1. What does one mean by the term "machine learning"?  
Ans. Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed for every specific task. It involves the development of techniques that allow machines to automatically learn from and analyze large amounts of data, identify patterns, and make data-driven predictions or decisions.

2.Can you think of 4 distinct types of issues where it shines?  
Ans. Predictive Analysis, Recommendation system, Natural Language processing, Image object recommendation.

3.What is a labeled training set, and how does it work?  
Ans. A labeled training set is a collection of data used to train a machine learning model where each data instance (or example) is associated with a known label or target value. The labels provide the ground truth information or desired outputs that the model aims to learn from during the training process.

The steps involved in working with a labeled training set are as follows:

Data Collection: Collect a representative set of data instances, ensuring they cover the range of input features and target values relevant to the problem at hand.

Data Preprocessing: Prepare the data for training by performing tasks such as data cleaning, removing outliers, handling missing values, and transforming the features into a suitable format for the chosen machine learning algorithm.

Feature Selection: Extract or select the relevant features from the data that are likely to have a strong influence on the target variable. This process helps to reduce dimensionality and focus on the most informative features.

Splitting the Data: Split the labeled training set into two subsets: the training set and the validation set. The training set is used to train the model, while the validation set is used to evaluate the model's performance during training and tune its hyperparameters.

Model Training: Feed the training set into the machine learning algorithm along with the corresponding labels. The model learns from the labeled examples, updating its internal parameters through an optimization process (such as gradient descent) to minimize the prediction errors.

Model Evaluation: After training, assess the model's performance using the validation set. Calculate metrics such as accuracy, precision, recall, or mean squared error, depending on the problem type. This evaluation helps to measure how well the model generalizes to unseen data and identify any issues like overfitting or underfitting.

Model Deployment and Prediction: Once satisfied with the model's performance, it can be deployed to make predictions on new, unseen data. The model uses the learned patterns and relationships to generate predictions or classifications based on the input features.

4.What are the two most important tasks that are supervised?  
Ans. Classification: Classification involves assigning data instances to predefined categories or classes based on their features. The goal is to build a model that can accurately classify new, unseen instances into the appropriate class. In classification tasks, the labeled training set consists of input features and corresponding class labels. The model learns to discriminate between different classes by identifying patterns and decision boundaries in the feature space. Classification is widely used in various domains, including spam detection, sentiment analysis, medical diagnosis, image recognition, and fraud detection.

Regression: Regression is concerned with predicting a continuous or numerical value based on input features. The objective is to build a model that can estimate or approximate a target variable. In regression tasks, the labeled training set contains input features and their corresponding numeric target values. The model learns the relationships between the features and the target variable, allowing it to make predictions on new instances. Regression is utilized in diverse fields such as sales forecasting, housing price prediction, demand estimation, stock market analysis, and weather prediction.

5.Can you think of four examples of unsupervised tasks?  
Ans. Clustering: Clustering is the process of grouping similar data instances together based on their inherent patterns or similarities. The goal is to discover underlying structures in the data without any prior knowledge of the class labels or target values. Common clustering algorithms include k-means clustering, hierarchical clustering, and DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Clustering is used in customer segmentation, image segmentation, document clustering, anomaly detection, and recommendation systems.

Dimensionality Reduction: Dimensionality reduction aims to reduce the number of input features while retaining the most relevant information. It is particularly useful when dealing with high-dimensional data that may suffer from the curse of dimensionality. Techniques like Principal Component Analysis (PCA) and t-SNE (t-Distributed Stochastic Neighbor Embedding) are commonly used for dimensionality reduction. This task is employed in visualization, feature selection, noise reduction, and speeding up subsequent learning tasks.

Anomaly Detection: Anomaly detection involves identifying unusual or abnormal instances in a dataset that deviate from the expected or normal behaviour. The goal is to detect data points or patterns that are significantly different from most of the data. Unsupervised anomaly detection methods include statistical approaches, clustering-based methods, and density estimation techniques like One-Class SVM (Support Vector Machines) and Isolation Forest. Anomaly detection finds applications in fraud detection, network intrusion detection, system health monitoring, and outlier detection in sensor data.

