Assignment - 3 Course Code ÷ CAP446

Name → Manish Kumar Choudhary
Regd No. → 12107386
Roll No. → RD2112A39
Section → D2112
Group → 1

Q1. Consider the given dataset and perform Naive Bayesian classification. Mention each step and calculation clearly. (Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = True)

Day	Outlook	Temperature	Humidity	Wind	Play Golf
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rain	Mild	High	False	Yes
5	Rain	Cool	Normal	False	Yes
6	Rain	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rain	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rain	Mild	High	True	No

⇒ Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimesion training dataset. Naive Bayes classifier is one of the simple and most effective classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predict on the basis of the probability on an object. Some popular examples of Naive Bayes algorithm are spam filtration, Sentimental analysis and classifying articles.

Naive Bayes Model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophiscated classification methods.

For example, a fruit may be considered to be an apple if it is red, round and about 3 inches in diameter. Even if these features depend on each other or upon the existance of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as Naive.

Why is it Called Naive Bayes?

The Naive Bayes algorithm is comprised of two words Naive and Bayes, which is described as ÷

⇒ Naive ÷ It is called Naive because it assumes that the occurence of a certain feature is independent of the occurence of other features. Such as if the fruit is identified on the bases of color, shape and taste then red, spherical and sweet fruit is recognized as an apple. Hence, each feature individually contributes to identify that it is an apple without depending on each other.

⇒ Bayes ÷ It is called Bayes because it depends on the principle of Bayes' Theorem.

Assignment - 3 Course Code :- CAP446

In Naive Bayes Algorithm, the example dataset is divided into two parts, namely feature matrix and the response vector.

* Feature matrix contains all the vectors (rows) of dataset in which each vector consists of the value of dependent features.
* Response vector contains the value of class variable (prediction or output) for each row of feature matrix.

Bayes' Theorem :-

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability. Bayes' theorem finds the probability of an event occuring given the probability of another event that has already occured.
Bayes' Theorem finds the probability of an event occuring given the probability of another event that has already occured.
Bayes' theorem is stated mathematically as the following equation :-

$$P(A/B) = \frac{P(B/A)\, P(A)}{P(B)}$$

where, A and B are events and $P(B) \neq 0$

* Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.
* P(A/B) is the posterior probability of class (A, target) given predictor (B, attributes). Probability of hypothesis A on the observed event B.
* P(B/A) is the likelihood which is the probability of predictor given class. Probability of the evidence given that the probability of a hypothesis is true.

* P(A) is the prior probability of class. Probability of hypothesis before observing the evidence.
* P(B) is Marginal probability or prior probability of predictor. Probability of evidence.

Working of Naive Bayes' classifier :-

Suppose, we have a dataset and their corresponding target variable. So using this dataset we need to decide that whether the condition satisfy or not according to the given Conditions.

To perform Naive Bayes' classification, we need to follow the below steps :-

1> Convert the given dataset into frequency tables.
2> Generate Likelihood table by finding the probabilities of given features.
3> Now, use Naive Bayesian equation to calculate the posterior probability for each class.
4> The class with the highest posterior probability is the outcome of prediction

Assignment - 3

Course Code ÷ CAP446

Day	Outlook	Temperature	Humidity	Wind	Play Golf
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rain	Mild	High	False	Yes
5	Rain	Cool	Normal	False	Yes
6	Rain	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rain	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rain	Mild	High	True	No

STEP-1 ÷ First of all, we have to calculate Prior probability of class label attribute.

$$P(\text{Play\_Golf} = \text{yes}) = \frac{9}{14} = 0.64$$

$$P(\text{Play\_Golf} = \text{No}) = \frac{5}{14} = 0.36$$

STEP-2 ÷ Now, we have to calculate Posterior probability of outlook attribute. For this, we have to follow the following steps ÷

Assignment - 3 Course Code :- CAP446

STEP - 2(a) :- Construct a frequency table for outlook attribute against the target attribute.

Frequency Table		Play Golf	
		Yes	No
OUTLOOK	Sunny	2	3
	overcast	4	0
	Rain	3	2

STEP - 2(b) :- Then, transforming the frequency table to likelihood tables and finally use the Naive Bayesian equation to calculate the Posterior probability of outlook attribute.

