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

Ans:

* Supervised Learning Builds a model based labelled data.
* Unsupervised Learning Builds a model based on a unlabelled data.
* Semi-Supervised Learning Builds a model based on a mix of labelled and unlabelled data. This sits between supervised and unsupervised learning approaches.

### Overview comparison between these methods

| **Category** | **Supervised** | **Unsupervised** | **Semi-supervised** |
| --- | --- | --- | --- |
| Input data | All data is labelled | All data is unlabelled | Partially labelled |
| Training? | External supervision | No supervision | ?? |
| Use | Calculate outcomes | Discover underlying patterns | ?? |
| Computational complexity | Simple | Complex | Depends |
| Accuracy | Higher | Lesser | Lesser |

| **Supervised** | **Unsupervised** | **Semi-supervised** |
| --- | --- | --- |
| Pre-processing of data may be time consuming | More time required by user e.g. for interpretation | Complex iterative process |
| Cannot give “unkown” information as per unsupervised learning | May result in less accurate predictions compared to supervised learning | Not as accurate as supervised learning |
| Cannot handle “complex tasks” | Computationally more complex that supervised learning | Cannot handle more “complex tasks” |

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

Ans:

Email Spam The goal is to predict whether an email is a *spam*and should be delivered to the Junk folder.

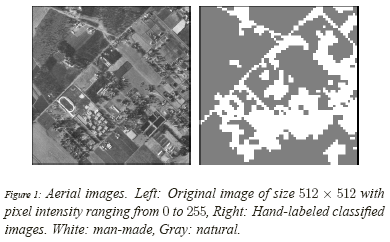
There are more than one method of identifying a mail as a spam. A simple method is discussed.The raw data comprises only the text part but ignores all images. Text is a simple sequence of words which is the input (X). The goal is to predict the binary response Y: spam or not.The first step is to process the raw data into a vector, which can be done in several ways. Given 57 most commonly occurring words and punctuation marks, then, in every e-mail message we would compute a relative frequency for each word, i.e., the percentage of times this word appears with respect to the total number of words in the email message.In the current example, 4601 email messages were considered in the training sample. These e-mail messages were identified as either a good e-mail or spam after reading the emails and assuming implicitly that human decision is perfect (an arguable point!). Relative frequency of the 57 most commonly used words and punctuation based on this set of emails was constructed. This is an example of supervised learning as in the training data the response Y is known. In the future when a new email message is received, the algorithm will analyze the text sequence and compute the relative frequency for these 57 identified words. This is the new input vector to be classified into spam or not through the learning algorithm.

### 2 - Handwritten Digit Recognition:The goal is to identify images of single digits 0 - 9 correctly.

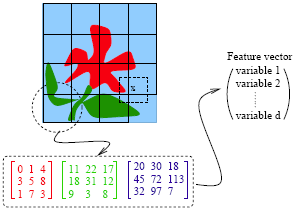
The raw data comprises images that are scaled segments from five-digit ZIP codes. In the diagram below every green box is one image. The original images are very small, containing only 16 × 16 pixels. For convenience the images below are enlarged, hence the pixelation or 'boxiness' of the numbers.Every image is to be identified as 0 or 1 or 2 ... or 9. Since the numbers are handwritten, the task is not trivial. For instance, a '5' sometimes can very much look like a '6', and '7' is sometimes confused with '1'.To the computer, an image is a matrix, and every pixel in the image corresponds to one entry in the matrix. Every entry is an integer ranging from a pixel intensity of 0 (black) to 255 (white). Hence the raw data can be submitted to the computer directly without any feature extraction. The image matrix was scanned row by row and then arranged into a large 256-dimensional vector. This is used as the input to train the classifier. Note that this is also a supervised learning algorithm where Y, the response, is multi-level and can take 10 values.

### 3 - Image segmentation

Here is a more complex example of an image processing problem. The satellite images are to be identified into man-made or natural regions. For instance, in the aerial images shown below, buildings are labeled as man-made, and the vegetation areas are labeled as natural.These grayscale images are much larger than the previous example. These images are 512 × 512 pixels and again because these are grayscale images we can present pixel intensity with numbers 0 to 255.



