1. **What does one mean by the term "machine learning"?**

Ans. Machine Learning refers to the use of algorithms and statistical models that enable computer systems to improve and statistical models that enable computer systems to improve and learn from experience without being explicitly programmed. It is unlocking insights from data and powering our AI-driven world.

The key aspects of Machine Learning are :

* It allows computers to find patterns and insights from large amount of data, rather than programming explicitly , such as using if else conditions.
* The algorithms iteratively learn from data , improving their performance over time.
* Automatically detect co-relations, that will be difficult for human on large data

1. **Can you think of 4 distinct types of issues where it shines?**

Use cases are as follows

* Clustering Problems : Recommender systems by streaming platforms like Netflix , amazon prime, to engage customers by showing them appropriate contents by analysing previous search and data of the user. Clustering algorithms are used in this system.
* Regression Problem : Housing price prediction at particular areas by analysing price trends over time.
* Classification Problem : Such as credit card default prediction, to know whether credit card holder default or not. Model can be created using old data from financial institutes. It can lot of losses occurring due to credit card defaults.
* Computer Vision Problem : Autonomous Driving Cars, Devices for specially abled people , object detection. Image classifier. Generative art creation.
* Natural Language Processing : Machine translation , speech recognition , music creation

1. **What is a labelled training set, and how does it work?**

Supervised machine learning requires labelled training data for training of model.

A training set contains examples or observations mapped to expected outputs.

Observations have features(variables , attributes) and labels (target variable)

More high Quality labelled data produces better model performance.

Model learn the relationships between features and labels.

For example, in an image classification task, the features could be the pixel values, and the label

could be the name of the object in the image.

Labels can be manually added by human annotators or come from sources like sensors. For image, text and audio data, labelling can be labour-intensive.

1. **What are the two most important tasks that are supervised?**

The two most important supervised learning tasks are:

* Classification - Predicting categorical labels or discrete classes. For example, classifying images as containing a cat or dog, labelling emails as spam or not spam, etc. Some common classification algorithms include logistic regression, naive Bayes classifier, support vector machines, random forests, and neural networks.
* Regression - Predicting continuous numerical values. For example, predicting house prices given their attributes, forecasting future sales revenue, etc. Common regression algorithms include linear regression, lasso and ridge regression, decision trees, and neural networks.

1. **Can you think of four examples of unsupervised tasks?**

* Clustering - Grouping unlabelled data points into clusters based on similarity. Algorithms include k-means, hierarchical clustering, etc. Used for customer segmentation, social network analysis, etc.
* Dimensionality Reduction - Reducing the number of variables in a dataset while retaining as much information as possible. Methods include PCA, matrix factorization, autoencoders. Useful for visualizing and understanding high-dimensional data.
* Anomaly Detection - Identifying outliers or unusual patterns that don't conform to expected behaviour. Has applications in fraud detection, fault detection, etc. Techniques involve statistical tests, proximity-based methods, neural networks.
* Generative Modelling - Generating new synthetic data similar to the training data.

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

The most suited machine learning method for a robot that must traverse a variety of unknown terrains is reinforcement learning. Here are a few primary causes:

\* Reinforcement learning enables a robot to learn skills and behaviours through erroneous interactions with its surroundings. For difficult terrain to be mastered, this capacity for experience-based learning is essential.

\* The robot can begin without any prior understanding of how to walk. It starts learning completely from scratch using a reward signal.

\* The robot can acquire reliable policies for walking across a variety of surfaces, including grass, gravel, sand, etc. by being given **the right rewards and punishments**.

\* Over time, the robot can improve at walking thanks to algorithms that are created to maximise cumulative long-term reward.

1. **Which algorithm will you use to divide your customers into different groups?**

I would use clustering algorithms to divide customers into different groups or segments. Some suitable options are:

* K-Means Clustering: A simple and commonly used clustering technique that partitions data points into k distinct clusters based on similarity. Effective for segmenting customers based on attributes like demographics, behavior, spending patterns etc.
* Hierarchical Clustering: Builds a hierarchy of clusters iteratively. Does not require pre-specifying the number of clusters like k-means. Useful when the natural groups in customer data are unknown.
* DBSCAN: A density-based clustering algorithm that can find arbitrary shaped clusters. Identifies clustered, sparse, and outlier points. Helps discover intrinsic customer segments.

1. **Will you consider the problem of spam detection to be a supervised or unsupervised learning problem?**

Spam detection is considered a supervised learning problem, since the model is trained on labeled data of spam and non-spam examples.

1. **What is the concept of an online learning system?**

Online Learning refers to a machine learning approach where the model is continuously trained and updated in real time as new data arrives sequentially.

Some key characteristics of online learning:

* Data comes in sequential order and the model is updated incrementaly rather than retrained on the full dataset.
* Useful when it is computationally expensive to retrain models on large dataset after every update.
* Often employs techniques like stochastic/mini-batch gradient rather than batch learning.
* Performs well where data distribution changes very often.
* Requires careful tuning of hyperparameters.

1. **What is out-of-core learning, and how does it differ from core learning?**

Out-of-core learning refers to machine learning on dataset that are too large to fit into computers main memory(RAM). It differs from core learning in following ways:

* Core learning assumes that the entire dataset fits in RAM, allowing fast access during model training. Out-of-core does not make this assumption.
* In out-of-core learning, the dataset is stored in external memory due to its massive size. Only small chunks are loaded into RAM at a time.
* Requires specialized algorithms and model architectures that can efficiently learn on disk-based data through sequential access.
* Examples include online learning , mini-batch stochastic gradient descent algorithms
* Core learning algorithms like ordinary gradient descent are inefficient for out-of-core setting as they require full dataset scans.
* Out-of-core frameworks like apache spark use distributed cluster computing to handle massive data.
* Applications dealing with big data used out-of -core approach.