Likelihood Table		Play Golf		
		Yes	No	
OUTLOOK	Sunny	$2/9 = 0.22$	$3/5 = 0.6$	$5/14 = 0.36$
	Overcast	$4/9 = 0.44$	$0/5 = 0$	$4/14 = 0.29$
	Rain	$3/9 = 0.33$	$2/5 = 0.4$	$5/14 = 0.36$
		$9/14 = 0.64$	$5/14 = 0.36$	

STEP - 3 :- Now, we have to calculate Posterior probability of temperature attribute. For this, we have to follow the following steps :-

Assignment-3 — Course Code ÷ CAP446

STEP-3(a) ÷ Construct a frequency table for temperature attribute against the target attribute.

Frequency Table		Play Golf	
		Yes	No
TEMP	Hot	2	2
	Mild	4	2
	Cool	3	1

STEP-3(b) ÷ Then, transforming the frequency table to likelihood table and finally use the Naive Bayesian equation to calculate the Posterior probability of temperature attribute.

Likelihood Table		Play Golf		
		Yes	No	
TEMP	Hot	$2/9 = 0.22$	$2/5 = 0.4$	$4/14 = 0.29$
	Mild	$4/9 = 0.44$	$2/5 = 0.4$	$6/14 = 0.42$
	Cool	$3/9 = 0.33$	$1/5 = 0.2$	$4/14 = 0.29$
		$9/14 = 0.64$	$5/14 = 0.36$	

STEP-4 ÷ Now, we have to calculate Posterior probability of humidity attribute. For this, we have to follow the following steps ÷

Assignment - 3

Course Code ÷ CAP446

STEP - 4(a) ÷ Construct a frequency table for humidity attribute against the target attribute.

Frequency Table		Play Golf	
		Yes	No
HUMI	High	3	4
	Normal	6	1

STEP - 4(b) ÷ Then, transforming the frequency table to likelihood table and finally use the Naive Bayesian equation to calculate the Posterior probability of humidity attribute.

Likelihood Table		Play Golf		
		Yes	No	
HUMI	High	$3/9 = 0.33$	$4/5 = 0.8$	$7/14 = 0.5$
	Normal	$6/9 = 0.67$	$1/5 = 0.2$	$7/14 = 0.5$
		$9/14 = 0.64$	$5/14 = 0.36$	

STEP - 5 ÷ Now, we have to calculate Posterior probability of wind attribute. For this, we have to follow the following steps ÷

STEP - 5(a) ÷ Construct a frequency table for wind attribute against the target attribute

Assignment-3 Course Code ÷ CAP446

Frequency Table		Play Golf	
		Yes	No
WIND	True	3	3
	False	6	2

STEP-5(b) ÷ Then, transforming the frequency table to likelihood table and finally use the Naive Bayesian equation to calculate the Posterior probability of wind attribute.

Likelihood Table		Play Golf		
		Yes	No	
WIND	True	$3/9 = 0.33$	$3/5 = 0.6$	$6/14 = 0.42$
	False	$6/9 = 0.67$	$2/5 = 0.4$	$8/14 = 0.56$
		$9/14 = 0.64$	$5/14 = 0.36$	

STEP-6 ÷ Now, we have to calculate Posterior probability $P(Yes/x)$ as ÷

$$P(Yes/x) = P(Sunny/yes) * P(Cool/yes) * P(High/yes) * P(True/Yes)$$
$$= 0.22 * 0.33 * 0.33 * 0.33$$
$$= 0.0079$$

STEP-7 ÷ Then, we have to calculate Posterior probability $P(No/x)$ as ÷

$$P(No/x) = P(Sunny/No) * P(Cool/No) * P(High/No) * P(True/No)$$
$$= 0.6 * 0.2 * 0.8 * 0.6$$
$$= 0.0576$$

CLASSMATE
Date :
Page :

Assignment - 3 Course Code ÷ CAP446

STEP- 8 ÷ Now, we have to calculate the final Posterior probability $P(Yes/x) * P(Yes)$ as ÷

$$P(Yes/x) * P(Yes) = 0.0079 * 0.64$$
$$= 0.005$$

STEP- 9 ÷ Then, we have to calculate the final Posterior probability $P(No/x) * P(No)$ as ÷

$$P(No/x) * P(No) = 0.0576 * 0.36$$
$$= 0.206$$

STEP- 10 ÷ Standardized the final Posterior probability between 0 and 1.

$$\frac{P(Yes/x) * P(Yes)}{[P(Yes/x) * P(Yes)] + [P(No/x) * P(No)]} = \frac{0.005}{0.005 + 0.206}$$
$$= \frac{0.005}{0.211}$$
$$= 0.0236$$

$$\frac{P(No/x) * P(No)}{[P(No/x) * P(No)] + [P(Yes/x) + P(Yes)]} = \frac{0.206}{0.206 + 0.005}$$
$$= \frac{0.206}{0.211}$$
$$= 0.9763$$

Since, $0.9763 > 0.0236$, it means new instance is classified as No. The meaning of this is that the person is not going to play the golf.