In the previous example of hand-written image identification, because of the small size of the images, no feature extraction was done. However in this problem feature extraction is necessary. A standard method of feature extraction in an image processing problem is to divide images into blocks of pixels or to form a neighborhood around each pixel. As is shown in the following diagram, after dividing the images into blocks of pixels or forming a neighborhood around each pixel, each block may be described by several features. As we have seen in the previous example, grayscale images can be represented by one matrix. Every entry in a greyscale image is an integer ranging from a pixel intensity of 0 (black) to 255 (white). Color images are represented by values of RGB (red, green and blue). Color images, therefore, are represented by 3 such matrices as seen below.



For each block, a few features (or statistics) may be computed using the color vectors for the pixels in the block. This set forms a feature vector for every block.

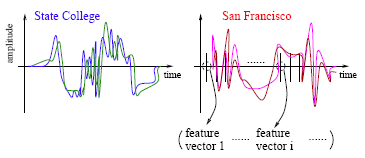
Examples of features:

* Average of R, G and B values for pixels in one block
* Variance of the brightness of the pixels (brightness is the average of RGB color values). Small variance indicates the block is visually smooth.

The feature vectors for the blocks sometimes are treated as independent samples from an unknown distribution. Ignoring f the spatial dependence among feature vectors results in performance loss. To make the learning algorithm efficient the spatial dependence needs to be exploited. Only then the accuracy in classification will improve.

### 4 - Speech Recognition

Another interesting example of data mining deals with speech recognition. For instance, if you call the University Park Airport, the system might ask you your flight number, or your origin and destination cities. The system does a very good job recognizing city names. This is a classification problem, in which each city name is a class. The number of classes is very big but finite.



The raw data involves voice amplitude sampled at discrete time points (a time sequence), which may be represented in the waveforms as shown above. In speech recognition, a very popular method is the *Hidden Markov Model*.

At every time point, one or more features, such as frequencies, are computed. The speech signal essentially becomes a sequence of frequency vectors. This sequence is assumed to be an instance of a hidden Markov model (HMM). An HMM can be estimated using multiple sample sequences under the same class (e.g., city name).HMM captures the time dependence of the feature vectors. The HMM has unspecified parameters that need to be estimated. Based on the sample sequences, model estimation takes place and an HMM is obtained. This HMM is like a mathematical signature for each word. Each city name, for example, will have a different signature. In the diagram above the signatures corresponding to State College and San Francisco are compared. It is possible that several models are constructed for one word or phrase. For instance, there may be a model for a female voice as opposed to another for a male voice.When a customer calls in for information and utters origin or destination city pairs, the system computes the likelihood of what the customer uttered under possibly thousands of models. The system finds the HMM that yields the maximum likelihood and identifies the word as the one associated with that HMM.

### 5 - DNA Expression Microarray

Our goal here is to identify disease or tissue types based on the gene expression levels.For each sample taken from a tissue of a particular disease type, the expression levels of a very large collection of genes are measured. The input data goes through a data cleaning process. Data cleaning may include but is certainly not limited to, normalization, elimination of noise and perhaps log-scale transformations. A large volume of literature exists on the topic of cleaning microarray data.

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

Ans: **Data Preprocessing:**. This involves handling missing values, dealing with outliers, and transforming the data into a format suitable for analysis. Data preprocessing also involves converting the data into numerical form, as most classification algorithms require numerical input.

Data Transformation: Data transformation involves scaling or normalizing the data to bring it into a common scale. This is done to ensure that all features have the same level of importance in the analysis.

**Feature Selection:**  
Feature selection involves identifying the most relevant attributes in the dataset for classification. This can be done using various techniques, such as correlation analysis, information gain, and principal component analysis.

**Principal Component Analysis:**is a technique used to reduce the dimensionality of the dataset. PCA identifies the most important features in the dataset and removes the redundant ones.

**Model Selection:**  
Model selection involves selecting the appropriate classification algorithm for the problem at hand. There are several algorithms available, such as decision trees, support vector machines, and neural networks.

