Assignment_2

Name:- Anand Sagar

Roll no. -1754302057

Q.1What is decision tree? Illustrate with an example.

Ans.Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.A decision tree is a very specific type of probability tree that enables you to make a decision about some kind of process. For example, you might want to choose between manufacturing item A or item B, or investing in choice 1, choice 2, or choice 3. Trees are an excellent way to deal with these types of complex decisions, which always involve many different factors and usually involve some degree of uncertainty. Although they can be drawn by hand, software is often used as the trees can become complex very quickly.

example

- Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.
- Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.
- We can represent any boolean function on discrete attributes using the decision tree.

Q2.Write short notes on decision tree construction algorithms:
ID3, C4.5, CART

Ans.**ID3 (Iterative Dichotomiser 3)** was developed in 1986 by Ross Quinlan. The algorithm creates a multiway tree, finding for each node (i.e. in a greedy manner) the categorical feature that will yield the largest information gain for categorical targets. Trees are grown to their maximum size and then a pruning step is usually applied to improve the ability of the tree to generalise to unseen data.

C4.5 is the successor to ID3 and removed the restriction that features must be categorical by dynamically defining a discrete attribute (based on numerical variables) that partitions the continuous attribute value into a discrete set of intervals. C4.5 converts the trained trees (i.e. the output of the ID3 algorithm) into sets of if-then rules. This accuracy of each rule is then evaluated to determine the order in which they should be applied. Pruning is done by removing a rule's precondition if the accuracy of the rule improves without it.

CART

CART (Classification and Regression Trees) is very similar to C4.5, but it differs in that it supports numerical target variables (regression) and does not compute rule sets. CART constructs binary trees using the feature and threshold that yields the largest information gain at each node.

Q.2Explain the basic algorithm for finding association rules.

Association Rule

Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. A typical example is Market Based Analysis.

Market Based Analysis is one of the key techniques used by large relations to show associations between items.It allows retailers to identify relationships between the items that people buy together frequently**.**

The Apriori algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset associations efficiently. It is the iterative process for finding the frequent itemsets from the large dataset.

This algorithm was given by the R. Agrawal and Srikant in the year 1994. It is mainly used for market basket analysis and helps to find those products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients**.**

.

Q.3Mention the top design decisions in architecturing neural networks.

-
- Create a network with hidden layers similar size order to the input, and all the same size, on the grounds that there is no particular reason to vary the size (unless you are creating an autoencoder perhaps).

- Start simple and build up complexity to see what improves a simple network.

- Try varying depths of network if you expect the output to be explained well by the input data, but with a complex relationship (as opposed to just inherently noisy).

- Try adding some dropout, it’s the closest thing neural networks have to magic fairy dust that makes everything better (caveat: adding dropout may improve generalisation, but may also increase required layer sizes and training times).