1. **What kind of learning algorithm makes predictions using a similarity measure?**

The machine learning algorithms that make predictions using a similarity are known as instance-based or memory-based learning algorithms.

The key characteristics are:

* They compare new input to previously seen labelled examples in memory during prediction.
* Similarity measure like Euclidean distance , cosine similarity are used to find how close new instances are to existing ones.
* The label of the most similar memorized example is predicted as new label for said unlabelled instance.
* A well-known example of an instance-based learning algorithm is the k-nearest neighbors (k-NN) algorithm. In k-NN, to make a prediction for a new data point, the algorithm identifies the k nearest neighbours to that point in the training data (based on a similarity measure, often Euclidean distance), and then it predicts the output for the new point based on the labels of those k neighbours.

**12.What's the difference between a model parameter and a hyperparameter in a learning algorithm?**

|  |  |
| --- | --- |
| **Model Parameters** | **Hyperparameters** |
| Model parameters are learned directly from the training data. Examples include weights in neural networks , coefficients in linear regression etc. | Model hyperparameters are settings that govern model learning process. They are specified by users before training begins, Examples include learning rate, number of epochs in ANN, number of splits and max\_features in decision trees etc. |
| Parameters are optimized during training process to minimize loss function. | Choosing optimal hyperparameters ensures good model training performance. |
| Parameters are leaned automatically once hyperparameters are set. Learned via gradient descent , etc algorithms | Hyperparameters tuning is done via methods like grid/random search , Bayesian optimization. |
| Parameters are saved as part of the learned model. | Hyperparameters specify how the learning algorithm operates. |
| **Tuning both properly is important for maximizing model performance.** | |

**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?**

Criteria :

* Finding model parameters that minimize cost function on training data, such as means squared error.
* Achieving goods generalization by balancing model complexity and overfitting.
* Ensuring the model makes sense logically and theoretically..
* Optimization metrics like accuracy , F1 score on validation data.

Most Popular Method:

* The most common approach is using optimization algorithms like gradient descent to iteratively update parameters to minimize loss function. This fits the model to the training data.

Prediction Method:

* Once trained , model-based algorithms make predictions on new input data using the learned parameters.
* For example , a linear regression model uses the learned weights coefficients to make numeric predictions.
* A trained neural network uses matrix multiplications with the learned weights to generate outputs.

**14. Can you name four of the most important Machine Learning challenges?**

Four most important challenges in machine learning:

* Generalization : Model overfits training data , not generalizing well with new real world data is the most common challenge in ML. Regularization are used to tackle with generalization problem like L1 and L2 regularization.
* Explainability : Lack of model interpretability and explainability , especially with complex models like deep neural networks.
* Data Dependence : Model performance heavily relies on quantity and quality of training data. Collecting and labelling large data sets can be bit difficult and expensive.
* Resource and Computational Expenses: Some times it can be more expensive and become resource constraint to train models on large datasets.

**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?**

There are few potential reasons and solutions if a model performs well on training set but fails to generalize to new real world data:

* Overfitting : The model has learned the training data too closely , including noise and outliers. This cause it to perform poorly on new data.

Solutions : include getting more data , regularization like L1 and L2 , reducing model complexity.

* Selection Bias : The training data is not representative of new real world data. Distributions are not same in train and test dataset.

Solutions : Stratified shuffle technique, same distribution of train and test datasets.

* Covariate Shirt : The distribution of input variables has changes but output remains same.

Solution : retrain model on updated data.

* Data Drift : Relationship between inputs and outputs itself has changes over time,

Solution : Continual learning and updated training

**16.What exactly is a test set, and why would you need one?**

Before model training we need to split data into train and test set.

Train set is used to model training.

Test set is nothing but real world to used for calculation of accuracy in real world scenario.

We do not use testing set while model training process.

Train and test should be of same distributions.

If we don’t check accuracy of test set, we would not know how model will react in production environment.

**17.What is a validation set's purpose?**

**\***  To evaluate model performance after training has been done.

\* To estimate real world performance of the model.

\* To detect Overfitting. If there is a large gap between training and validation performance, it indicates the model is overfitting and needs regularization or more data.

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

In old era , there was very less data available for training and testing purpose, at that time we used to split data in train and test sets as 60:40 or 80:20 split. But in this modern era where we are having millions are data points for model training, we needed to change the way we split the data. To save computational resources as well as time , we started splitting data into train, dev and test set.

Example:

If we have around 10 millions records for model training , we can split dev as follows

Train : 90 %

Dev : 5%

Test : 5 %

Like above split, splitting of data is changed as per computational capabilities.

* Validation/dev set - Used to evaluate model performance during training and tune hyperparameters/other settings.
* Training set - Used to fit the parameters of the model.
* Test set - Used only once to evaluate the final model. Mimics a real-world dataset.

Note : Distribution of dev and test set should be same. The key ways the train-dev kit is utilized are:

* Train on train set, evaluate on validation set during model development.
* Tune hyperparameters to optimize validation performance.
* Choose the best model based on validation scores.
* Perform early stopping when validation metric stops improving.
* Final model is evaluated once on the test set.

**19.What could go wrong if you use the test set to tune hyperparameters?**

It will bias the final evaluation of the model and lead to optimistic estimated of real world performance.

Some key problem that could occur are :

* Model may end up overfitting indirectly to the test set , since hyperparameters are chosen to optimize test set metrics.
* This states the test set useless for unbiased final evaluation.
* The real world performance will be lower than the reported test set metrics, due to information leakage.
* Can not estimate overfitting and underfitting.
* Selection bias stays undetected.
* Training on test set doubts experimental integrity of the candidate.