.**Model Training:**  
Model training involves using the selected classification algorithm to learn the patterns in the data. The data is divided into a training set and a validation set. The model is trained using the training set, and its performance is evaluated on the validation set.

**Model Evaluation:**  
Model evaluation involves assessing the performance of the trained model on a test set. This is done to ensure that the model generalizes well

Classification is a widely used technique in data mining and is applied in a variety of domains, such as email filtering, sentiment analysis, and medical diagnosis.

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

Ans: Support Vector Machine(SVM) is a [supervised machine learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) algorithm used for both classification and regression. he objective of the SVM algorithm is to find a [hyperplane](https://www.geeksforgeeks.org/separating-hyperplanes-in-svm/) in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.T he objective of the SVM algorithm is to find a [hyperplane](https://www.geeksforgeeks.org/separating-hyperplanes-in-svm/) in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three. finds the maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these types of cases are called soft margins. When there is a soft margin to the data set, the SVM tries to minimize (1/margin+∧(∑penalty)). Hinge loss is a commonly used penalty. If no violations no hinge loss. violations hinge loss proportional to the distance of violation. The SVM kernel is a function that takes low-dimensional input space and transforms it into higher-dimensional space, ie it converts nonseparable problems to separable problems. Simply put the kernel, does some extremely complex data transformations and then finds out the process to separate the data based on the labels or outputs defined.A screenshot of a computer

Description automatically generated

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

Ans: benefits

SVM works better when the data is Linear

* It is more effective in high dimensions
* With the help of the kernel trick, we can solve any complex problem
* SVM is not sensitive to outliers
* Can help us with Image classification

Drawbacks:

* Choosing a good kernel is not easy
* It doesn’t show good results on a big dataset
* The SVM hyperparameters are Cost -C and gamma. It is not that easy to fine-tune these hyper-parameters. It is hard to visualize their impact

6. Go over the kNN model in depth.

Ans: The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



* Firstly, we will choose the number of neighbors, so we will choose the k=5.
* Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



* By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



* As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

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

Ans: The value of k determines the error rate if decrease in value of K to 1, our predictions become less stable.i.e A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.The accuracy decreases and the metric “F-Measure” becomes more sensitive to outliers. For better results, increase the value of K until the F-Measure value is higher than the threshold.Elbow method is used for finding out optimal value of K.

Validation error also depends on K to find out the optimal value of K in KNN. Start with the minimum value of k i.e, K=1, and run cross-validation, measure the accuracy.

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

Ans:

9. Create the kNN algorithm.

Ans: The following operations have happened during each iteration of the algorithm. For each of the unseen or test data point, the kNN classifier must:

