Assignment : Machine Learning, Artificial Intelligence, Cloud Computing

January 9, 2024

Q1. Definition of machine learning and examples of its applications in daily life.

Machine learning is a subfield of AI concerned with teaching computers to carry out tasks automatically via the use of pre-programmed algorithms and statistical models. Data patterns and conclusions are its foundation. Machine learning has many different uses in everyday life. They consist of:

1. Personalized recommendations are a feature that companies like Netflix or Amazon use to suggest movies or products to users based on their actions.
2. Speech recognition is the backbone of digital helpers such as Google Assistant and Siri.
3. Gmail's spam filters detect and remove spam emails by using machine learning.
4. When predicting traffic, Google Maps use machine learning to sift through data and recommend routes.
5. When it comes to healthcare, machine learning algorithms are a huge boon for illness diagnosis and personalized treatment plans.
6. Used by banks and other financial organizations for the purpose of detecting fraudulent activity.
7. Algorithms used in social media feeds, such as Facebook's News Feed, tailor material to individual users according on their activities and interests.

Machine learning has the potential to greatly improve and streamline many parts of our daily lives, as these examples show.

Q2. Differences between Machine Learning and Deep Learning.

Deep learning and machine learning vary in many important ways; this table compares and contrasts them. It also highlights the two methods' approaches, data needs, dependence on hardware, interpretability, feature engineering, application scope, training time, and output complexity. **Table 1 on page 2 refers to the differences between Machine Learning and Deep Learning.**

Aspect	Deep Learning	Machine Learning	Main Difference
Approach	Uses neural networks with multiple layers	Uses algorithms based on statistical methods	Depth and complexity of model architecture
Data Requirement	Requires large amounts of data	Can work with less data	Amount of data required for effective learning
Hardware Dependency	Highly dependent on high-end hardware	Less dependent on advanced hardware	Hardware requirements for processing
Interpretability	Often seen as a "black box" due to complexity	Generally more interpretable	Ease of understanding how decisions are made
Feature Engineering	Automated feature extraction	Requires manual feature selection	Involvement in feature extraction process
Application Scope	Ideal for high-complexity tasks like image and speech recognition	Used for simpler tasks like classification and regression	Complexity of tasks they are best suited for
Training Time	Longer training time due to complexity	Comparatively shorter training time	Duration required for model to learn
Output	Hierarchical, complex models	Simple to complex models, but usually less layered	Complexity of the output model

Table 1: Comparison between Deep Learning and Machine Learning

Q3. Basic objectives of the Canny edge detection method.

f A well-known methodology in the field of image processing, the Canny Edge Detection method has many primary goals, including the following:

1. **Noise Reduction:** Before edge detection, the image is smoothed using a Gaussian filter to reduce noise. This makes it less likely that false edges will be detected. For example, a noisy image might show false edges around specks of dust, but after applying a Gaussian filter, these are less likely to be detected as edges.

2. **Finding Intensity Gradients:** The method finds the edge strength by looking for the highest gradient in the image brightness. An example is detecting the outline of an object where there is a sharp change in intensity.

3. **Non-maximum Suppression:** This step thins out the edges. The algorithm goes through all the points on the gradient intensity and removes non-maximum pixels, enhancing the quality of the edges. In practice, this means that the edges of an object in an image will be sharp and fine, rather than thick and blurred.

4. **Double Thresholding:** The detected edges are then subjected to two thresholds: a high and a low. Strong edges are identified by the high threshold, while the low threshold helps to track weaker edges that are connected to strong edges. For instance, a faint line connected to a well-defined edge might be included as part of the edge.

5. **Edge Tracking by Hysteresis:** Finally, it involves suppressing all the other edges that are weak and not connected to strong edges, ensuring the detection of the most significant edges. This means that in a scenario like a photo with shadows, the edges of the actual objects are detected, while the edges formed by shadows are ignored.

Q4. Differences between supervised and unsupervised learning, with examples.

Table 2 on page 3 refers to the differences between supervised and unsupervised learning, with examples.

