**1. What is Semi-supervised Machine Learning?**

Semi-supervised learning is the blend of supervised and unsupervised learning. The algorithm is trained on a mix of labelled and unlabelled data. Generally, it is utilized when we have a very small, labelled dataset and a large unlabelled dataset.

In simple terms, the unsupervised algorithm is used to create clusters and by using existing labelled data to label the rest of the unlabelled data. A Semi-supervised algorithm assumes continuity assumption, cluster assumption, and manifold assumption.

It is generally used to save the cost of acquiring labelled data. For example, protein sequence classification, automatic speech recognition, and self-driving cars.

**2. How do you choose which algorithm to use for a dataset?**

Apart from the dataset, you need a business use case or application requirements. You can apply supervised and unsupervised learning to the same data.

Generally:

* Supervised learning algorithms require labelled data.
  + Regression algorithms require continuous numerical targets.
  + Classification algorithms require categorical targets.
* Unsupervised learning algorithms require unlabelled data.
* Semi-supervised learning requires the combination of labelled and unlabelled datasets.
* Reinforcement learning algorithms require environment, agent, state, and reward data.

**3. Explain the K Nearest Neighbour Algorithm.**

The K Nearest Neighbour (KNN) is a supervised learning classifier. It uses proximity to classify labels or predict the grouping of individual data points. We can use it for regression and classification. KNN algorithm is non-parametric, meaning it doesn't make an underlying assumption of data distribution.

In the KNN classifier:

* We find K-neighbours nearest to the white point. In the example below, we chose k=5.
* To find the five nearest neighbours, we calculate the Euclidean distance between the white point and the others. Then, we chose the 5 points closest to the white point.
* There are three red and two green points at K=5. Since the red has a majority, we assign a red label to it.

**4. What is Feature Importance in machine learning, and how do you determine it?**

Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. It plays a critical role in understanding the data's underlying structure, the behaviour of the model, and making the model more interpretable.

There are several methods to determine feature importance:

1. **Model-based Importance**: Certain algorithms like Decision Trees and Random Forests provide built-in methods to evaluate feature importance. For example, Random Forests calculate the decrease in node impurity weighted by the probability of reaching that node, averaged over all trees.
2. **Permutation Importance**: This involves shuffling individual variables in the validation set and observing the effect on model performance. A significant decrease in model performance indicates high importance.
3. **SHAP:** This approach uses game theory to measure the contribution of each feature to the prediction in a complex model. [SHAP values](https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability) provide a deep insight into the model's behaviour and are particularly useful for complex models like gradient boosting machines or neural networks.
4. **Correlation Coefficients**: Simple statistical measures like Pearson or Spearman correlation can provide insights into the linear relationship between each feature and the target variable.

Understanding feature importance is crucial for model optimization, reducing overfitting by removing non-informative features, and improving model interpretability, especially in domains where understanding the model's decision process is critical.

**5. Is it true that we need to scale our feature values when they vary greatly?**

Yes. Most of the algorithms use Euclidean distance between data points, and if the feature value varies greatly, the results will be quite different. In most cases, outliers cause machine learning models to perform worse on the test dataset.

We also use feature scaling to reduce convergence time. It will take longer for gradient descent to reach local minima when features are not normalized.

**6. The model you have trained has a low bias and high variance. How would you deal with it?**

Low bias occurs when the model is predicting values close to the actual value. It is mimicking the training dataset. The model has no generalization which means if the model is tested on unseen data, it will give poor results.

To fix these issues, we will use bagging algorithms as it divides a data set into subsets using randomized sampling. Then, we generate sets of models using these samples with a single algorithm. After that, we combine the model prediction using voting classification or averaging.

For high variance, we can use regularization techniques. It penalized higher model coefficients to lower model complexity. Furthermore, we can select the top features from the feature importance graph and train the model.

**7. Which cross-validation technique would you suggest for a time-series dataset and why?**

Cross-validation is used to evaluate model performance robustly and prevent overfitting. Generally, cross-validation techniques randomly pick samples from the data and split them into train and test data sets. The number of splits is based on the K value.

For example, if the K = 5, there will be four folds for the train and one for the test. It will repeat five times to measure the model performed on separate folds.

We cannot do it with a time series dataset because it doesn't make sense to use the value from the future to forecast the value of the past. There is a temporal dependency between observations, and we can only split the data in one direction so that the values of the test dataset are after the training set.

The diagram shows that time series data k fold split is unidirectional. The blue points are the training set, the red point is the test set, and the white is unused data. As we can observe with every iteration, we are moving forward with the training set while the test set remains in front of the training set, not randomly selected.

**8. Why can the inputs in computer vision problems get huge? Explain it with an example.**

Imagine an image of 250 X 250 and a fully connected hidden first layer with 1000 hidden units. For this image, the input features are 250 X 250 X 3 = 187,500, and the weight matrix at the first hidden layer will be 187,500 X 1000-dimensional matrix. These numbers are huge for storage and computation, and to combat this problem, we use convolution operations.

**9. When you have a small dataset, suggest a way to train a convolutional neural network.**

If you do not have enough data to train a convolutional neural network, you can use transfer learning to train your model and get state-of-the-art results. You need a pre-trained model which was trained on a general but larger dataset. After that, you will fine-tune it on newer data by training the last layers of the models.

Transfer learning allows data scientists to train models on smaller data by using fewer resources, computing, and storage. You can find open-source pre-trained models for various use cases easily, and most of them have a commercial license which means you can use them to create your application.