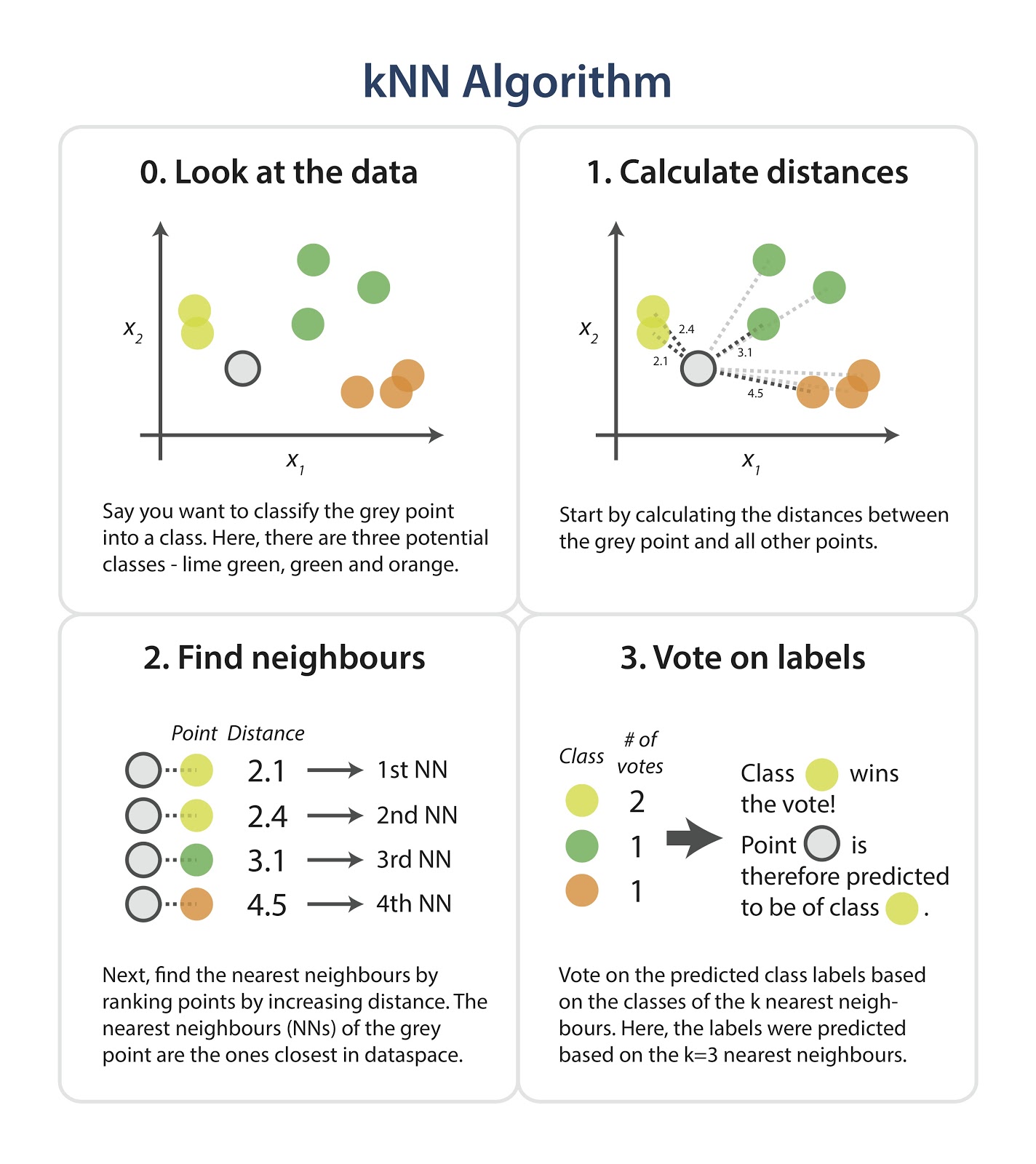
Step-1: Calculate the distances of test point to all points in the training set and store them

Step-2: Sort the calculated distances in increasing order

Step-3: Store the K nearest points from our training dataset

Step-4: Calculate the proportions of each class

Step-5: Assign the class with the highest proportion



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

Ans:Decision trees are a type of supervised learning algorithm which are used for mainly classification and regression.They have a tree like structure in which the internal nodes are "tests" for attributes and the branches are the results of the "tests". The leaf nodes will be the class labels i.e., the output of the learner. Given below is the basic structure of a decision tree.



* + Root Node: It is the topmost node in the tree,  which represents the complete dataset. It is the starting point of the decision-making process.
* Decision/Internal Node: A node that symbolizes a choice regarding an input feature. Branching off of internal nodes connects them to leaf nodes or other internal nodes.
* Leaf/Terminal Node: A node without any child nodes that indicates a class label or a numerical value.
* Splitting: The process of splitting a node into two or more sub-nodes using a split criterion and a selected feature.
* Branch/Sub-Tree: A subsection of the decision tree starts at an internal node and ends at the leaf nodes.
* Parent Node: The node that divides into one or more child nodes.
* Child Node: The nodes that emerge when a parent node is split.

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

Ans: the different ways to scan a decision tree:

#### Entropy:

Entropy is the measure of the degree of randomness or uncertainty in the dataset. In the case of classifications, It measures the randomness based on the distribution of class labels in the dataset.

#### Gini Impurity or index:

Gini Impurity is a score that evaluates how accurate a split is among the classified groups. The Gini Impurity evaluates a score in the range between 0 and 1, where 0 is when all observations belong to one class, and 1 is a random distribution of the elements within classes. In this case, we want to have a Gini index score as low as possible. Gini Index is the evaluation metric we shall use to evaluate our Decision Tree Model.

