states. It is widely employed in various fields, including speech recognition, natural language processing, bioinformatics, finance, and more.

2. Key components of a Hidden Markov Model:

- States: HMM consists of a set of hidden states, which represent the underlying, unobservable variables of the system. These states form a Markov chain, implying that the probability of transitioning from one state to another only depends on the current state and not on the history.
- Observations: Each hidden state generates an observable output (observation) according to a specific probability distribution. However, the exact state generating an observation remains unknown (hence "hidden").
- State Transition Probabilities: HMM incorporates state transition probabilities, indicating the likelihood of moving from one state to another. These probabilities are usually represented in a transition matrix.
- Emission Probabilities: Each state is associated with an emission probability distribution, which models the likelihood of generating a particular observation given the state. These probabilities are often presented in an emission matrix.

The three fundamental problems associated with Hidden Markov Models:

- Evaluation
- Decoding
- Learning

Applications of Hidden Markov Models:

Speech Recognition: HMMs are used to model phonemes and words in speech recognition systems, aiding in speech-to-text conversion.

Natural Language Processing: HMMs are employed for part-of-speech tagging, named entity recognition, and other sequential text processing tasks.

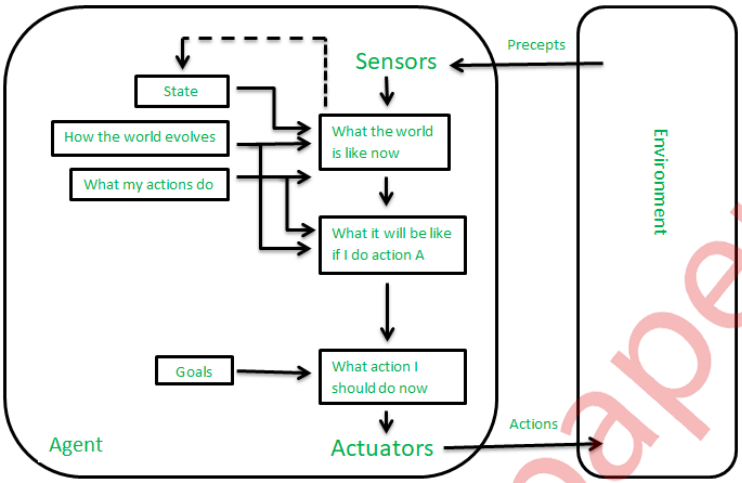
Bioinformatics: HMMs are applied to analyze DNA and protein sequences, identifying genes, regulatory elements, and functional domains.

Finance: HMMs are used for predicting market trends and volatility.

Q5. Attempt any Five of the following **[15]**

Q5(a) Explain the concept of goal-based agent. **(5)**

Ans:



1. A goal-based agent is a type of intelligent agent in artificial intelligence that operates by pursuing specific goals or objectives. The main idea behind a goal-based agent is to achieve its objectives effectively and efficiently while interacting with its environment. These agents are designed to take actions that will bring them closer to their desired goals and avoid actions that are counterproductive.

2. Key components of a goal-based agent:

- Goals: The agent's goals define the desired states or outcomes that it aims to achieve. These goals can be simple or complex, short-term or long-term, and may be predefined by the designer or learned from experience.
- Perception: The agent perceives its environment through sensors, which provide information about the current state of the environment and the agent's internal state.
- Decision-making: The agent uses its internal knowledge, reasoning capabilities, and possibly learning mechanisms to make decisions about which actions to take. The goal-based agent evaluates potential actions based on their expected outcomes and their alignment with achieving its goals.

- Execution and Feedback Loop: The agent interacts with the environment, performs actions, and then receives feedback or observations about the consequences of its actions. This feedback is used to update the agent's internal state and knowledge, which can affect future decisions and actions.

Examples of goal-based agents:

- Chess-playing Agent: In a chess-playing agent, the goal is to win the game. The agent observes the current state of the chessboard, analyzes potential moves, and selects the move that maximizes the likelihood of winning.

Q5(b) How to measure problem solving performance? (5)

Ans. 1. Measuring problem-solving performance involves assessing how well an individual or an AI system can identify, analyze, and solve problems in a given context.