Association Rule Mining: Association rule mining aims to discover interesting relationships, patterns, or associations among items in a transactional or market basket dataset. It identifies frequently co-occurring items and generates rules like "if item A is purchased, then item B is also likely to be purchased." The Apriori algorithm and FP-Growth algorithm are commonly used for association rule mining. This task is widely used in market basket analysis, recommendation systems, cross-selling strategies, and customer behavior analysis.

6.State the machine learning model that would be best to make a robot walk through various unfamiliar terrains?  
Ans. Reinforcement learning.

7.Which algorithm will you use to divide your customers into different groups?  
Ans. To divide customers into different groups, a commonly used algorithm is k-means clustering. K-means clustering is an unsupervised learning algorithm that partitions data into k distinct clusters based on their similarity.

8.Will you consider the problem of spam detection to be a supervised or unsupervised learning problem?  
Ans. To divide customers into different groups, a commonly used algorithm is k-means clustering. K-means clustering is an unsupervised learning algorithm that partitions data into k distinct clusters based on their similarity.

9.What is the concept of an online learning system?  
Ans. The concept of an online learning system refers to a type of machine learning system that can continuously update and adapt its model in real-time as new data becomes available. Unlike traditional batch learning systems where the model is trained offline on a fixed dataset, online learning systems can incrementally learn from streaming or dynamically changing data.

10.What is out-of-core learning, and how does it differ from core learning?  
Ans. Out-of-core learning, also known as "online learning with external memory" or "incremental learning," is a technique used in machine learning to train models on large datasets that do not fit into the available memory (RAM) of a computer. It is a strategy for handling data that exceeds the memory capacity by sequentially reading and processing smaller subsets of data, or mini batches, from disk storage.

11.What kind of learning algorithm makes predictions using a similarity measure?  
Ans. The learning algorithm that makes predictions using a similarity measure is called k-nearest neighbours (k-NN). K-nearest neighbours is a supervised learning algorithm used for both classification and regression tasks.

12.What's the difference between a model parameter and a hyperparameter in a learning algorithm?  
Ans. Model Parameters:

Model parameters are internal variables that the learning algorithm optimizes during the training process.

They directly influence the behaviour of the model and are learned from the training data.

Model parameters are typically determined by the learning algorithm itself and are updated iteratively to minimize a defined objective function.

Examples of model parameters include the weights and biases in a neural network, the coefficients in linear regression, or the splitting criteria in decision trees.

Hyperparameters:

Hyperparameters, on the other hand, are external to the model and cannot be learned directly from the training data.

They are set before the learning algorithm is applied and determine the behaviour and characteristics of the learning algorithm itself.

Hyperparameters influence the learning algorithm's performance, complexity, and generalization ability.

Examples of hyperparameters include the learning rate in gradient descent, the number of hidden layers in a neural network, the regularization parameter in regularization techniques, or the depth of a decision tree.

Hyperparameters are typically set by the user or determined through a hyperparameter tuning process.

13.What are the criteria that model-based learning algorithms look for? What is the most popular method they use to achieve success? What method do they use to make predictions?  
Ans. Model-based learning algorithms look for patterns, relationships, or structures in the training data to build a model that can make predictions or perform tasks. The criteria that these algorithms typically aim to optimize or achieve include:

Accuracy: Model-based algorithms strive to achieve high prediction accuracy or task performance on unseen or test data. The goal is to minimize the difference between the predicted outcomes or outputs and the actual values or labels.

Generalization: Model-based algorithms aim to generalize well to unseen data by capturing the underlying patterns or relationships in the training data without overfitting. Overfitting occurs when the model memorizes the training data too closely, leading to poor performance on new, unseen instances.

Complexity: Model-based algorithms consider the complexity of the learned model. They seek to strike a balance between capturing the essential patterns in the data and avoiding excessive complexity, which can lead to overfitting and poor generalization.

