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

ANSWER:

* **Supervised Learning:** Involves training a model on labeled data, where the input data is paired with corresponding output labels. The goal is to learn a mapping from inputs to outputs.
* **Semi-Supervised Learning:** Combines elements of supervised and unsupervised learning. It uses a small amount of labeled data along with a large amount of unlabeled data for training. The model learns from both labeled and unlabeled examples to improve performance.
* **Unsupervised Learning:** Involves training a model on unlabeled data, where the algorithm tries to find patterns or structure in the input data without explicit guidance. The goal is typically to discover hidden patterns or groupings within the data.

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

ANSWER: **Examples of Classification Problems:**

* Email spam detection
* Image classification (e.g., identifying animals in images)
* Sentiment analysis (e.g., classifying movie reviews as positive or negative)
* Disease diagnosis (e.g., classifying patients as healthy or diseased based on symptoms)
* Credit risk assessment (e.g., classifying applicants as low, medium, or high risk)

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

ANSWER:

* **Data Collection:** Gathering and preparing the dataset, including selecting relevant features and labeling the data.
* **Data Preprocessing:** Cleaning the data, handling missing values, and performing feature scaling or normalization.
* **Feature Selection:** Identifying the most relevant features for classification, which can involve techniques like correlation analysis or feature importance ranking.
* **Model Selection:** Choosing an appropriate classification algorithm based on the problem requirements and characteristics of the dataset.
* **Model Training:** Training the selected model on the labeled training data to learn the underlying patterns or relationships.
* **Model Evaluation:** Assessing the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score.
* **Model Deployment:** Deploying the trained model into production to make predictions on new, unseen data.

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

ANSWER: SVMs aim to find the optimal hyperplane that separates data points of different classes. Depending on the dataset and problem at hand, SVMs can:

* Linearly separate classes with a linear kernel.
* Handle non-linearly separable data using kernels such as polynomial, Gaussian (RBF), or sigmoid.
* Adjust parameters like the regularization parameter (C) to control the trade-off between maximizing the margin and minimizing misclassification.
* Handle multi-class classification using strategies like one-vs-one or one-vs-all.

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

ANSWER:

* **Benefits:**
  + Effective in high-dimensional spaces.
  + Versatile due to the kernel trick for non-linear decision boundaries.
  + Robust against overfitting, especially in high-dimensional spaces.
  + Memory efficient as it uses a subset of training points as support vectors.
* **Drawbacks:**
  + Choosing an appropriate kernel and parameters can be challenging.
  + Computationally intensive, especially for large datasets.
  + Sensitive to noise and outliers.
  + Interpretability can be limited compared to simpler models.

6. Go over the kNN model in depth.

ANSWER: kNN is a non-parametric and instance-based learning algorithm where prediction is based on the majority vote of its k nearest neighbors. Key aspects of kNN include:

* Choosing an appropriate value of k.
* Selecting a distance metric (e.g., Euclidean distance, Manhattan distance).
* Handling categorical and numerical features appropriately.
* Dealing with imbalanced datasets.

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

ANSWER: **kNN Algorithm's Error Rate and Validation Error:**

* The error rate of the kNN algorithm can be measured using metrics such as accuracy, precision, recall, or F1-score.
* Validation error is typically assessed using techniques like cross-validation, where the dataset is split into training and validation sets multiple times to evaluate the model's performance on different data subsets.

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

ANSWER: The difference between test and training results in kNN can be evaluated using metrics such as the error rate, accuracy, or other performance measures calculated on both the training and test datasets.

9. Create the kNN algorithm.

ANSWER:

function KNN(X\_train, y\_train, X\_test, k):

for each test instance x\_test:

compute distances between x\_test and all instances in X\_train

select the k nearest neighbors based on distances

predict the class label by majority voting among the k neighbors

return predicted labels for all test instances

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

ANSWER: A decision tree is a tree-like model where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents the outcome or class label. Types of nodes in a decision tree include:

* **Root Node:** The topmost node that represents the initial decision point.
* **Internal Node:** Represents a decision point based on a feature.
* **Leaf Node:** Represents the final decision or outcome.

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

* ANSWER: **Depth-First Search (DFS):** Traverses the decision tree from the root node to leaf nodes, exploring each branch fully before backtracking.
* **Breadth-First Search (BFS):** Traverses the decision tree level by level, exploring all nodes at each level before moving to the next level.

12. Describe in depth the decision tree algorithm.

ANSWER: The decision tree algorithm recursively splits the dataset into subsets based on the most significant attribute, creating a tree-like structure where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents the outcome or class label. The algorithm selects the best attribute for splitting at each node based on criteria like information gain or Gini impurity.

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

ANSWER:

* **Inductive Bias:** Decision tree algorithms have an inductive bias towards simpler trees, preferring shorter trees with fewer nodes to avoid overfitting.
* **Preventing Overfitting:** Techniques to prevent overfitting in decision trees include pruning, limiting the tree depth, and using ensemble methods like random forests.

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

ANSWER:

* **Advantages:**
  + Easy to understand and interpret, suitable for visual representation.
  + Handles both numerical and categorical data.
  + Requires minimal data preprocessing.
* **Disadvantages:**
  + Prone to overfitting, especially with deep trees.
  + Can be biased towards features with more levels or categories.
  + Instability - small variations in the data can lead to significantly different trees.

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

ANSWER: **Problems Suitable for Decision Tree Learning:**

* Problems with discrete and continuous input features.
* Problems where interpretability and explanation of decisions are important.
* Problems with non-linear relationships between features and target variable.

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

ANSWER: Random forest is an ensemble learning method based on decision trees. It builds multiple decision trees during training and outputs the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Random forest introduces randomness during tree construction to improve generalization and reduce overfitting.

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

ANSWER: **Random Forest: OOB Error and Variable Importance:**

* **OOB Error (Out-of-Bag Error):** The OOB error is an estimate of the model's performance on unseen data. It is calculated by evaluating the predictions made by each tree in the forest on the data points not used in its training (out-of-bag samples).
* **Variable Importance:** Random forests measure the importance of variables by computing how much the tree nodes that use a particular variable reduce impurity on average (e.g., Gini impurity or entropy).