Assignment - 3 Course Code ÷ CAP446

Q2. Briefly explain Decision trees and its terminologies by taking an example.

⇒ Decision Tree is a supervised learning method used in data mining. They can be used to solve both regression and classification problems. It is a tree that helps us in decision-making purposes. The decision tree creates classification or regression model as a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final tree is a tree 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 test, and each leaf node corresponds to a class label. The topmost node in the tree is the root node. Broadly, we can say that, a decision tree is a tree with the decision nodes and leaf nodes. A decision node has atleast two branches. The leaf nodes show a classification or decision. we can't accomplish more split on leaf-nodes. The uppermost decision node in a tree that relates to the best predictor called the root node. Decision trees can deal with both categorical and numerical data. we can represent any boolean function on discrete attributes using the decision tree. This type of mining belongs to supervised class learning. In supervised learning, the target result is already known. Decision trees can be used for both categorical and numerical data. The categorical data represent gender, marital status, etc. while the numerical data represent age, temperature, etc.

A tree classification algorithm is used to compute a decision tree. Decision trees are easy to understand and modify, and the model developed can be expressed as a set of decision rules. This algorithm scales well, even where there are

Assignment - 3 Course Code ÷ CAP446

varying numbers of training examples and considerable numbers of attributes in large databases.

Decision Tree classification generates the output as a binary tree-like structure, which gives fairly easy interpretation to the marketing people and easy identification of significant variables for the churn management. A Decision Tree model contains rules to predict the target variable. The Tree classification algorithm provides an easy-to-understand description of the underlying distribution of the data.

The intuition is that, by classifying larger datasets, you will be able to improve the accuracy of the classification model. In classification, the given situation is a set of example records, called a training dataset, where each record consists of several field or attributes. Attributes are either numerical (coming from an ordered domain), or categorical (coming from an unordered domain). One of the attributes, called the class label field (target field), indicates the class to which each example belongs. The objective of classification is to build a model of the class label based on the other attributes. After a model is built, it can be used to determine the class label of unclassified records. Application of classification arise in diverse fields, such as retail target marketting, customer retention, fraud detection and medical diagnosis.

Among these models, decision trees are particularly suited for data mining. Decision trees can be constructed relatively quickly, compared to other method. Another advantage is that decision tree models are simple and easy to understand. A decision tree is a class discriminator that recursively partitions the training set until each partition consists entirely or dominantly of examples from one class. Each non-leaf node of the tree contains a split point that is a test on one or more attributes and determines how the data is partitioned. The tree is built by recursively partitioning the data. Partitioning continues until each partition is either 'pure' (all members belong to the same class) or sufficiently small (a parameter set by the user). The initial lists created from the training set are associated with the root of the decision tree. As the tree is grown and nodes are split to create new children, the attribute lists for each node are partitioned and associated with the children.

Overall, We can say that Decision Tree is used to build classification and regression models. It is used to create data models that will predict class labels or values for the decision-making process. The models are built from the training dataset fed to system (supervised learning). Using a decision tree, we can visualize the decisions that make it easy to understand and thus it is a popular data mining technique.

A decision tree classifier is built in two phases ÷

* A growth phase
* A prune phase

After the initial tree has been built (the growth phase), a sub-tree is built with the least estimated error rate (the prune phase). The process of pruning the initial tree consists of removing small, deep nodes of the tree resulting from 'noise' contained in the training data, thus reducing the risk of 'overfitting', and resulting in a more accurate classification of unknown data. While the decision tree is being built, the goal at each node is to determine the split attribute and the split point that best divides the training records belonging to that leaf. The value of a split point depends on how well it separates the classes. Several splitting indices have been proposed in the past to evalute the quality of the split. Intelligent Miner uses the gini index.

Assignment-3 Course Code :- CAP446

Important terms of Decision Tree in Data Mining :-

Here are some of the important terms of a decision tree in data mining given below :-

* **Root Node** :-
  This is the first node where the splitting takes place. Or we can say that, the blue decision is called the 'root node'. This is at all the times the primary node in the path. It is the knot from which all other choices, forecasts and end knots finally divide.

* **Leaf Node** :-
  This is the node after which there is no more branching. Or we can say that, the lavender end nodes are called the 'leaf nodes'. These display the conclusion of a decision route (or outcome). Every time you recognize a leaf node because it doesn't fragment, or subdivide any more like a real leaf.

