AI (MST 2) SYLLABUS.

1. Probabilistic Reasoning Probability:- Probabilistic reasoning is a type of reasoning that involves uncertainty or incomplete information. It is used to make decisions or predictions based on the likelihood of different outcomes. Probability theory provides a mathematical framework for probabilistic reasoning.

There are two main types of probability:

1. Classical probability - This type of probability is based on equally likely outcomes .For example, flipping a coin has two equally likely outcomes: heads or tails. The probability of getting heads is 1/2, and the probability of getting tails is also 1/2.

2. Bayesian probability - This type of probability is based on prior knowledge or experience. It is updated as new information becomes available. For example, if you are trying to predict the weather, you might start with a prior probability based on historical data, but then update that probability based on current weather patterns.

Examples of classical probability include rolling a die (each number has an equal chance of being rolled) and drawing a card from a deck (each card has an equal chance of being drawn). An example of Bayesian probability is predicting the outcome of an election based on polling data and other factors.

conditional probability and its type with example:- Conditional probability is a type of probability that measures the likelihood of an event occurring, given that another event has already occurred. It is expressed as the probability of event A occurring given that event B has occurred, and is denoted by P(A|B).

There are two types of conditional probability: dependent and independent.

Dependent probability occurs when the occurrence of one event affects the probability of another event. For example, suppose a bag contains 5 red balls and 3 blue balls. If we randomly select one ball from the bag without replacement, the probability of selecting a red ball on the first draw is 5/8. However, if we then select a second ball without replacement, the probability of selecting another red ball is now 4/7, since there are only 4 red balls left in the bag. The probability of selecting are ball on the second draw is dependent on whether or not a red ball was selected on the first draw.

Independent probability occurs when the occurrence of one event does not affect the probability of another event. For example, suppose we flip a fair coin twice. The probability of getting heads on the first flip is 1/2, and the probability of getting heads on the second flip is also 1/2. The outcome of the first flip does not affect the outcome of the second flip, so these events are independent.

2. Bayes Rule and it’s type with example:- Bayes' rule is a fundamental theorem in probability theory that describes the relationship between conditional probabilities. It is named after the Reverend Thomas Bayes, an 18th-century British statistician and theologian, who first formulated the rule.

   Bayes' rule states that the probability of an event A given that event B has occurred is equal to the probability of event B given that event A has occurred, multiplied by the prior probability of event A, divided by the prior probability of event B. This can be written mathematically as:

   P(A|B) = P(B|A) * P(A) / P(B)

   where P(A|B) is the conditional probability of A given B, P(B|A) is the conditional probability of B given A, P(A) is the prior probability of A, and P(B) is the prior probability of B.

   There are several types of Bayes' rule, including:

   1. Naive Bayes: This is a classification algorithm that uses Bayes' rule to predict the probability of a particular class given a set of input features. Naive Bayes assumes that all input features are independent of each other, which makes it computationally efficient but may result in lower accuracy in some cases.

   2. Bayesian inference: This is a general method for updating probabilities based on new evidence or data. Bayesian inference starts with a prior probability distribution over some unknown quantity (such as the success rate of a medical treatment), and then updates this distribution based on observed data using Bayes' rule.

   Bayesian networks: These are graphical models that represent probabilistic relationships between variables using directed acyclic graphs (DAGs). Bayesian networks allow us to reason about complex systems with many interdependent variables and uncertain outcomes.

3. Bayesian Networks- representation construction and inference its type and example:
   Bayesian Networks are probabilistic graphical models that are used to represent and reason about uncertain knowledge. They consist of a directed acyclic graph (DAG) where nodes represent random variables and edges represent probabilistic em. Each node in the graph is associated with a probability distribution that describes the uncertainty about its value given the values of its parents in the graph.