2. The evaluation metrics and methods can vary based on the nature of the problem, the available resources, and the specific goals of the assessment. Here are some common approaches to measure problem-solving performance:

- Accuracy: Accuracy is a straightforward metric used to evaluate problem-solving performance. It measures the percentage of correctly solved problems out of the total number of problems attempted. This metric is suitable when there is a clear and unambiguous definition of a correct solution.
- Time to Solve: The time taken to solve a problem can be an essential performance metric, especially in time-critical scenarios. It assesses how quickly an individual or an AI system can arrive at a solution. However, it is crucial to strike a balance between speed and accuracy, as a rushed solution might be less accurate.
- Solution Quality: Some problems may have multiple acceptable solutions or varying levels of correctness. In such cases, solution quality can be used as a metric. For example, in optimization problems, the closeness of the solution to the optimal one can be measured.
- Robustness: Evaluating the problem-solving performance under different conditions or scenarios can help assess the robustness of the approach. A solution that works well in various situations is considered more robust than one that is sensitive to changes.
- Exploration vs. Exploitation Trade-off: In problems with uncertainty or where

exploration is necessary, evaluating how well an agent balances exploration (trying new approaches) and exploitation (using known strategies) can be important.

- Learning Curves: For AI systems, learning curves can be used to measure problem-solving performance over time. This involves plotting performance metrics against the number of training examples or problem-solving attempts to understand the system's learning progress.
- Human Evaluation: In some cases, especially in creative or subjective problem-solving tasks, human evaluation may be necessary. Experts or independent judges can assess the quality of solutions generated by individuals or AI systems.

Q5(c) What is Linear Regression? State its types. (5)

Ans. 1. Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data.

2.The goal of linear regression is to find the best-fitting line (or hyperplane in higher dimensions) that minimizes the difference between the predicted values and the actual values of the dependent variable.

3.It is one of the simplest and most widely used techniques in statistics and machine learning for predicting continuous numerical outcomes.

4.There are different types of linear regression based on the number of independent variables and the specific characteristics of the data. The two main types are:

- Simple Linear Regression: Simple linear regression involves one independent variable (x) and one dependent variable (y). It models the relationship between the two variables as a straight line.
- Multiple Linear Regression: It models the relationship between the dependent variable and multiple independent variables as a hyperplane in a multi-dimensional space.

Other variations and extensions of linear regression include:

- Polynomial Regression: Polynomial regression is a type of multiple linear regression, but the independent variables are transformed by raising them to different powers. This allows the model to fit more complex curves.
- Ridge Regression (L2 Regularization): Ridge regression adds a penalty term to the

linear regression cost function to prevent overfitting and handle multicollinearity issues in multiple linear regression.

Q5(d) How supervised learning algorithm works explain with example? (5)

Ans. 1. Supervised learning is a type of machine learning where the algorithm learns from a labeled dataset, meaning it is provided with input-output pairs during the training process. The goal is for the algorithm to learn the mapping between the input and the corresponding output so that it can make accurate predictions on new, unseen data.

2. supervised learning algorithm works:

Example: House Price Prediction

- Data Collection: First, you need to gather a dataset with labeled examples. It would contain rows of data, where each row represents a house, and the columns represent the features (size, bedrooms, location) and the target variable (price). Each row would look like this:
- Data Preprocessing: Before training the model, you need to preprocess the data. This includes tasks like handling missing values, converting categorical variables (like "Location") into numerical representations (e.g., one-hot encoding), and scaling the features to bring them to a similar range.
- Training the Model: Now, you can use the labeled data to train a supervised learning algorithm, such as a linear regression model. The algorithm will learn the relationship between the input features (size, bedrooms, location) and the target variable (price) by adjusting its internal parameters during the training process.
- Model Evaluation: After training, you need to evaluate the model's performance to see how well it can predict house prices on new, unseen data. You can split the dataset into a training set and a test set. The training set is used to train the model, and the test set is used to evaluate its performance. Common evaluation metrics for regression tasks include Mean Squared Error (MSE) and R-squared (coefficient of determination).
- Making Predictions: Once the model is trained and evaluated, you can use it to make predictions on new houses. For instance, if you have the features of a house that you want to sell, such as its size, number of bedrooms, and location, you can input these

values into the trained model, and it will predict the corresponding price for that house.

Q5(e)What is Statistical Learning? (5)

Ans. 1. Statistical learning is a field of study that focuses on developing and applying methods to analyze and model data to make predictions and gain insights.