Supervised Learning	Unsupervised Learning	Main Difference
Involves training a model using labeled data, where the correct output is known. **Example:** Predicting house prices based on features like size, location, and age, where past data includes actual prices.	Involves training a model using unlabeled data, without specified outcomes. **Example:** Segmenting customers into groups based on purchasing behavior without predefined categories.	The presence or absence of labeled data for training. In Supervised Learning, the model learns from provided examples to predict outcomes. In Unsupervised Learning, the model identifies patterns or structures in the data without prior input.

Table 2: Differences between Supervised and Unsupervised Learning

Q5. Topics on input and hidden layer weights, activation functions in neural networks.

The most basic principles are those that define the design and operation of neural networks, as well as those that influence their capacity to learn and generate predictions.

1. **Input and Hidden Layer Weights:** In neural networks, weights are parameters that determine the strength of the connection between neurons in different layers. The input layer weights are the connections from the input to the first hidden layer, and hidden layer weights are connections between hidden layers or between a hidden layer and the output layer.
2. **Activation Functions:** Activation functions are mathematical equations that determine the output of a neural network node. They add non-linearity to the network, allowing it to learn complex patterns. Common activation functions include Sigmoid, Tanh, and ReLU (Rectified Linear Unit). Each activation function has its characteristics and is chosen based on the specific requirements of the neural network.

Q6.Application of clustering in the real world

Clustering, a fundamental technique in machine learning, finds extensive real-world applications. It involves grouping data points or objects in such a way that objects in the same group (or cluster) are more similar to each other than to those in other groups. Here are some applications:

1. **Market Segmentation:** Businesses use clustering to segment customers based on purchasing patterns, interests, or behaviors, enabling personalized marketing strategies.
2. **Social Network Analysis:** Clustering helps in identifying communities or groups within social networks, based on interactions or shared interests.
3. **Search Result Grouping:** Search engines use clustering to group similar results, enhancing user experience by categorizing information.
4. **Image Segmentation:** In image processing, clustering is used to partition an image into segments, aiding in tasks like object recognition or content-based image retrieval.
5. **Anomaly Detection:** Clustering helps in identifying unusual data points that differ significantly from the rest of the data, useful in fraud detection or monitoring system health.

Each application makes use of the capability of clustering to discover patterns or groups in data, which may either provide insights or improve decision-making processes.

Q7. Explanation of the ROC curve, false negatives, and true positives.

The ROC (Receiver Operating Characteristic) curve is a graphical representation used in machine learning to evaluate the performance of classification models. It plots the True Positive Rate (TPR)

against the False Positive Rate (FPR) at various threshold settings. True Positives (TP) are instances correctly identified as positive by the model, while False Negatives (FN) are actual positives that the model incorrectly classified as negative. The ROC curve helps in understanding the trade-off between sensitivity (TPR) and specificity (1-FPR), and an ideal model's ROC curve would reach towards the top left corner, indicating a higher TPR and a lower FPR.

Q8. Overview of the Naïve Bayesian Model.

In the field of machine learning, the Naïve Bayesian Model is a method that is both simple and powerful, and it is used for classification problems. An outline of its proceedings is as follows:

1. **Model Assumption:** The Naïve Bayesian model assumes that each feature it uses to make a prediction is independent of all other features. This simplification is why it's termed 'naïve'.
2. **Probability Calculation:** For each class, the algorithm calculates the probability of the data point belonging to that class. This is done using Bayes' Theorem, which relates current evidence to prior probability.
3. **Training:** During training, the model learns the probabilities of the features by examining the training dataset. It calculates the likelihood of each feature appearing in each class.
4. **Prediction:** For a new data point, the model applies the probabilities it learned during training to estimate the likelihood of this point belonging to each class.
5. **Class Selection:** The model selects the class with the highest calculated probability for the new data point.
6. **Output:** Finally, it outputs the class label with the highest probability as its prediction.

Because of its ease of use and performance with high-dimensional data, the model shines in text categorization and spam filtering.