   Bayesian Networks can be constructed using expert knowledge or learned from data using machine learning techniques such as Maximum Likelihood Estimation (MLE) or Bayesian Learning. The construction process involves identifying the relevant variables, their dependencies, and specifying the probability distributions that describe their relationships. Once the network is constructed, it can be used for various inference tasks such as prediction, diagnosis, and decision-making.

There are several types of inference algorithms that can be used with Bayesian Networks, including exact inference algorithms and approximate inference algorithms. Exact inference algorithms, such as variable elimination and junction tree algorithms, compute the exact posterior probabilities of the variables in the network. Approximate inference algorithms, such as Markov Chain Monte Carlo (MCMC) methods and Variational Inference (VI), provide approximate solutions to inference problems in large and complex networks.

An example of a Bayesian Network is a medical diagnosis system where the nodes represent symptoms, diseases, and test results. The edges represent causal relationships between these variables, and the probability distributions represent the likelihood of each variable given its parents in the graph. The system can be used to predict the probability of a patient having a particular disease given their symptoms and test results.

4. hidden Markov model and its type :- A hidden Markov model (HMM) is a statistical model used to describe a system where the system being modeled is assumed to be a Markov process with unobserved (hidden) states. HMMs are widely used in various fields, including speech recognition, bioinformatics, and natural language processing.

   The basic idea behind an HMM is that there is an underlying process with a finite number of states, and each state has a probability distribution over possible observations. The goal is to infer the sequence of hidden states based on the observed sequence of outputs. This is done using the forward-backward algorithm or the Viterbi algorithm.

   There are several types of HMMs, including:

   1. Discrete-state HMMs: In this type of HMM, the state space is discrete, meaning that each state corresponds to a categorical variable. For example, in speech recognition, the states might correspond to different phonemes.

   2. Continuous-state HMMs: In this type of HMM, the state space is continuous, meaning that each state corresponds to a continuous variable. For example, in speech recognition, the states might correspond to different acoustic features.

   3. Semi-Markov HMMs: In this type of HMM, the duration of each state is modeled explicitly as a random variable. This allows for more flexible modeling of systems where the duration of each state can vary.

5. Markov Decision process MDP formulation and its type:- Markov Decision Process (MDP) is a mathematical framework used to model decision-making problems where

outcomes are partially random and partially under the control of a decision-maker. MDPs are widely used in artificial intelligence, operations research, control engineering, and economics. The formulation of MDP involves five components:

1. State Space: A set of all possible states that the system can be in. The state can be fully observable or partially observable.

2. Action Space: A set of all possible actions that the decision-maker can take in each state.

3. function that defines the probability of moving from one state to another after taking a particular action.

4. Reward Function: A function that assigns a numerical reward to each state-action pair.

5. Discount Factor: A scalar value that determines the importance of future rewards relative to immediate rewards.

MDP can be classified into several types based on their properties:

1. Finite vs. Infinite Horizon: In finite horizon MDP, the decision-making process has a fixed number of steps, while in infinite horizon MDP, there is no fixed limit on the number of steps.

2. continuous state space MDP, the state space is uncountable, while in discrete state space MDP, the state space is countable.

3. Continuous vs. Discrete Action Space: In continuous action space MDP, the action space is uncountable, while in discrete action space MDP, the action space is countable.

4. Markovian vs. Non-Markovian: In Markovian MDP, the transition probability function depends only on the current state and action and not on any previous states or actions, while in non-Markovian MDP, the transition probability function depends on previous states and actions.

6. utility theory and its types:- Utility theory is a branch of economics that studies the preferences of individuals and how they make decisions based on those preferences. It assumes that individuals have a set of preferences over different outcomes and that they choose the option that provides them with the highest level of satisfaction or utility. The concept of utility is subjective and varies from person to person, which makes it difficult to measure.

   There are several types of utility theory, including:

1. Cardinal utility theory: This theory assumes that utility can be measured and assigned a numerical value. It suggests that individuals can compare the level of satisfaction they derive from different outcomes and choose the option that provides them with the highest level of utility.