The most popular method used by model-based learning algorithms to achieve success is to formulate an optimization problem. The algorithms define an objective function or a loss function that quantifies the discrepancy between the model's predictions and the true outcomes. The optimization process then aims to minimize this loss function by adjusting the model's parameters.

14.Can you name four of the most important Machine Learning challenges?  
Ans. Here are four of the most important challenges in machine learning:

Data Quality and Quantity: Machine learning heavily relies on high-quality and representative data for training accurate models. However, acquiring labelled data can be expensive and time-consuming. Challenges in data quality include missing values, outliers, noise, and biased or unrepresentative samples. Insufficient data quantity can lead to overfitting, where the model performs well on the training data but fails to generalize to unseen examples.

Model Selection and Hyperparameter Tuning: Choosing the right machine learning model and configuring its hyperparameters are crucial for achieving good performance. Selecting an appropriate model requires understanding the problem domain, the characteristics of the data, and the trade-offs between different algorithms. Tuning hyperparameters, such as learning rate, regularization strength, or network architecture, can significantly impact the model's performance, and finding the optimal combination is a challenging optimization problem.

Overfitting and Underfitting: Overfitting occurs when a model learns the training data too closely, capturing noise or irrelevant patterns, which leads to poor generalization to unseen examples. Underfitting, on the other hand, occurs when the model is too simple to capture the underlying patterns in the data, resulting in low training and test performance. Balancing the model's complexity and capacity to generalize is a key challenge in machine learning.

Interpretability and Explainability: As machine learning is increasingly applied in critical domains, the interpretability and explainability of models have gained importance. Black-box models, such as deep neural networks, can achieve high accuracy but lack transparency in their decision-making process. Interpreting the model's predictions and understanding the factors influencing them is crucial for building trust, detecting bias, and ensuring ethical and fair deployment of machine learning systems.

15.What happens if the model performs well on the training data but fails to generalize the results to new situations? Can you think of three different options?  
Ans. If a model performs well on the training data but fails to generalize to new situations, it indicates a problem of overfitting, where the model has learned the specific patterns and noise in the training data too well, resulting in poor generalization. Here are three different options to address this issue:

Regularization: Regularization is a technique used to prevent overfitting by adding a penalty term to the model's objective function. This penalty discourages the model from assigning too much importance to complex or unnecessary features. Regularization methods, such as L1 regularization (Lasso), L2 regularization (Ridge), or elastic net, control the model's complexity and help in generalizing better to new situations. By tuning the regularization strength, the model can strike a balance between fitting the training data and avoiding overfitting.

Cross-validation: Cross-validation is a technique used to assess a model's performance and generalization ability. Instead of relying solely on the training set, cross-validation involves partitioning the data into multiple subsets, performing training and evaluation on different combinations of these subsets, and averaging the results. This approach provides a more robust estimate of the model's performance and helps identify potential overfitting. If the model consistently performs well on the training set but poorly on the validation or test sets, it suggests overfitting, and adjustments can be made accordingly.

Increasing Data Size: Insufficient data can contribute to overfitting, as the model may try to fit noise or irrelevant patterns. Increasing the size of the training data can help mitigate overfitting by providing more diverse examples for the model to learn from. With more data, the model can capture the underlying patterns better and generalize well to new situations. Data augmentation techniques, such as artificially generating new training instances or introducing variations, can also be beneficial in expanding the training dataset.

16.What exactly is a test set, and why would you need one?  
Ans. A test set, also known as a validation set or holdout set, is a subset of data that is independent of the training data and is used to assess the performance of a trained machine learning model. The test set is used to simulate real-world scenarios where the model encounters unseen data, and its predictions or performance are evaluated based on this new data.

