1. Recognize the differences between supervised, semi-supervised, and unsupervised learning.

Supervised learning

In supervised learning, we have a basic idea beforehand as to what the result is going to be. The input data is labeled and the main task of the model is to either map the output with the input labels or when the input label is mapped as a continuous output

Unsupervised learning

In unsupervised learning, the data is unlabeled and its goal is to find out the natural patterns present within data points in the given dataset. It does not have a feedback mechanism unlike supervised learning and hence this technique is known as unsupervised learning.

Semi-Supervised learning

Semi-supervised learning falls in-between supervised and unsupervised learning. Here, while training the model, the training dataset comprises of a small amount of labeled data and a large amount of unlabeled data. This can also be taken as an example for weak supervision.

2. Describe in detail any five examples of classification problems.

* Logistic regression.
* Decision trees.
* Random forest.
* XGBoost.
* Light GBM.
* Voting classifiers.
* Artificial neural networks.

3. Describe each phase of the classification process in detail.

The process of building a classification model typically involves the following steps:

Data preparation: This step involves cleaning and pre-processing the data, such as removing missing values and transforming the data into a format that can be used by the classification algorithm.

Model selection: This step involves choosing an appropriate classification algorithm based on the characteristics of the data and the desired outcome. Common algorithms include decision trees, k-nearest neighbors, and support vector machines.

Model training: This step involves using the training data to train the classification algorithm and build the model. The model is trained by adjusting its parameters to minimize the difference between the predicted class labels and the actual class labels.

Model evaluation: This step involves evaluating the performance of the classification model on a test dataset that is separate from the training data. This can be done by calculating metrics such as accuracy, precision, recall, and F1-score.

Model deployment: This step involves deploying the classification model in a production environment, where it can be used to make predictions on new instances.

4. Go through the SVM model in depth using various scenarios.

Support Vector Machine (SVM) is a supervised learning machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems, such as text classification. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the optimal hyper-plane that differentiates the two classes very well

5. What are some of the benefits and drawbacks of SVM?

Advantages of Support Vector Machine:

* SVM works relatively well when there is a clear margin of separation between classes.
* SVM is more effective in high dimensional spaces.
* SVM is effective in cases where the number of dimensions is greater than the number of samples.
* SVM is relatively memory efficient

Disadvantages of Support Vector Machine:

* SVM algorithm is not suitable for large data sets.
* SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
* In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
* As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.

6. Go over the kNN model in depth.

* Load the data
* Initialise the value of k
* For getting the predicted class, iterate from 1 to total number of training data points
* Calculate the distance between test data and each row of training dataset. Here we will use Euclidean distance as our distance metric since it’s the most popular method. The other distance function or metrics that can be used are manhattan distance, Minkowski distance, Chebyshev, cosine, etc. If there are categorical variables, hamming distance can be used.
* Sort the calculated distances in ascending order based on distance values
* Get top k rows from the sorted array
* Get the most frequent class of these rows
* Return the predicted class

7. Discuss the kNN algorithm’s error rate and validation error.

error rate initially decreases and reaches a minima. After the minima point, it then increase with increasing K. To get the optimal value of K, you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K

8. For kNN, talk about how to measure the difference between the test and training results.

* Get each characteristic from your dataset;
* Subtract each one, example, (line 1, column 5) — (line1,column5) = X …(line 1, column 13) — (line1,column13) = Z;
* After get the subtract of all columns, you will get all the results and sum it X+Y +Z… ;
* So you wil get the sum’s square root ;

9. Create the kNN algorithm.

# Import necessary modules

**from** sklearn.neighbors **import** KNeighborsClassifier

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.datasets **import** load\_iris

# Loading data

irisData **=** load\_iris()

# Create feature and target arrays

X **=** irisData.data

y **=** irisData.target

# Split into training and test set

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(

             X, y, test\_size **=** 0.2, random\_state**=**42)

knn **=** KNeighborsClassifier(n\_neighbors**=**7)

knn.fit(X\_train, y\_train)

# Predict on dataset which model has not seen before

print(knn.predict(X\_test))

10. What is a decision tree, exactly? What are the various kinds of nodes? Explain all in depth.

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes. In a Decision tree, there are **two nodes**, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches

11. Describe the different ways to scan a decision tree.

A decision tree model is formed using a hierarchy of branches. Each path from the root node through internal nodes to a leaf node represents a classification decision rule. These decision tree pathways can also be represented as 'if-then' rules.

12. Describe in depth the decision tree algorithm.

Tree depth is a measure of how many splits a tree can make before coming to a prediction. This process could be continued further with more splitting until the tree is as pure as possible. The problem with many repetitions of this process is that this can lead to a very deep classification tree with many nodes

13. In a decision tree, what is inductive bias? What would you do to stop overfitting?

Two approaches to avoiding overfitting are distinguished: pre-pruning (generating a tree with fewer branches than would otherwise be the case) and post-pruning (generating a tree in full and then removing parts of it). Results are given for pre-pruning using either a size or a maximum depth cutoff.

14.Explain advantages and disadvantages of using a decision tree?

Advantages of Decision Trees.

* Interpretability.
* Less Data Preparation.
* Non-Parametric.
* Versatility.
* Non-Linearity.

Disadvantages of Decision Tree.

* Overfitting.
* Feature Reduction & Data Resampling.
* Optimization.

15. Describe in depth the problems that are suitable for decision tree learning.

* instances are represented by attribute-value pairs.
* The target function has discrete output values.
* Disjunctive descriptions may be required.
* The training data may contain errors.
* The training data may contain missing attribute values.
* Suitable for classification.

16. Describe in depth the random forest model. What distinguishes a random forest?

Random Forest is one of the most popular and commonly used algorithms by Data Scientists. Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It performs better for classification and regression tasks

17. In a random forest, talk about OOB error and variable value.

The out-of-bag (OOB) error is the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated whilst being trained