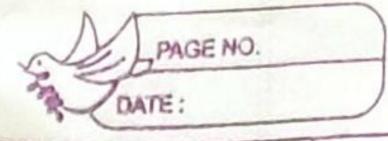
Chaitanya m. Nawale PAGE NO. BECOB215 MI Assignment No. 4 * Title :- To implement decision tree classifier technique. Objective: - To learn how to apply decision tree algorithm for given dataset using SK-Learn. * Outcomes: - Ability to apply decision tree classification technique to solve the given problem. * Software requirements ?- python3, jupyter notebook, scikit-learn. Problem Statement of Adataset collected in a cosmetics shop showing details of customers and whether or not they responded to a special offer to buy a new lip-stick is shown in table below. Use this dataset to build a decision tree, with buys as the target variables to help in buying lip-sticks in the futher future Find the root node of decision tree. According to the decision tree you have made from previous trainging dataset, what is the decision for the test data: [Age < 21, Income= Low, Gender = Female, Marital Status = Married] ?

BEC	0821	5
7	00000	Ger

9	1	and
00%	E-PAGE NO.	
1	JUNE:	
	The second of the Park Control of the San State of the Sa	-

ID	Age	Income	Gender	Marital Status	Buys
1	<21	High	Male	Single	No
2	<21	High	Male		No
3	21-35	High	Male	Single	Yes
4	>35	Medium	Male	Single	Yes
5	>35	Low		Single	Yes
6	>35	Low		Married	No
7	21-35	Low		Married	Yes
8	C21	Medium	Male	Single	No
9	(21	Low		Married	Yes
10	>35	TO THE REAL PROPERTY OF THE PERSON OF THE PE			Yes
H	2211	Medium			Yes
12	21-35	Medium		Married	Yes
13	21-35	High	Female	Single	Yes
14	>35	Medium	Male	Married	No
	- plinted	mai my la m	rule soil	tamana 1	

Theory: - Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represents the decision rules or conditions It is mostly used in Machine Learning and Data Mining applications using Python.
As you might intuit from the name, decision tree learners, build a model in the form of a tree structure. The model itself comprises a series of logical decisions similar to a flowchart, with decision notes that indicate a decision to be made an an attribute. These split into branches that Indicate the decision's choices. The tree



is terminated by leaf nodes Calso known as terminal nodes) that denote the result of following combination of decisions.

· Binary Decision Trees 3

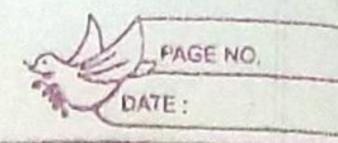
A binary decision tree is a structure based on a sequencial decision process. Starting from the root, a feature is evaluated and one Of the two branches is selected. This procedure is repeated until a final leaf is reached, which normally represents the classification target we're looking for. Considering other algorithms, decision tree seems to be simpler in their dynamics; however, if the dataset is splittable while keeping an internal balance, the overall process is intuitive and rather fast in its predictions. Moreover, decision trees can work efficiently with unnormalized datasets because their internal structure is not influenced by the values assumed by each feature

Binary decisions Let's consider an input dataset X:

X= { x, x, ..., xn} where x; ERM

Forery vector is made up of m features,
so each of them can be a good candidate
to create a node based on the (feature,
threshold) tuple:
According to the feature and the threshold

BEC08215 17



The structure of the tree will change.

Intuitively we should pick the feature that best separates our data in other words, a perfect separating feature will be present on in a node and the two subsequent branches won't be based on it anymore. In real problems, this is often impossible, so it's necessary to find the feature that minimizes the number of decision steps.

·Impurity measures:

To define the most used impurity measures we need to consider the total number of target classes!

Jn a certain node j, we can define
the probability p(ilj) where i is an inderfine
associated with each class. In other words,
according to a frequentist approach, this
value is the ratio between the number
of samples belonging to class i and the
total number of samples belonging to the
selected node

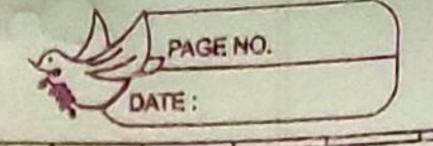
- Decision tree classification with scikit-learn:

 Spliting dataset into train and test datasets

 >>> X_train, X_test, y_train, y-test= train_test_

 split (X, Y, test_size = 0.3, random_state=100)
- 1) Décision Tree Classifier with criterion Gini Inderi

BECOB215



>>> clf-gini = Decision Tree Classifier (criterion= "gini", random state = 100, max depth=3, mean min-samples leaf = 5) Gini Index prediction on test dataset. >>> y-pred = clf_gini.predict (x_test) Decision Tree model with Gini Index accuracy: >>> print "Accuracy is", accuracy-score Cy-test, 4-pred) * 100 2) Decision Tree Classifier with criterion information gain: >>> df_entropy = Decision Tree Classifier (
criterion = "entropy", random_state = 100, max

-depth = 3, min_samples_leaf = S) >>> clf_entropy.fit(x-train, y train) Information Gain Decision tree prediction on test >>> upred_en=clf_entropy.prediction (X_test) Decision Tree classifier with Information gain >>> print "Accuracy is", accuracy_score(y-test,y-pred_en) /a * Canclusion & Thus we have successfully implemented decision tree algorithm for given dutasets using &k-Learn.