2. Ordinal utility theory: This theory assumes that utility cannot be measured but can be ranked in order of preference. It suggests that individuals can compare the level of satisfaction they derive from different outcomes and choose the option that is ranked higher in their preference order.

3. Expected utility theory**:** This theory assumes that individuals make decisions based on their expectations about the future outcomes. It suggests that individuals evaluate the probability of different outcomes and their associated utilities and choose the option with the highest expected utility.

7. utility functions and its types:- A utility function is a mathematical representation of a person's preferences over different outcomes. It is used in decision theory, economics, and game theory to model how individuals make choices. The utility function assigns a numerical value to each possible outcome, representing the level of satisfaction or happiness that the individual would derive from that outcome. The higher the utility value, the more desirable the outcome is for the individual.

   There are different types of utility functions, including:

   1. Linear Utility Function: In this type of utility function, the increase in utility is directly proportional to the increase in the outcome. For example, if an individual receives $10 for completing a task and $20 for completing two tasks, then the increase in utility is linear.

   2. Quadratic Utility Function: In this type of utility function, the increase in utility is proportional to the square of the increase in the outcome. For example, if an individual receives $10 for completing a task and $40 for completing four tasks, then the increase inutility is quadratic.

   3. Exponential Utility Function: In this type of utility function, the increase in utility is proportional to the exponential of the increase in the outcome. For example, if an individual receives $10 for completing a task and $80 for completing eight tasks, then the increase in utility is exponential.

8. value iteration and its type with example:- Value iteration is a dynamic programming algorithm used to solve Markov decision processes (MDPs). MDPs are mathematical models used to analyze decision-making problems in situations where outcomes are partly random and partly under the control of a decision maker. Value iteration is an iterative algorithm that computes the optimal value function and the

optimal policy for an MDP.

The value function represents the expected long-term reward that can be obtained by following a particular policy in a given state. The optimal value function gives the maximumexpected long-term reward that can be achieved from any starting state by following the optimal policy. The optimal policy is the one that maximizes the expected long-term reward.

The value iteration algorithm starts with an initial estimate of the value function and iteratively improves it until it converges to the optimal value function. At each iteration, the algorithm updates the value function for each state by taking the maximum over all possible actions and their corresponding expected rewards, discounted by a factor called the discount rate. The discount rate represents how much less future rewards are worth compared to immediate rewards.

The algorithm continues iterating until the difference between consecutive estimates of the value function is below a certain threshold, indicating convergence Once the optimal value function is obtained, it is straightforward to compute the optimal policy by choosing, for each state, the action that maximizes the expected
function.

There are two types of value iteration algorithms: synchronous and asynchronous. Synchronous value iteration updates all states in parallel at each iteration, whereas asynchronous value iteration updates only one state at a time, either randomly or according to some fixed order.

9. policy iteration and partially observable MDPs and its types:- Policy Iteration is a type of algorithm that is used to solve Markov Decision Processes (MDPs). It is an iterative process that involves two steps: policy evaluation and policy improvement .In the policy evaluation step, the value function of the current policy is calculated. In the policy improvement step, a new policy is generated based on the value function of the current policy.

   Partially Observable Markov Decision Processes (POMDPs) are a type of MDP where the agent does not have access to the full state of the environment. Instead, it receives observations that are affected by the underlying state. The agent must inferther underlying state based on these observations and take actions accordingly.

   To solve POMDPs, a technique called Belief State Policy Iteration (BSPI) is used. BSP Iis an extension of Policy Iteration that takes into account the belief state of the agent. In BSPI, the value function and policy are defined over belief states instead of states.

   There are several types of POMDPs, including:

   1. Dec-POMDPs**:** Decentralized POMDPs are a type of POMDP where multiple agents

make decisions in a decentralized manner. Each agent has its own observation and action space, and must coordinate with other agents to achieve a common goal.