Q9. Procedure of the Genetic Algorithm

One search heuristic that takes its cues from Charles Darwin's idea of natural selection is the Genetic Algorithm (GA). In optimization and search issues, the GA is often used when obtaining a precise solution proves to be unrealistic or unattainable. The key to its success is finding the sweet spot between taking use of existing best practices and investigating novel genomic combinations. Several steps are involved in the procedure:

1. **Initial Population:** The process begins with a set of individuals, known as the population. Each individual represents a potential solution to the problem at hand and is characterized by a set of parameters (genes).
2. **Fitness Function:** Each individual's fitness is evaluated using a fitness function, which measures how close it comes to achieving the set goals.
3. **Selection:** Individuals are selected for reproduction based on their fitness. The fitter the individual, the higher the chance it has to be selected.
4. **Crossover:** Selected individuals (parents) are then crossed over to form a new generation. This involves swapping some genes between two parents to create offspring.
5. **Mutation:** To maintain genetic diversity within the population and avoid premature convergence, offspring may undergo mutations. This involves altering some genes randomly.
6. **Replacement:** The new generation of individuals replaces the old generation.
7. **Termination:** This process repeats for many generations, or until a termination condition is met (like a satisfactory fitness level or a maximum number of generations).

Q10. Explanation of Artificial Neural Networks, focusing on input & hidden layer weights and activation functions.

An example of a computer model that draws inspiration from the human brain is the Artificial Neural Network (ANN). They are like neurons in that they are composed of layers of linked nodes. During training, the weight of each edge (connection) in the network changes. The ANN's capacity to learn and provide predictions relies heavily on these weights.

Artificial Neural Networks (ANNs) are computational models inspired by the human brain. They consist of layers of interconnected nodes, resembling neurons. Each connection, or edge, in the network has a weight that adjusts during the learning process. These weights are key to the ANN's ability to learn and make predictions.

1. **Input Layer:** The first layer of an ANN, it receives the input data. Each node in this layer represents one feature of the input data.
2. **Hidden Layers:** These are layers between the input and output layers. The nodes in these layers apply transformations to the inputs they receive from the previous layer. The weights on these connections determine the strength and direction of these transformations.
3. **Activation Functions:** These are functions applied to the output of each node in the network. They decide whether a neuron should be activated or not, influencing the network's ability to learn complex patterns. Common examples include sigmoid, ReLU, and softmax functions.

Q11. The knight-knave problem

The Knight-Knave problem is a classic logic puzzle that involves differentiating truth-tellers (Knights) from liars (Knaves). In this problem, you encounter characters who either always tell the truth (Knights) or always lie (Knaves), and you must determine their true nature based on their statements. The challenge lies in using logical deduction to ascertain who is a Knight and who is a Knave based on the statements they make. This problem is often used to illustrate concepts in logic, reasoning, and sometimes in algorithm design, where determining the truth from limited or contradictory information is crucial.

Q12. Consider the five points: P1(0.07,083), P2(0.85,0.14), P3(0.66,0.89), P4(0.49,0.64) and P5(0.80,0.46). Group first two points considering single linkage hierarchical clustering technique.

In single linkage hierarchical clustering, the goal is to group points based on the minimum Euclidean distance between them. We will analyze a set of five points to determine the first group of points to be linked together.

Problem Statement

Given the following five points:

- P1: (0.07, 0.83)
- P2: (0.85, 0.14)
- P3: (0.66, 0.89)
- P4: (0.49, 0.64)
- P5: (0.80, 0.46)

our task is to identify the first two points that will be grouped together using the single linkage hierarchical clustering technique.

Method

The Euclidean distance between two points $P(x_1, y_1)$ and $Q(x_2, y_2)$ is given by the formula:

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Distance Calculations

The Euclidean distances between each pair of points are calculated as follows:

$$\text{Distance between P1 and P2} = \sqrt{(0.85 - 0.07)^2 + (0.14 - 0.83)^2}$$
$$\text{Distance between P1 and P3} = \sqrt{(0.66 - 0.07)^2 + (0.89 - 0.83)^2}$$
$$\text{Distance between P1 and P4} = \sqrt{(0.49 - 0.07)^2 + (0.64 - 0.83)^2}$$
$$\text{Distance between P1 and P5} = \sqrt{(0.80 - 0.07)^2 + (0.46 - 0.83)^2}$$
$$\text{Distance between P2 and P3} = \sqrt{(0.66 - 0.85)^2 + (0.89 - 0.14)^2}$$
$$\text{Distance between P2 and P4} = \sqrt{(0.49 - 0.85)^2 + (0.64 - 0.14)^2}$$
$$\text{Distance between P2 and P5} = \sqrt{(0.80 - 0.85)^2 + (0.46 - 0.14)^2}$$
$$\text{Distance between P3 and P4} = \sqrt{(0.49 - 0.66)^2 + (0.64 - 0.89)^2}$$
$$\text{Distance between P3 and P5} = \sqrt{(0.80 - 0.66)^2 + (0.46 - 0.89)^2}$$
$$\text{Distance between P4 and P5} = \sqrt{(0.80 - 0.49)^2 + (0.46 - 0.64)^2}$$

Results

The calculated distances are:

- Distance between P3 and P4: approximately 0.302.
- Distance between P2 and P5: approximately 0.324.

Q13. Supervised and Unsupervised learning

In the realm of machine learning, two fundamental approaches are Supervised and Unsupervised Learning. These methodologies are foundational to understanding how algorithms learn from data to make predictions or discover patterns.

Supervised Learning

Definition

Supervised learning is a type of machine learning where the algorithm is trained on a labeled dataset. This means that each example in the training set contains the input data as well as the corresponding correct output.

Characteristics

- The algorithm makes predictions based on the training data.
- It requires a dataset where the desired output is already known.
- Commonly used for classification and regression tasks.

Examples

- Predicting house prices based on various features (regression).
- Identifying if an email is spam or not (classification).

Unsupervised Learning

Unsupervised learning involves training an algorithm on a dataset without labeled responses. The goal is to explore the structure of the data to extract meaningful information.

Characteristics

- It discovers hidden patterns or intrinsic structures in input data.
- Labeled data is not required.
- Commonly used for clustering, association, and dimensionality reduction.

Examples

- Segmenting customers into different groups based on purchasing behavior (clustering).
- Finding items frequently bought together from a shopping dataset (association).

While supervised learning is driven by the goal of predicting outcomes based on known labels, unsupervised learning seeks to understand and model the underlying structure of unlabeled data. Both play crucial roles in the field of machine learning.

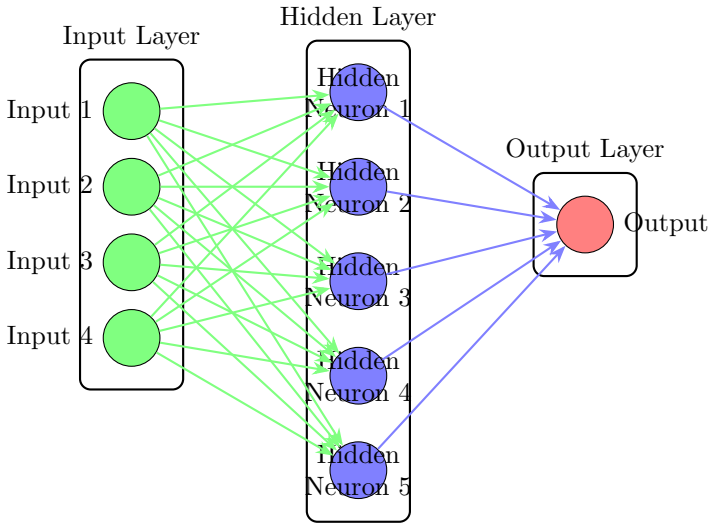
Q14. Draw the diagram of a three-layer artificial neural network for a classification task. There are 3 input features and 4 classes. Use 2 neurons in the hidden layer. ignore the bias units.

The network consists of an input layer, one hidden layer, and an output layer.

Neural Network Architecture

Diagram

The structure of the neural network is depicted below:



Description

The input layer consists of neurons representing the features of the input data. The hidden layer, crucial for the network's ability to learn complex patterns, processes inputs from the input layer. The output layer produces the final classification result.

