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

Machine Learning means to train a machine using some statistical tools so that it can explore , analyze and perform predictions. It is a subset of Artificial Intelligence.

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

Weather prediction , Spam detection, Disease prediction, Recommendation system

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

Data labeling (or data annotation) is the process of adding target attributes to training data and labeling them so that a machine learning model can learn what predictions it is expected to make. This process is one of the stages in preparing data for supervised machine learning.

It is used to train the model .

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

Classification Task and Regression Task

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

* K-means clustering.
* KNN (k-nearest neighbors)
* Hierarchal clustering.
* Anomaly detection.
* Neural Networks.

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

Reinforced Learning

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

Clustering Algorithms

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

It is a supervised learning problem.

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

Online learning is an ML method in which data is available in sequential order, and we use it in order to predict future data at each time step.

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

It is a way to train your model on data that cannot fit your core memory.” Out-of-core learning refers to the machine learning algorithms working with data that cannot fit into a single machine's memory but can easily fit into some data storage, such as a local hard disk or web repository

in-core means the solver can hold the entire matrix in RAM for solving.

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

Instance-based learning algorithms use a measure of similarity to generalize to new cases. In an instance-based learning system, the algorithm learns the examples by heart, then uses the similarity measure to generalize.

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

A hyperparameter is a parameter of the learning algorithm, not the model. For example, in a simple linear regression problem our model is parameterized by theta which is a vector of weights. In order to find the best values for theta we have a cost function which is run repeatedly by the gradient descent algorithm. Gradient descent has a hyperparameter called alpha which is the learning rate of the algorithm.

1. 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?

The goal for a model-based algorithm is to be able to generalize to new examples. To do this, model based algorithms search for optimal values for the model's parameters, often called theta. This searching, or "learning", is what machine learning is all about. Model-based system learn by minimizing a cost function that measures how bad the system is at making predicitons on new data, plus a penalty for model complexity if the model is regularized. To make a prediction, a new instance's features are fed into a hypothesis function which uses the minimized theta found by repeatedly running the cost function.

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

* Not gathering enough data, or sampling noise. Sampling noise means we'll have non-representative data as a result of chance.
* Using a dataset that is not representative of the cases you want to generalize to. This is called sampling bias. For example, if you want to train an algorithm with "cat videos", and all your videos are from YouTube, you're actually training an algorithm to learn about "YouTube cat videos."
* Your dataset is full of missing values, outliers, and noise (poor measurments).
* The features in your dataset are irrelevant. Garbage in, garbage out.
  + Feature selection - choose the most relevant features from your dataset
  + Feature extraction - combine features in your dataset to generate a new, more useful feature
* When your model performs well on the training data, but not on test data, you've over fit your model. Models that suffer from overfitting do not generalize well to new examples. Overfitting happens when the model is too complex relative to the amount and noisiness of the data.
  + Try simplyfying the model by reducing the number of features in the data or constraining the parameters by reducing the degrees of freedom.
  + Gather more training data.
  + Reduce noise in the training data by fixing errors and removing outliers.
* When your model is too simple to learn the underlying structure of the data you've underfit your model.
  + Select a more powerful model with more parameters
  + Use feature engineering to feed better features to the model
  + Reduce the constraints of the model (increase degrees of freedom, reduce regularization parameter, etc.)

1. 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?

This is a case where the model is overfitting the training data. To couteract overfitting, we can reduce the complexity of the model by removing features or constraining the parameters. We could gather more data. Finally we can reduce noisiness in the data by fixing errors and removing outliers.

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

When we want to know how well our model generalizes to new cases we prefer to use a test set instead of actually deploying the system. To build the test set we split the training data (50-50, 60-40, 80-20 are common splits) into a training set and test set. Our model is training with the training set. Then we use the model to run predictions on the test set. Our error rate on the test set is called the generalization error or out-of-sample error. This error tells us how well our model performs on examples it has never seen before.

If the training error is low, but the generalization error is high, it means we're overfitting our model.

1. What is a validation set's purpose?

Let's say we have a linear model and we want to perform some hyperparameter tuning to reduce the generalization error. One way to do this 100 different models with 100 different hyperparameter values using the training set and finding the generalization error with the test set. You find the best hyperparameter value gives you 5% generalization error.

So you launch the model into production and find you're seeing 15% generalization error. This isn't going as expected. What happened?

The problem is that for each iteration of hyperparameter tuning, you measured the generalization error then updated the model using the same test set. In other words, your produced the best generalization error for the test set. The test set no longer represents cases the model hasn't seen before.

A common solution to this problem is to have a second holdout set called the validation set. You train multiple models with various hyperparameters using the training set, you select the model and hyperparameters that perform best on the validation set, and when you are happy about your model you run a single final test against the test set to get an estimate of the generalization error.

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

It is basically used to split the data into train and test datasets . Train is used for training the model whereas test is used to test the accuracy of the model.

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

Your model will not be generalizable to new examples.