The purpose of having a test set is to estimate the model's performance on unseen data and assess its ability to generalize beyond the training set. Here are a few key reasons why a test set is necessary:

Performance Evaluation: The test set allows you to objectively evaluate the performance of your trained model. By measuring the model's performance on the test set, you can obtain an unbiased estimate of how well it is likely to perform on new, unseen data. This evaluation helps you gauge the model's effectiveness, identify potential issues like overfitting, and compare different models or parameter settings.

Generalization Assessment: The ultimate goal of machine learning is to build models that can generalize well to new, unseen data. The test set provides a realistic assessment of the model's generalization ability. If the model performs well on the test set, it suggests that it has successfully captured the underlying patterns and can make accurate predictions on new instances.

Hyperparameter Tuning: The test set is often used in conjunction with cross-validation or other techniques for hyperparameter tuning. By splitting the data into training, validation, and test sets, you can iteratively train and validate models on the training and validation sets, adjust hyperparameters based on the validation set performance, and finally evaluate the selected model on the independent test set. This helps in selecting the best-performing model with optimal hyperparameters.

Preventing Overfitting: Having a separate test set is crucial to prevent overfitting. If you evaluate the model's performance on the same data used for training, it may give an overly optimistic estimate of the model's performance. The test set ensures that the model is evaluated on truly unseen data, providing a more accurate assessment of its ability to generalize.

17.What is a validation set's purpose?  
Ans. The purpose of a validation set, also known as a development set or holdout set, is to fine-tune and assess the performance of a machine learning model during the model development process. It serves as an intermediate set between the training set and the final test set. The validation set has several important purposes:

Model Selection: The validation set helps in comparing and selecting different models or variations of a model. During the model development phase, multiple models with different architectures, hyperparameters, or feature engineering techniques may be trained. The validation set is used to evaluate the performance of these models and select the one that performs best. By assessing the model's performance on the validation set, you can make informed decisions about the model's configuration.

Hyperparameter Tuning: Hyperparameters are settings or configurations of a learning algorithm that are not learned directly from the data but need to be set by the user. Examples of hyperparameters include the learning rate, regularization strength, number of hidden layers in a neural network, or the depth of a decision tree. The validation set is used to fine-tune these hyperparameters by training and evaluating the model with different settings. By comparing the model's performance on the validation set for different hyperparameter values, you can select the optimal combination that yields the best performance.

Early Stopping: In some cases, training a model for too many epochs or iterations can lead to overfitting. Overfitting occurs when the model becomes too specialized to the training data and performs poorly on new, unseen data. The validation set can be used to implement early stopping, a technique where the model's training is stopped when its performance on the validation set starts to deteriorate. This helps prevent overfitting and ensures that the model is trained for an appropriate number of iterations.

18.What precisely is the train-dev kit, when will you need it, how do you put it to use?

19.What could go wrong if you use the test set to tune hyperparameters?  
Ans. Using the test set to tune hyperparameters can lead to an over-optimistic estimation of the model's performance and can invalidate the unbiased evaluation of the model's generalization ability. Here are a few issues that can arise when the test set is used for hyperparameter tuning:

Overfitting to the Test Set: When hyperparameters are tuned based on the test set performance, the model's hyperparameters become tailored specifically to the test set. As a result, the model may become overfit to the test set, meaning it is optimized to perform well on that particular subset of data but may not generalize well to new, unseen data. The model's performance on the test set may no longer accurately reflect its performance on unseen data.

Loss of Unbiased Evaluation: The purpose of having a separate test set is to provide an unbiased evaluation of the model's performance on new, unseen data. However, if the test set is used for hyperparameter tuning, it becomes part of the model development process. Consequently, the test set is no longer independent, and its evaluation loses its unbiased nature. This makes it difficult to accurately estimate how the model will perform on truly unseen data.

Risk of Overfitting the Test Set: Iteratively tuning hyperparameters based on the test set performance increases the likelihood of finding hyperparameter configurations that perform well specifically on the test set. This can lead to a false sense of high performance and may result in selecting hyperparameters that are not truly optimal but are tailored to the test set's characteristics. As a consequence, the model's performance may suffer when applied to new, real-world data.