**10. What is Syntactic Analysis?**

Syntactic Analysis, also known as Syntax analysis or Parsing, is a text analysis that tells us the logical meaning behind the sentence or part of the sentence. It focuses on the relationship between words and the grammatical structure of sentences. You can also say that it is the processing of analyzing the natural language by using grammatical rules.

**11. Why do we need “Deep” Q learning?**

Simple Q learning is great. It solves the problem on a smaller scale, but on a larger scale, it fails.

Imagine if the environment has 1000 states and 1000 actions per state. We will require a Q table of millions of cells. The game of chess and Go will require an even bigger table. This is where Deep Q-learning comes for the rescue.

It utilizes a neural network to approximate the Q value function. The neural networks recipe states as an input and outputs the Q-value of all possible actions.

**12. What are the assumptions of linear regression?**

Linear regression is used to understand the relation between features (X) and target (y). Before we train the model, we need to meet a few assumptions:

1. The residuals are independent.
2. There is a linear relation between X independent variable and y dependent variable.
3. Constant residual variance at every level of X
4. The residuals are normally distributed.

**13. What is the activation function in Machine Learning?**

The activation function is a non-linear transformation in neural networks. We pass the input through the activation function before passing it to the next layer.

The net input value can be anything between -inf to +inf, and the neuron doesn't know how to bound the values, thus unable to decide the firing pattern. The activation function decides whether a neuron should be activated or not to bound the net input values.

**14.What is Overfitting, and How Can You Avoid It?**

The Overfitting is a situation that occurs when a model learns the training set too well, taking up random fluctuations in the training data as concepts. These impact the model’s ability to generalize and don’t apply to new data.

When a model is given the training data, it shows 100 percent accuracy—technically a slight loss. But, when we use the test data, there may be an error and low efficiency. This condition is known as overfitting.

There are multiple ways of avoiding overfitting, such as:

* Regularization. It involves a cost term for the features involved with the objective function.
* Making a simple model. With lesser variables and parameters, the variance can be reduced.
* Cross-validation methods like k-folds can also be used.
* If some model parameters are likely to cause overfitting, techniques for regularization like LASSO can be used that penalize these parameters.

**15. How Do You Handle Missing or Corrupted Data in a Dataset?**

One of the easiest ways to handle missing or corrupted data is to drop those rows or columns or replace them entirely with some other value.

There are two useful methods in Pandas:

* IsNull () and dropna () will help to find the columns/rows with missing data and drop them.
* Fillna () will replace the wrong values with a placeholder value.
* How Can You Choose a Classifier Based on a Training Set Data Size?
* When the training set is small, a model that has a right bias and low variance seems to work better because they are less likely to overfit.
* For example, [Naive Bayes](https://www.simplilearn.com/tutorials/machine-learning-tutorial/naive-bayes-classifier) works best when the training set is large. Models with low bias and high variance tend to perform better as they work fine with complex relationships.
* What is Deep Learning?
* The [Deep learning](https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning) is a subset of machine learning that involves systems that think and learn like humans using artificial neural networks. The term ‘deep’ comes from the fact that you can have several layers of neural networks.
* One of the primary [differences between machine learning and deep learning](https://www.simplilearn.com/machine-learning-vs-deep-learning-major-differences-you-need-to-know-article) is that feature engineering is done manually in machine learning. In the case of deep learning, the model consisting of neural networks will automatically determine which features to use (and which not to use).

**16.What Are the Applications of Supervised Machine Learning in Modern Businesses?**

Applications of supervised machine learning include:

* Email Spam Detection

Here we train the model using historical data that consists of emails categorized as spam or not spam. This labelled information is fed as input to the model.

* Healthcare Diagnosis

By providing images regarding a disease, a model can be trained to detect if a person is suffering from the disease or not.

* Sentiment Analysis

This refers to the process of using algorithms to mine documents and determine whether they’re positive, neutral, or negative in sentiment.

* Fraud Detection

By training the model to identify suspicious patterns, we can detect instances of possible fraud.

**17.What Are Unsupervised Machine Learning Techniques?**

There are two techniques used in unsupervised learning: clustering and association.

Clustering

Clustering problems involve data to be divided into subsets. These subsets, also called clusters, contain data that are like each other. Different clusters reveal different details about the objects, unlike classification or regression.

Association

In an association problem, we identify patterns of associations between different variables or items.

For example, an e-commerce website can suggest other items for you to buy, based on the prior purchases that you have made, spending habits, items in your Wishlist, other customers’ purchase habits, and so on.

**18.What is the Difference Between Supervised and Unsupervised Machine Learning?**

* Supervised learning - This model learns from the labelled data and makes a future prediction as output.
* Unsupervised learning - This model uses unlabelled input data and allows the algorithm to act on that information without guidance.

**19.What Is ‘naive’ in the Naive Bayes Classifier?**

* The classifier is called ‘naive’ because it makes assumptions that may or may not turn out to be correct.
* The algorithm assumes that the presence of one feature of a class is not related to the presence of any other feature (absolute independence of features), given the class variable.
* For instance, some fruit may be a cherry if it is red in colour and round, regardless of other features. This assumption may or may not be right.

**20.How Will You Know Which Machine Learning Algorithm to Choose for Your Classification Problem?**

While there is no fixed rule to choose an algorithm for a classification problem, you can follow these guidelines:

* If accuracy is a concern, test different algorithms and cross-validate them.
* If the training dataset is small, use models that have low variance and high bias.
* If the training dataset is large, use models that have high variance and little bias.