LAB Manual

Part A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.05**

**A.1 Aim:**

Write Python function to implement ID3 Decision Tree Classifier using Pre and Post Pruning.

**A.2 Prerequisite:**

Python programming and understanding the working of classification Algorithm

**A.3 Outcome:**

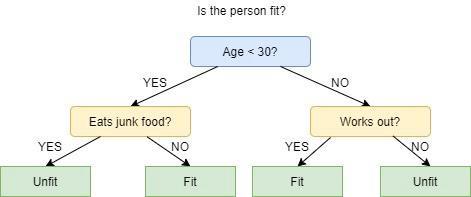
**After successful completion of this experiment students will be able to**

1. To understand the concept of Data Mining by implementing some data mining algorithm.
2. To understand the ID3 Classifier.
3. To understand when to apply pre pruning and Post pruning.
4. To understand hyper parameters for Decision Tree Classifier

**A.4 Theory**

In simple words, a decision tree is a structure that contains nodes (rectangular boxes) and edges(arrows) and is built from a dataset (table of columns representing features/attributes and rows corresponds to records). Each node is used either to make a decision (known as decision node) or represent an outcome (known as leaf node).

Decision tree Example



The picture above depicts a decision tree that is used to classify whether a person is Fit or Unfit.

The decision nodes here are questions like ‘’‘Is the person less than 30 years of age?’, ‘Does the person eat junk?’, etc. and the leaves are one of the two possible outcomes viz. Fit and Unfit.

Looking at the Decision Tree we can say make the following decisions:

if a person is less than 30 years of age and doesn’t eat junk food then he is Fit, if a person is less than 30 years of age and eats junk food then he is Unfit and so on.

The initial node is called the root node (colored in blue), the final nodes are called the leaf nodes (colored in green) and the rest of the nodes are called intermediate or internal nodes.

The root and intermediate nodes represent the decisions while the leaf nodes represent the outcomes.

As mentioned previously, the ID3 algorithm selects the best feature at each step while building a Decision tree.

Before you ask, the answer to the question: ‘How does ID3 select the best feature?’ is that ID3 uses Information Gain or just Gain to find the best feature.

Information Gain calculates the reduction in the entropy and measures how well a given feature separates or classifies the target classes. The feature with the highest Information Gain is selected as the best one.

In simple words, Entropy is the measure of disorder and the Entropy of a dataset is the measure of disorder in the target feature of the dataset.

In the case of binary classification (where the target column has only two types of classes) entropy is 0 if all values in the target column are homogenous(similar) and will be 1 if the target column has equal number values for both the classes.

We denote our dataset as S, entropy is calculated as:

Entropy(S) = - ∑ pᵢ \* log₂(pᵢ) ; i = 1 to n

where,

n is the total number of classes in the target column (in our case n = 2 i.e YES and NO)

pᵢ is the probability of class ‘i’ or the ratio of “number of rows with class i in the target column” to the “total number of rows” in the dataset.

Information Gain for a feature column A is calculated as:

IG(S, A) = Entropy(S) - ∑((|Sᵥ| / |S|) \* Entropy(Sᵥ))

where Sᵥ is the set of rows in S for which the feature column A has value v, |Sᵥ| is the number of rows in Sᵥ and likewise |S| is the number of rows in S.

**ID3 Steps**

1. Calculate the Information Gain of each feature.
2. Considering that all rows don’t belong to the same class, split the dataset S into subsets using the feature for which the Information Gain is maximum.
3. Make a decision tree node using the feature with the maximum Information gain.
4. If all rows belong to the same class, make the current node as a leaf node with the class as its label.
5. Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

**PART B**

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

|  |  |
| --- | --- |
| Roll No.: C026 | Name: Anirbaan Ghatak |
| Class: B | Batch: EB1 |
| Date of Experiment: 5/09/2023 | Date of Submission |
| Grade : |  |

**B. ID3 Code written by student:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.tree import plot\_tree

# Converting categorical variables into dummies/indicator variables

df = pd.read\_csv('playgolf\_data.csv')

df\_getdummy=pd.get\_dummies(data=df, columns=['Temperature', 'Humidity', 'Outlook', 'Wind'])

y = df\_getdummy['PlayGolf']

X = df\_getdummy.drop('PlayGolf',axis=1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=101)

# importing Decision Tree Classifier via sklean

dtree = DecisionTreeClassifier(criterion='entropy',max\_depth=2)

dtree.fit(X\_train,y\_train)

predictions = dtree.predict(X\_test)

# visualising the decision tree diagram

import matplotlib.pyplot as plt

fig = plt.figure(figsize=(16,12))

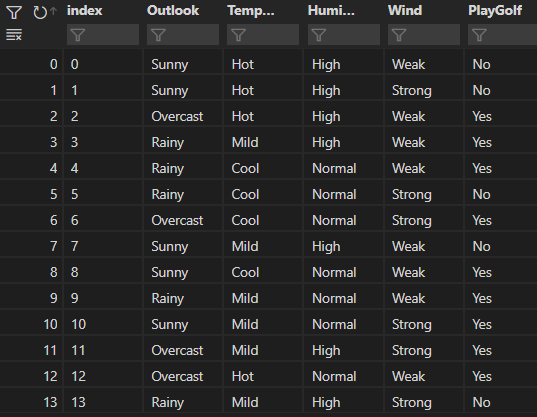
a = plot\_tree(dtree, feature\_names=df\_getdummy.columns, fontsize=12, filled=True,