2. Factored POMDPs: Factored POMDPs represent the underlying state as a set of variables that can be observed or unobserved. The value function and policy are defined over these variables.

3. Partially Observable Stochastic Games (POSGs): POSGs are a generalization of POMDPs where there are multiple agents interacting with each other in a stochastic environment.

10. Reinforcement Learning Passive reinforcement learning its type with example:-

Reinforcement Learning is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or punishments, which helps it learn to make better decisions in the future. Thereare two types of reinforcement learning: Passive Reinforcement Learning and Active Reinforcement Learning.

Passive Reinforcement Learning is a type of reinforcement learning where the agent learns from a fixed dataset without interacting with the environment. In other words, the agent does not receive any feedback or rewards during the learning process. Instead, it learns from a pre-existing dataset that has already been labeled with rewards or punishments.

11. direct utility estimation and its type with example:- Direct Utility Estimation (DUE)is a method used in decision-making processes to estimate the value of a particular action or outcome. It is a technique that helps in determining the utility of an alternative, which is defined as the satisfaction or benefit derived from it. DUE involves the direct measurement of utility using either a rating scale or a ranking system.

One type of DUE is Conjoint Analysis, which is used to determine how people value different attributes or features of a product or service. In this method, respondents are presented with different combinations of attributes and asked to choose their preferred option. The results are then analyzed to determine the relative importance of each attribute and its contribution to overall utility.

Another type of DUE is the Willingness-to-Pay (WTP) approach, which estimates the maximum amount that an individual is willing to pay for a particular good or service .This method is often used in pricing studies, where researchers want to determine the optimal price point for a product.

12. dynamic programming and its type with example:- Dynamic programming is a technique used in computer science, mathematics, and economics to solve complex problems by breaking them down into smaller subproblems. It is an optimization method that seeks to find the optimal solution to a problem by solving smaller subproblems and combining their solutions.

    There are two types of dynamic programming: **top-down** and **bottom-up**. Top-down dynamic programming starts with the original problem and breaks it down into smaller subproblems until it reaches the base case. It then solves each subproblem recursively and combines their solutions to get the final solution to the original problem. Bottom-up dynamic programming, on the other hand, starts with the base case and builds up to the original problem by solving each subproblem iteratively.

    To calculate the nth number in the Fibonacci sequence, we can use dynamic programming. We can start with the base cases where F(0) = 0 and F(1) = 1. Then we can use either top-down or bottom-up dynamic programming to calculate F(n) for any n greater than 1

13. Temporal difference learning:- Temporal difference (TD) learning is a type of machine learning algorithm used in reinforcement learning. It is a method for predicting the value function of a Markov decision process (MDP) based on observed sequences of state transitions. TD learning updates the value function incrementally, by bootstrapping from estimates of the value function at subsequent time steps.

    There are two main types of TD learning: on-policy and off-policy TD learning. On-policy TD learning updates the value function based on the actions taken by the current policy, while off-policy TD learning updates the value function based on the actions taken by a different policy.

    An example of on-policy TD learning is SARSA (state-action-reward-state-action), which is a type of TD control algorithm that learns an optimal policy by iteratively updating the Q-values (the expected cumulative reward for taking an action in a given state) using the observed state-action-reward-state-action sequences. SARSA uses an epsilon-greedy policy to balance exploration and exploitation during the learning process

14. Active reinforcement learning- Q learning:- Active reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with its environment. In active reinforcement learning, the agent takes actions in the environment and receives feedback in the form of rewards or punishments. The goal of the agent is to learn a policy that maximizes the cumulative reward over time.

    Q-learning is a popular algorithm used in active reinforcement learning. It is a model-free algorithm that does not require knowledge of the underlying dynamics of the environment. Q-learning works by estimating the optimal action-value function, which is the expected cumulative reward for taking a particular action in a particular state and following an optimal policy thereafter. The optimal policy is then derived from the optimal action-value function.