2.It involves using statistical techniques and mathematical models to learn patterns and relationships from data.

Here are five key points that summarize statistical learning:

- Data-Driven Approach: Statistical learning is a data-driven approach that relies on analyzing and processing data to extract meaningful patterns and relationships. It uses historical data to build models that can be used to make predictions on new, unseen data.
- Supervised and Unsupervised Learning: Statistical learning can be categorized into supervised learning and unsupervised learning. In supervised learning, the model is trained on labeled data with input features and corresponding output labels. In unsupervised learning, the model identifies patterns and structures within the data without explicit labels.
- Model Building and Evaluation: The core of statistical learning involves building mathematical models, such as linear regression, decision trees, support vector machines, neural networks, and more. These models are trained on training data and evaluated on test data to assess their predictive accuracy and generalization performance.

3. Applications: Statistical learning has wide-ranging applications in various fields, including machine learning, data science, artificial intelligence, computer vision, natural language processing, finance, healthcare, marketing, and more. It is used to solve complex problems, make data-driven decisions, and improve business processes.

Q5(f)Explain any one application of Reinforcement learning? (5)

Ans. 1. Reinforcement learning is a type of machine learning paradigm where an agent learns to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties based on its actions.

2. The goal of the agent is to maximize the cumulative reward over time, leading to the discovery of optimal strategies or policies.

3. One application of reinforcement learning is in robotics. In robotic control, reinforcement learning can be used to teach robots how to perform tasks autonomously by learning from their interactions with the environment.

4.For example, a robot can learn to walk, pick up objects, or navigate through a cluttered environment using reinforcement learning.

5.The robot receives rewards or penalties based on the success or failure of its actions, allowing it to improve its movements and behaviors over time, ultimately becoming more efficient and adaptive in completing tasks.

Q5(g)Explain Heuristic function ? (5)

Ans. 1. A heuristic function is a problem-solving technique used in various algorithms and search strategies, particularly in the context of artificial intelligence and optimization.

2.It is a function that provides an estimate or approximation of the cost or value of reaching a goal state from a given state in a search space.

3.Estimation of Cost: The heuristic function evaluates the potential cost or distance from the current state to the goal state. It provides an informed estimate, which means it guides the search algorithm towards promising paths that are likely to lead to the goal.

4.Admissibility: A heuristic function is considered admissible if it never overestimates the true cost to reach the goal state from the current state. In other words, the heuristic function is optimistic and is guaranteed to be equal to or less than the actual cost.

5.Influence on Search Algorithms: Heuristic functions play a crucial role in guiding search algorithms like A* (A star) and Best-First Search. By using the heuristic function, these algorithms prioritize exploring states that are expected to lead to the goal, making the search more efficient and effective.

6. The effectiveness of a heuristic function lies in its ability to strike a balance between

providing a good estimate of the cost while avoiding excessive computational complexity. A well-designed heuristic function can significantly speed up search algorithms and help find solutions more quickly in various AI and optimization problems.

Q5(h)Write a short note on Non parametric model ? (5)

Ans. 1. A non-parametric model is a type of statistical model that does not make explicit assumptions about the underlying distribution of the data. Unlike parametric models, which have a fixed number of parameters, non-parametric models can adapt to the complexity of the data and often require more data to learn effectively.

2.Key characteristics of non-parametric models:

- Flexibility: Non-parametric models are highly flexible and can represent complex relationships between variables without imposing specific constraints on the data distribution. They can handle both linear and non-linear relationships, making them suitable for a wide range of data types and patterns.
- Adaptability: Non-parametric models can adjust to the size of the dataset, becoming more accurate as more data becomes available. They do not assume fixed parameter values, making them well-suited for situations where the underlying data distribution may be unknown or change over time.
- Example: One of the most popular non-parametric models is the k-nearest neighbors (KNN) algorithm. In KNN, predictions are made based on the majority class of the k-nearest data points to the target instance. It does not make assumptions about the data distribution and can handle complex decision boundaries in classification tasks.
- Non-parametric models are commonly used in various machine learning tasks, such as classification, regression, density estimation, and clustering. However, they may require more computational resources and larger datasets compared to parametric models due to their increased flexibility and adaptabilit