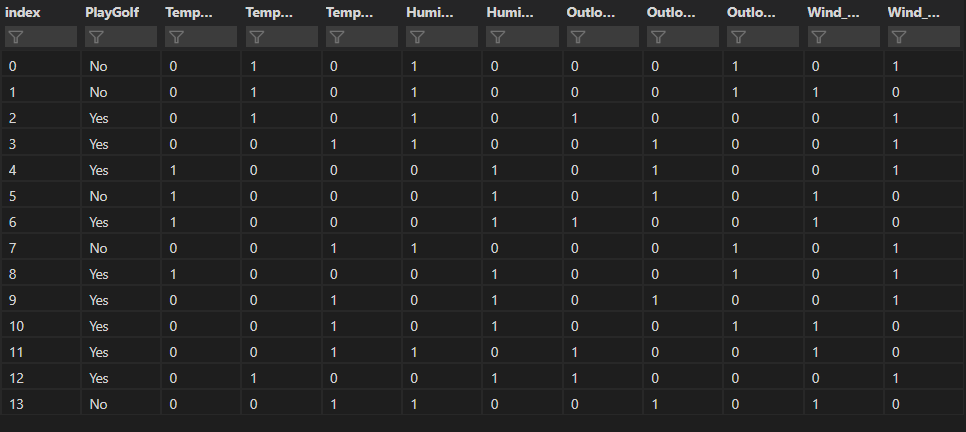
              class\_names=['Not\_Play', 'Play'])

**B.2 Input and Output:**

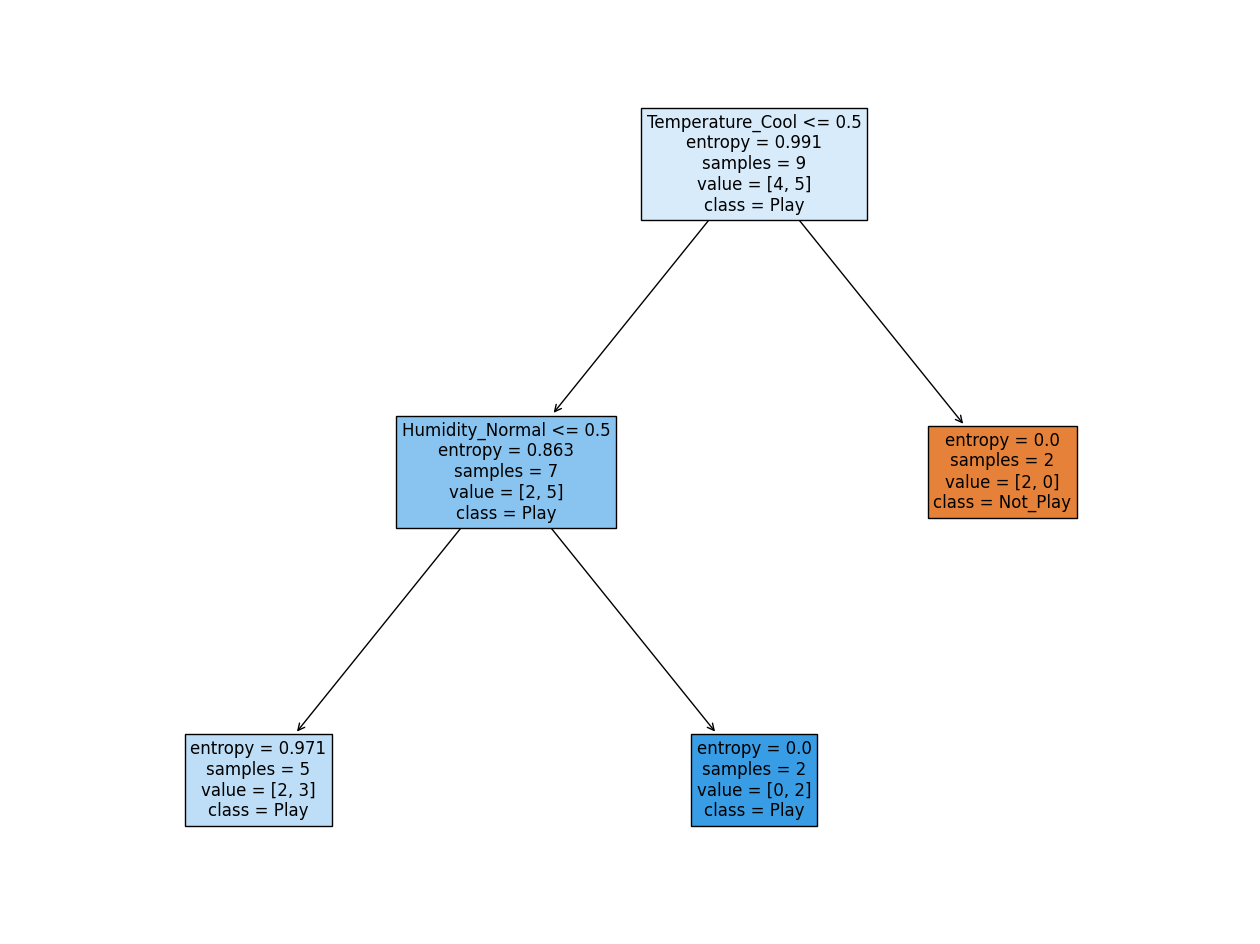
***(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)***

**Input Data:**





**Output:**



**B.3 Observations and learning:**

*Understood the concept of the ID3 Classifier and when to apply pre pruning and Post pruning along with the introduction to the concept hyper parameters for Decision Tree Classifier*

**B.4 Conclusion:**

*Implemented ID3 decision algorithm on selected dataset*