A screenshot of a computer

Description automatically generated

Here,

* pi is the proportion of elements in the set that belongs to the ith category.

#### Information Gain:

Information gain measures the reduction in entropy or variance that results from splitting a dataset based on a specific property. It is used in decision tree algorithms to determine the usefulness of a feature by partitioning the dataset into more homogeneous subsets with respect to the class labels or target variable. The higher the information gain, the more valuable the feature is in predicting the target variable.

The information gain of an attribute A, with respect to a dataset S, is calculated as follows:

A screenshot of a computer

Description automatically generated

where

* A is the specific attribute or class label
* |H| is the entropy of dataset sample S
* |HV| is the number of instances in the subset S that have the value v for attribute A

Information gain measures the reduction in entropy or variance achieved by partitioning the dataset on attribute A. The attribute that maximizes information gain is chosen as the splitting criterion for building the decision tree.

Information gain is used in both classification and regression decision trees. In classification, entropy is used as a measure of impurity, while in regression, variance is used as a measure of impurity. The information gain calculation remains the same in both cases, except that entropy or variance is used instead of entropy in the formula.

12. Describe in depth the decision tree algorithm.

Ans: : Decision trees are a type of supervised learning algorithm which are used for mainly classification and regression.They have a tree like structure in which the internal nodes are "tests" for attributes and the branches are the results of the "tests". The leaf nodes will be the class labels i.e., the output of the learner. Given below is the basic structure of a decision tree.

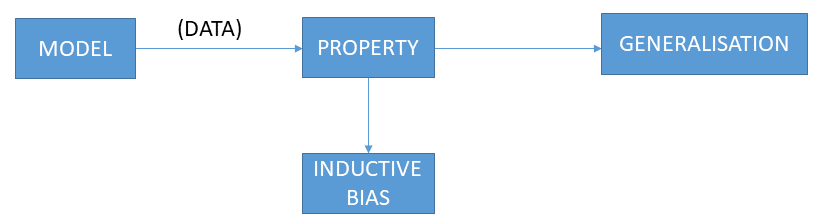


A tree can be “learned” by splitting the source set into subsets based on Attribute Selection Measures. Attribute selection measure (ASM) is a criterion used in decision tree algorithms to evaluate the usefulness of different attributes for splitting a dataset. The goal of ASM is to identify the attribute that will create the most homogeneous subsets of data after the split, thereby maximizing the information gain. This is done using information gain, Cal gini index or entropy. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of a decision tree classifier does not require any domain knowledge or parameter setting and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high-dimensional data.

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

Ans: Before learning a model given a data and a learning algorithm, there are a few assumptions a learner makes about the algorithm. These assumptions are called the inductive bias. It is like the property of the algorithm.

For eg. in the case of decision trees, the depth of the tress is the inductive bias. If the depth of the tree is too low, then there is too much generalisation in the model. Similarly, if the depth of the tree is too much, there is too less generalisation and while testing the model on a new example, we might reach a particular example used to train the model. This may give us incorrect results.



Avoid overfitting in Decision Trees ♣Two strategies: Stop growing the tree earlier, before perfect classification Allow the tree to overfit the data, and then post-prune the tree ♣Training and validation set ♣split the training in two parts (training and validation) and use validation to assess the utility of post-pruning ♣Reduced error pruning

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

Ans: Decision trees are able to generate understandable rules.

* Decision trees perform classification without requiring much computation.
* Decision trees are able to handle both continuous and categorical variables.
* Decision trees provide a clear indication of which fields are most important for prediction or classification.
* Ease of use: Decision trees are simple to use and don’t require a lot of technical expertise, making them accessible to a wide range of users.
* Scalability: Decision trees can handle large datasets and can be easily parallelized to improve processing time.
* Missing value tolerance: Decision trees are able to handle missing values in the data, making them a suitable choice for datasets with missing or incomplete data.
* Handling non-linear relationships: Decision trees can handle non-linear relationships between variables, making them a suitable choice for complex datasets.
* Ability to handle imbalanced data: Decision trees can handle imbalanced datasets, where one class is heavily represented compared to the others, by weighting the importance of individual nodes based on the class distribution.

