Name: Akash Omprakash Singh
Branch: CSE (IOT)

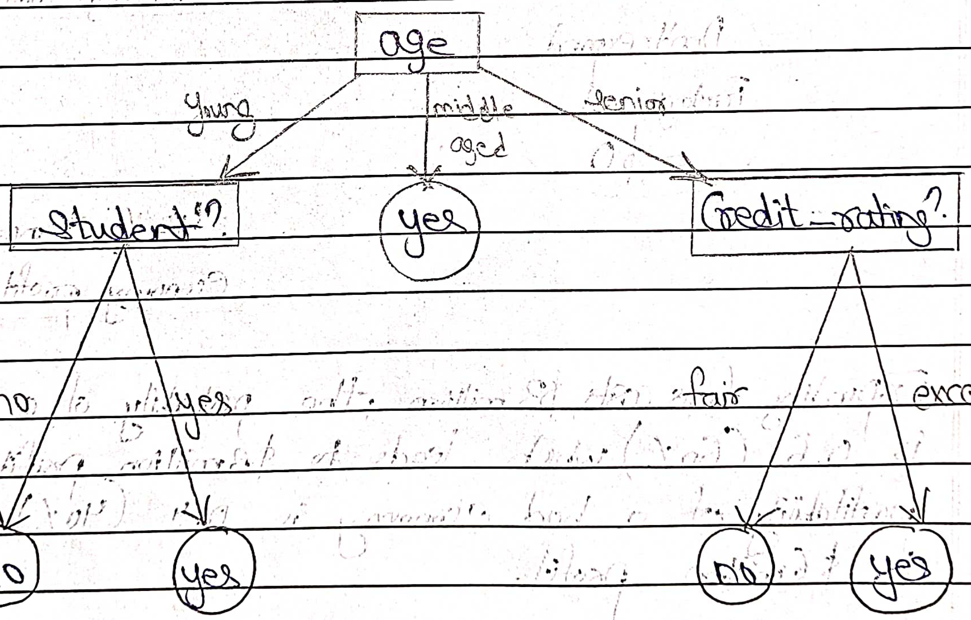
Roll No: 63
Subject: DWM
Date: 13/08/23

Assignment No. 3

Q.1) Explain with one example Decision Tree.

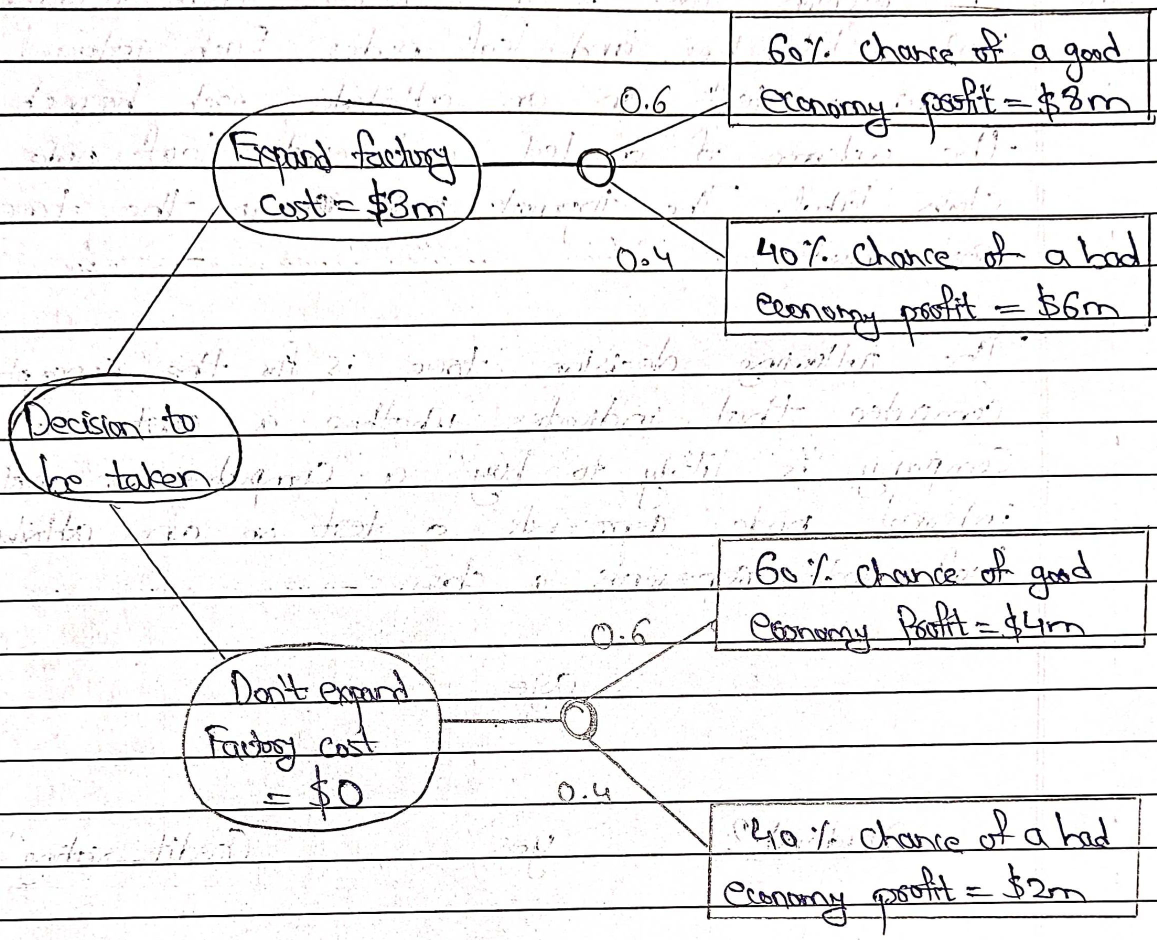
ans. A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy-computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



