|  |  |
| --- | --- |
| 1 | What does one mean by the term "machine learning"? |
| Ans. | Machine Learning is about building systems that can learn from  data. Learning means getting better at some tasks, given some  performance measure. |
| 2 | Can you think of 4 distinct types of issues where it shines? |
| Ans. | Machine Learning is great for complex problems for which we  have no algorithmic solution, to replace long lists of hand-tuned  rules, to build systems that adapt to fluctuating environments, and  finally to help humans learn (e.g., data mining). |
| 3 | What is a labeled training set, and how does it work? |
| Ans. | A labeled training set is a training set that contains the desired  solution (a.k.a. a label) for each instance. |
| 4 | What are the two most important tasks that are supervised? |
| Ans. | The two most common supervised tasks are regression and  classification. |
| 5 | Can you think of four examples of unsupervised tasks? |
| Ans. | Common unsupervised tasks include clustering, visualization,  dimensionality reduction, and association rule learning. |
| 6 | State the machine learning model that would be best to make a robot walk through various unfamiliar terrains? |
| Ans. | Reinforcement Learning is likely to perform best if we want a  robot to learn to walk in various unknown terrains, since this is  typically the type of problem that Reinforcement Learning  tackles. It might be possible to express the problem as a  supervised or semisupervised learning problem, but it would be  less natural. |
| 7 | Which algorithm will you use to divide your customers into different groups? |
| Ans. | If you don’t know how to define the groups, then you can use a  clustering algorithm (unsupervised learning) to segment your  customers into clusters of similar customers. However, if you  know what groups you would like to have, then you can feed  many examples of each group to a classification algorithm  (supervised learning), and it will classify all your customers into  these groups. |
| 8 | Will you consider the problem of spam detection to be a supervised or unsupervised learning problem? |
| Ans. | Spam detection is a typical supervised learning problem: the  algorithm is fed many emails along with their labels (spam or not  spam). |
| 9 | What is the concept of an online learning system? |
| Ans. | An online learning system can learn incrementally, as opposed to  a batch learning system. This makes it capable of adapting rapidly  to both changing data and autonomous systems, and of training on  very large quantities of data. |
| 10 | What is out-of-core learning, and how does it differ from core learning? |
| Ans. | Out-of-core algorithms can handle vast quantities of data that  cannot fit in a computer’s main memory. An out-of-core learning  algorithm chops the data into mini-batches and uses online  learning techniques to learn from these mini-batches. |
| 11 | What kind of learning algorithm makes predictions using a similarity measure? |
| Ans. | An instance-based learning system learns the training data by  heart; then, when given a new instance, it uses a similarity  measure to find the most similar learned instances and uses them  to make predictions. |
| 12 | What's the difference between a model parameter and a hyperparameter in a learning algorithm? |
| Ans. | A model has one or more model parameters that determine what it  will predict given a new instance (e.g., the slope of a linear  model). A learning algorithm tries to find optimal values for these  parameters such that the model generalizes well to new instances.  A hyperparameter is a parameter of the learning algorithm itself,  not of the model (e.g., the amount of regularization to apply). |
| 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 search for an optimal value for  the model parameters such that the model will generalize well to  new instances. We usually train such systems by minimizing a  cost function that measures how bad the system is at making  predictions on the training data, plus a penalty for model  complexity if the model is regularized. To make predictions, we  feed the new instance’s features into the model’s prediction  function, using the parameter values found by the learning  algorithm. |
| 14 | Can you name four of the most important Machine Learning challenges? |
| Ans. | Some of the main challenges in Machine Learning are the lack of  data, poor data quality, nonrepresentative data, uninformative  features, excessively simple models that underfit the training  data, and excessively complex models that overfit the data. |
| 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 great on the training data but generalizes  poorly to new instances, the model is likely overfitting the  training data (or we got extremely lucky on the training data).  Possible solutions to overfitting are getting more data,  simplifying the model (selecting a simpler algorithm, reducing  the number of parameters or features used, or regularizing the  model), or reducing the noise in the training data. |
| 16 | What exactly is a test set, and why would you need one? |
| Ans. | A test set is used to estimate the generalization error that a model  will make on new instances, before the model is launched in  production. |
| 17 | What is a validation set's purpose? |
| Ans. | A validation set is used to compare models. It makes it possible to  select the best model and tune the hyperparameters. |
| 18 | What precisely is the train-dev kit, when will you need it, how do you put it to use? |
| Ans. | The train-dev set is used when there is a risk of mismatch  between the training data and the data used in the validation and  test datasets (which should always be as close as possible to the  data used once the model is in production). The train-dev set is a  part of the training set that’s held out (the model is not trained on  it). The model is trained on the rest of the training set, and  evaluated on both the train-dev set and the validation set. If the  model performs well on the training set but not on the train-dev  set, then the model is likely overfitting the training set. If it  performs well on both the training set and the train-dev set, but  not on the validation set, then there is probably a significant data  mismatch between the training data and the validation + test data,  and you should try to improve the training data to make it look  more like the validation + test data. |
| 19 | What could go wrong if you use the test set to tune hyperparameters? |
| Ans. | If you tune hyperparameters using the test set, you risk overfitting  the test set, and the generalization error you measure will be  optimistic (you may launch a model that performs worse than you  expect). |