Drawback:

* Decision trees are able to generate understandable rules.
* Decision trees perform classification without requiring much computation.
* Decision trees are able to handle both continuous and categorical variables.
* Decision trees provide a clear indication of which fields are most important for prediction or classification.
* Ease of use: Decision trees are simple to use and don’t require a lot of technical expertise, making them accessible to a wide range of users.
* Scalability: Decision trees can handle large datasets and can be easily parallelized to improve processing time.
* Missing value tolerance: Decision trees are able to handle missing values in the data, making them a suitable choice for datasets with missing or incomplete data.
* Handling non-linear relationships: Decision trees can handle non-linear relationships between variables, making them a suitable choice for complex datasets.
* Ability to handle imbalanced data: Decision trees can handle imbalanced datasets, where one class is heavily represented compared to the others, by weighting the importance of individual nodes based on the class distribution.

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

### Ans: Applications of Decision Trees

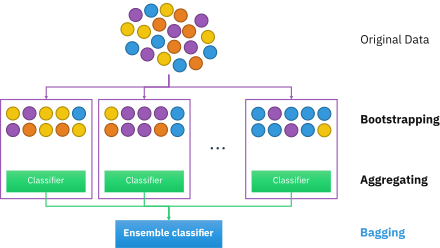
#### 1. Assessing prospective growth opportunities:One of the applications of decision trees involves evaluating prospective growth opportunities for businesses based on historical data. Historical data on sales can be used in decision trees that may lead to making radical changes in the strategy of a business to help aid expansion and growth.

#### 2. Using demographic data to find prospective clients:Another application of decision trees is in the use of [demographic data](https://corporatefinanceinstitute.com/resources/knowledge/economics/demographics/) to find prospective clients. They can help streamline a marketing budget and make informed decisions on the target market that the business is focused on. In the absence of decision trees, the business may spend its marketing market without a specific demographic in mind, which will affect its overall revenues.

#### 3. Serving as a support tool in several fields:Lenders also use decision trees to predict the probability of a customer defaulting on a loan by applying predictive model generation using the client’s past data. The use of a decision tree support tool can help lenders evaluate a customer’s creditworthiness to prevent losses.

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

Ans: Random forest works on the Bagging principle which chooses a random sample/random subset from the entire data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently, which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting, is known as aggregation.



#### Steps Involved in Random Forest Algorithm

Step 1: In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively.

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

Ans: OOB (out-of-bag) score is a performance metric specifically for ensemble models such as random forests( which is 1 – accuracy). It is calculated using the samples that are not used in the training of the model, which is called out-of-bag samples. These samples are used to provide an unbiased estimate of the model’s performance, which is known as the OOB score. OOB (out-of-bag) errors are an estimate of the performance of a random forest classifier or regressor on unseen data. If the OOB error is significantly higher than the validation score, it may indicate that the model is overfitting and not generalizing well to unseen data. On the other hand, if the OOB error is significantly lower than the validation score, it may indicate that the model is underfitting and not learning the underlying patterns in the data.

Using the OOB error as minimization criterion, carry out variable elimination from random forest, by successively eliminating the least important variables (with importance as returned from random forest). [varSelRF](https://rdrr.io/cran/varSelRF/man/varSelRF.html)(xdata, Class, c.sd = 1, mtryFactor = 1, ntree = 5000,

ntreeIterat = 2000, vars.drop.num = [**NULL**](https://rdrr.io/r/base/NULL.html), vars.drop.frac = 0.2,

whole.range = [**TRUE**](https://rdrr.io/r/base/logical.html), recompute.var.imp = [**FALSE**](https://rdrr.io/r/base/logical.html), verbose = [**FALSE**](https://rdrr.io/r/base/logical.html),

returnFirstForest = [**TRUE**](https://rdrr.io/r/base/logical.html), fitted.rf = [**NULL**](https://rdrr.io/r/base/NULL.html), keep.forest = [**FALSE**](https://rdrr.io/r/base/logical.html))