Conclusion

This three-layer neural network is a basic architecture used for classification tasks. Its ability to learn from data makes it suitable for a wide range of applications in machine learning and artificial intelligence.

Q15. Write five name of supervised machine learning algorithm

Algorithm Name	**Uses**
Linear Regression	Predicting continuous outcomes, e.g., housing prices
Logistic Regression	Binary classification, e.g., email spam detection
Decision Trees	Classification and regression, e.g., customer segmentation
Support Vector Machines	Classification and regression, e.g., image classification
Random Forests	Ensemble learning for classification and regression tasks

Table 3: Supervised Machine Learning Algorithms and Their Uses

Q16. Write the below supervised machine learning algorithm is classification or regression.

Algorithm Name	**Classification**	**Regression**
Naïve Bayes	Yes	No
K-nearest Neighbor	Yes	Yes
Logistic Regression	Yes	No
Linear Regression	No	Yes
Support Vector Machine	Yes	Yes

Table 4: Classification of Supervised Machine Learning Algorithms

Naïve Bayes and **Logistic Regression** are typically used for classification problems. **Linear Regression** is used for regression tasks. **K-nearest Neighbor** and **Support Vector Machine** can be used for both classification and regression, depending on how they are implemented.

Q17. Weak and Strong learner Ensemble Learning in Machine learning

It is common practice in ensemble learning to join a number of poor learners in order to produce a strong learner. This is founded on the premise that a group of weak learners may collectively produce more accurate predictions than a single good learner, particularly when dealing with complicated situations. A number of different approaches, such as bagging, boosting, or stacking, may be used to accomplish the combination. Each of these approaches takes a distinctive approach to maximizing the capabilities of individual models in order to enhance overall performance.

1. **Weak Learner:** A weak learner is a model that performs slightly better than random guessing but is generally inaccurate and unreliable on its own. The strength of weak learners lies in their

simplicity and speed. They are often used in ensemble methods because their simplicity allows for a diversity of perspectives when combined.

2. **Strong Learner:** A strong learner, on the other hand, is a highly accurate and reliable model. It performs significantly better than random guessing and often approaches or achieves the best possible performance on a given task.

Q18. Given ML algorithm name, categorize them into a)supervised learning, a)unsupervised learning

Algorithm Name	Supervised Learning	Unsupervised Learning
Support Vector Machine	Yes	No
Linear Regression	Yes	No
Clustering	No	Yes
Dimensionality Reduction	No	Yes
Classification	Yes	No

Table 5: Categorization of Machine Learning Algorithms

This table differentiates between supervised and unsupervised learning methods by categorizing each algorithm as one of the two. For the purpose of training, supervised learning algorithms such as Support Vector Machine, Linear Regression, and Classification need data that has been labeled. Finding structure in the data may be accomplished with the help of unsupervised learning methods such as Clustering and Dimensionality Reduction. These algorithms do not need labeled data to function.

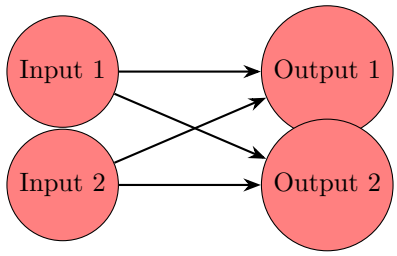
Q19. Draw a Diagram of one layer two input and two output neural network.

1 Neural Network Architecture

The neural network consists of a single layer with two input neurons and two output neurons.

1.1 Diagram

The structure of the neural network is depicted below:



This neural network diagram with one layer demonstrates the connections that exist between two neurons that are input and two neurons that are output. The study of such structures is essential for gaining a fundamental grasp of neural networks.

Q20. Reinforcement learning

In the field of machine learning, reinforcement learning (RL) refers to a sort of learning in which an agent learns to make choices by carrying out activities in an environment in order to gain the highest possible cumulative reward. To name a few key points:

Key Points

- **Agent and Environment:** The agent performs actions within an environment, receiving rewards or penalties as feedback.
- **Trial and Error:** Learning is achieved primarily through trial and error, reinforcing successful actions.
- **Reward System:** The agent's decision-making is driven by a system of rewards, seeking to maximize long-term gains.
- **Applications:** RL is applicable in robotics, gaming, autonomous navigation, and more.
- **Algorithms:** Commonly used algorithms include Q-learning and Policy Gradient methods.