* **Decision Node** :-
  The node formed after splitting data from a previous node is known as a decision node. Generally, it represents a decision and is normally displayed with a square.

* **Branch** :-
  Subsection of a tree containing information about the aftermath of split at the decision node.

* **Prunning** :-
  When removing a decision node's sub-nodes to cater to an outlier or noisy data is called pruning. It is also thought to be the opposite of splitting. Rarely

Assignment-3 Course Code ÷ CAP446

decision trees can become attractively miscellaneous. In these circumstances, they can close up giving too much load to immaterial information. To sidestep this difficulty, we can eliminate definite nodes using a procedure well known as 'pruning'. Pruning is precisely what it echoes like if the tree develops branches we don't require, we basically cut them off. Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.
There are two approaches to prune a tree ÷
* Pre-pruning ÷ The tree is pruned by halting its construction early.
* Post-pruning ÷ This approach removes a sub-tree from a fully grown tree.

* Internal nodes ÷
In between the origin knots and the leaf knots, we can have any number of internal ties. These can comprise decisions and chance nodes (for ease, this image only uses chance nodes). It is really easy to identify an internal node as each internal nodes have branches of its own while also joining to the earlier node.

* Splitting ÷
Dividing or splitting is said when any node divides two or more substitute nodes. These substitute nodes can also be another internal node, or they can tip to result (a leaf/ end node).

* Chance nodes ÷
It represents chance or confusion and is normally displayed with a circle.

* End nodes ÷
It represents a result and is normally displayed with a triangle.

Assignment - 3 Course Code ÷ CAP446

How to select attributes for creating a tree?

In Decision Tree, the major challenge is to identification of the attribute for the root node in each level. This process is known as attribute selection. Attribute selection measures are also called splitting rules to decide how the tuples are going to split. The splitting criteria are used to best partition the dataset. These measures provide a ranking to the attributes for partitioning the training tuples.

We have three popular attribute selection measures ÷
(i) Information Gain
(ii) Gain Ratio
(iii) Gini Index

⇒ Information Gain ÷
This method is the main method that is used to build decision trees. It reduces the information that is required to classify the tuples. It reduces the number of tests that are needed to classify the given tuple. The attribute with the highest information gain is selected. When we use a node in a decision tree to partition the training instances into smaller subsets the entrophy changes. Information gain is a measure of this change in entrophy.

To calculate the information gain, we use following steps ÷
STEP-1 ÷ Calculate entropy of the target.
STEP-2 ÷ The dataset is then split on the different attributes. The

Assignment - 3 Course Code ÷ CAP446

entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entrophy is subtracted from the entropy before the split. The result is the information Gain, or decrease in entropy.

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

STEP-3 ÷ Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

STEP-4a ÷ A branch with entropy of 0 is a leaf node.

STEP-4b ÷ A branch with entropy more than 0 needs further splitting.

STEP-5 ÷ The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

Entropy ÷

This information gain is also called Entropy. Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content. A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). A decision tree algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

To build a decision tree, we need to calculate two types of entropy using frequency table as follows ÷

Assignment-3 Course Code ÷ CAP446

(a) Entropy using the frequency table of one attribute ÷

$$E(S) = \sum_{i=1}^{c} - P_i \log_2 P_i$$

(b) Entropy using the frequency table of two attributes ÷

$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

Information gain is the difference between the original and expected information that is required to classify the tuples of dataset D.

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

Gain is the reduction of information that is required by knowing the value of X. The attribute with the highest information gain is chosen as "best".

Building Decision Tree using information gain the essentials ÷

* Start with all training instances associated with the root node.
* using info gain to choose which attribute to label each node with.
* Note : No root-to-leaf path should contain the same discrete attribute twice.
* Recursively construct each subtree on the subset of training instances that would be classified down that path in the tree.

Assignment - 3 Course Code ÷ CAP446

The border cases ÷

* If all the positive or negative training instances remain, label that node "yes" or "no" accordingly.
* If no attributes remain, label with a majority vote of training instances left at that node.
* If no instances remain, label with a majority vote of the parent's training instances.

⇒ Gain Ratio ÷

Information gain might sometimes result in portioning useless for classification. However, the Gain ratio splits the training data set into partitions and considers the number of tuples of the outcome with respect to the total tuples. The attribute with the max gain ratio is used as a splitting attribute.

$$\text{Gain Ratio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfo}(D)}$$

⇒ Gini Index ÷

Gini Index is a metric to measure how often a randomly choosen element would be incorrectly identified.