Date:

Consider the given example of a factory where



Expanding facts costs $3 million, the probabily of a good economy is 0.6 (60%) which leads to $8 million profit, and the profitability of a bad economy is 0.4 (40%) which leads to $6 million profit.

Not expanding factors with 0$ cost, the probability of a good economy is 0.6 (60%), which leads to $4 million profit, and the probability of a bad economy is 0.4, which leads to $2 million profit.

Date:

The management teams need to take a data-driven decision to expand or not based on the given data.
Net expand = $(0.6*8+0.4*6)-3 = \$4.2M$
Net Not expand = $(0.6*4+0.4*2)-0 = \$3M$
$\$4.2M > \$3M$, therefore the factory should be expanded.

The benefits of having a decision tree are as follows-

1) It does not require any domain knowledge.
2) It is easy to comprehend.
3) The learning and classification steps of a decision tree are simple and fast.

(4)

Date:

Q. 2) Explain with one example Naive Bayes Algorithm

ans Naive Bayes is a classification algorithm that's commonly used in data mining and machine learning. It's based on Baye's theorem and assumes that the features used for classification are conditionally independent, which is where the "naive" part comes from. Despite this simplifying assumption, Naive Bayes often performs surprisingly well in various real-world scenarios.

Here's a simple example of how Naive Bayes could be used in spam email classification:

Problem: Classifying emails as either "spam" or "not spam" based on the words contained in the email.

Dataset: Imagine you have a dataset of emails, each labeled as either "spam" or "not spam", and you want to build a model to automatically classify new, unseen mails.

Features: For simplicity, let's consider only two features: the presence or absence of the words "lottery" and "prince".

Date:

Training phase:

1) Data preparation: You calculate the probabilities of each word ("lottery" and "prince") occuring in spam and non-spam emails based on the training data.

2) Calculate Prior Probabilities: You calculate the prior probabilities of an email being spam (P(spam)) and not spam (P(not spam)) based on the training data.

3) Conditional Probabilities: For each word ("lottery" and "prince"), you calculate the conditional probabilities of the word occuring given the class (spam or not spam). For instance, P(lottery | spam) is the probability of seeing the word "lottery" in an email given that it's a spam email.

Classification Phase:

Now, given a new email, you want to classify it as spam or not spam using Naive Bayes.

1) Calculate likelihood: For the new email, you calculate like likelihood of seeing the words "lottery" and "prince" based on the conditional probabilities calculated during the training phase.

SIGCE

Date:

2) <u>Calculate Posterior Probabilities:</u>

Using Baye's theorem, you calculate the posterior probabilities of the email being spam or not spam given the presence of the words "lottery" and "prince".

3). <u>Make a Decision:</u>

You classify the email as spam if the posterior probability of its being spam is higher than the posterior probability of it being not spam.

In this example, the Naive Baye's algorithm makes the classification decision based on the probabilities of word occurrences and the prior probabilities of spam and non-spam emails. Despite the naive assumption of feature independence, Naive Bayes can often work surprisingly well for text classification tasks like spam filtering, sentiment analysis, and more.

* * *