Q21. PEAS in ML

PEAS (Performance measure, Environment, Actuators, Sensors) is a framework used for designing intelligent agents in Machine Learning. It comprises the following components:

1. **Performance Measure:** Defines the criteria for the agent's success, assessing how well it achieves its objectives.
2. **Environment:** The external context or space in which the agent operates, encompassing all challenges and stimuli it faces.
3. **Actuators:** The means through which the agent interacts with its environment, which can be output devices or actions.
4. **Sensors:** The tools or methods the agent uses to perceive its surroundings, essential for informed decision-making.

Understanding PEAS is crucial for developing systems that effectively interact with complex and dynamic environments.

Q22. Decision tree basic

A decision tree is a flowchart-like tree structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. It is one of the most popular machine learning algorithms used for both classification and regression tasks.

Components

A decision tree consists of:

- **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is the process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, it is called a decision node.
- **Leaf/Terminal Node:** Nodes that do not split are called Leaf or Terminal nodes.
- **Pruning:** Removing sub-nodes from a decision node is called pruning. It is the opposite of splitting.
- **Branch / Sub-Tree:** A subsection of the entire tree is called a branch or sub-tree.

How it Works

The decision of making strategic splits significantly affects a tree's accuracy. The decision criteria are different for classification and regression trees. Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, purity of the node increases with respect to the target variable.

Algorithm

The core algorithm for building decision trees is the greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner. The most popular algorithms are:

- **ID3 (Iterative Dichotomiser 3)**
- **C4.5 (successor of ID3)**
- **CART (Classification And Regression Tree)**

These algorithms use various metrics like Gini Index, Information Gain, and Chi-square, among others, to decide the best split.

Advantages and Disadvantages

1.2 Advantages

- Simple to understand, interpret, and visualize.
- Requires little data preparation.
- Able to handle both numerical and categorical data.
- Performs well with large datasets.

1.3 Disadvantages

- Overfitting can occur with complex trees.
- Small variations in data can result in a different tree being generated.
- Decision trees can be biased with imbalanced datasets.

Decision trees are a versatile algorithm used widely in classification and regression tasks. Their ease of use and understanding makes them popular, but care must be taken to handle issues like overfitting and bias.

Q23. Naïve Baise Basic

Naïve Bayes is a simple yet effective classification algorithm based on Bayes' Theorem with the assumption of independence among predictors. It is particularly useful in text classification and has advantages in terms of efficiency and ease of implementation.

Bayes' Theorem

Bayes' Theorem is the basis of the Naïve Bayes algorithm. It is expressed as:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Where:

- $P(A|B)$ is the probability of hypothesis A given the data B (posterior probability).
- $P(B|A)$ is the probability of the data B given that the hypothesis A is true.
- $P(A)$ is the probability of hypothesis A being true (irrespective of the data).
- $P(B)$ is the probability of the data (irrespective of the hypothesis).

Naïve Bayes Classifier

The Naïve Bayes classifier applies this theorem with the "naïve" assumption of conditional independence between every pair of features given the value of the class variable. The classifier combines this model with a decision rule, often choosing the hypothesis that is most probable.

Model Training

During training, the algorithm estimates the parameters of the probability distribution for each feature and class label. These parameters are then used to make predictions on new data.

Prediction

Given a new instance to classify, the algorithm calculates the posterior probability for each class label and selects the label with the highest probability.

Advantages and Disadvantages

Advantages:

- Efficient and easy to implement.
- Requires a small amount of training data to estimate parameters.
- Performs well with categorical input variables.

Disadvantages:

- Relies on an often unrealistic assumption of feature independence.
- Less effective with numerical features unless the distribution is known.

Naïve Bayes is a fundamental classification technique in machine learning, especially useful in text analysis. Despite its simplicity, it can yield powerful results and serves as a baseline for many classification tasks.