* It means an attribute with lower Gini Index should be preferred.
* SKlearn supports "Gini" criteria for Gini Index and by default, it takes "gini" value.
* The formula for the calculation of the Gini Index is given below.

Assignment-3 Course Code : CAP446

Gini Index is calculated for binary variables only. It measures the impurity in training tuples of Dataset D, as

$$Gini = 1 - \sum_{i} P(i/t)^2$$

P is the probability that tuple belongs to class C. The gini index that is calculated for binary split dataset D by attribute A is given by :

$$Gini_{split} = \sum_{i=1}^{K} \frac{n_i}{n} Gini(i)$$

where n is the n^{th} partition of the dataset D.
The reduction in impurity is given by the difference of the gini index of the original dataset D and gini index after partition by attribute A.
The maximum reduction in impurity or max gini index is selected as the best attribute for splitting

Assignment - 3 Course Code : CAP446

Example :

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
Young	Yes	Software	Up
Young	No	Hardware	Up
Young	No	Software	Up

$$\text{Information Gain} = \frac{-P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

$$\text{Entropy}(A) = \sum_{i=1}^{V} \frac{P_i + N_i}{P+N} * I(P_i N_i)$$

Here, In this dataset, Profit is our class attribute

$$\text{Information Gain (IG)} = -\left(\frac{5}{10}\log_2\left(\frac{5}{10}\right) + \frac{5}{10}\log_2\left(\frac{5}{10}\right)\right)$$

$$= -\left(0.5 * \log_2 2^{-1} + \frac{5}{10}\log_2 2^{-1}\right)$$

$$= -\left(0.5 * (-1.\log_2 2) + 0.5 * (-1.\log_2 2)\right)$$

Assignment-3 Course Code ÷ CAP446

$$= -(-0.5 - 0.5)$$
$$= -(-1)$$
$$\boxed{= 1}$$

Calculate the Entropy of Age attribute ÷

Age	Down	up
old	3	0
Mid	2	2
New	0	3

$$IG(old) = -\left(\frac{3}{3}\log_2\left(\frac{3}{3}\right) + \frac{0}{3}\log_2\left(\frac{0}{3}\right)\right) = 0$$

$$E(old) = 0 * \frac{3}{10} = 0$$

$$IG(Mid) = -\left(\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right) = 1$$

$$E(Mid) = 1 * \frac{4}{10} = 0.4$$

$$IG(young) = -\left(\frac{0}{3}\log_2\left(\frac{0}{3}\right) + \frac{3}{3}\log_2\left(\frac{3}{3}\right)\right) = 0$$

$$E(young) = 0 * \frac{3}{10} = 0$$

$$E(Age) = E(old) + E(Mid) + E(young)$$
$$= 0 + 0.4 + 0$$
$$= 0.4$$

Assignment-3 Course Code : CAP446

Gain = Information Gain (Profit) - Entropy (Age)
= 1 - 0.4
$\boxed{= 0.6}$

Calculate the Entropy of Competition attribute ÷

Competition	Down	up
Yes	3	1
No	2	4

$$IG(Yes) = -\left(\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = 0.81$$

$$E(yes) = 0.81 * \frac{4}{10} = 0.32$$

$$IG(No) = -\left(\frac{2}{6}\log_2\left(\frac{2}{6}\right) + \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right) = 0.91$$

$$E(No) = 0.91 * \frac{6}{10} = 0.54$$

E (Competition) = E (yes) + E(No)
= 0.32 + 0.54
= 0.86

Gain = Information Gain (Profit) - Entropy (Competition)
= 1 - 0.86
$\boxed{= 0.14}$

Calculate the Entropy of type attribute :-

Type	Down	up
Software	3	3
Hardware	2	2

$$IG(Software) = -\left(\frac{3}{6}\log_2\left(\frac{3}{6}\right) + \frac{3}{6}\log_2\left(\frac{3}{6}\right)\right) = 1$$

$$E(Software) = 1 * \frac{6}{10} = 0.6$$

$$IG(Hardware) = -\left(\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right) = 1$$

$$E(Hardware) = 1 * \frac{4}{10} = 0.4$$

$$E(Type) = E(Software) + E(Hardware)$$
$$= 0.6 + 0.4$$
$$= 1$$

Gain = Information Gain (Profit) - Entropy (competition)

$$= 1 - 1$$
$$\boxed{= 0}$$

Assignment - 3

Course Code : CAP446

Gain (Age) = 0.6
Gain (Competition) = 0.14
Gain (Type) = 0