S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 3

Aim: Apply the knowledge to study and implement the Decision Tree learning.

Name of Student : Shrutika Pradeep Bagdi

Roll No. : CS22130

Semester/Year : 6^{th}/ 3^{rd}

Academic Session : 2024-2025

Date of Performance :

Date of Submission :

AIM: Apply the knowledge to study and implement Decision Tree Learning.

OBJECTIVE/EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- It allows an individual or organization to weigh possible actions against one another based on their costs, probabilities, and benefits.
- A decision tree is a map of the possible outcomes of a series of related choices.
- A decision model provides a way to visualize the sequences of events that can occur following alternative decisions (or actions) in a logical framework.
- An objective decision is one which is not influenced by one's personal feelings, perspectives, interests and biases.
- A decision tree is a map of the possible outcomes of a series of related choices. It allows an individual or organization to weigh possible actions against one another based on their costs, probabilities, and benefits.

THEORY:

DECISION TREE:-

A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered. The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization.

The decision tree may not always provide a clear-cut answer or decision. Instead, it may present options so the data scientist can make an informed decision on their own. Decision trees imitate human thinking, so it's generally easy for data scientists to understand and interpret the results.

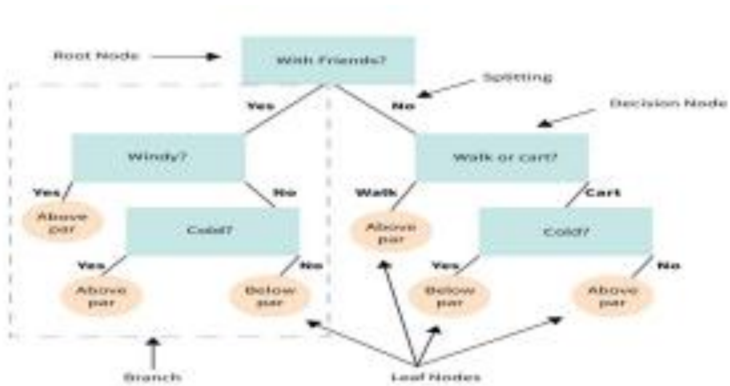
How Does the Decision Tree Work?

Some key terms of a decision tree are as follows:.

- Root node: The base of the decision tree.

- Splitting: The process of dividing a node into multiple sub-nodes.
- Decision node: When a sub-node is further split into additional sub-nodes.
- Leaf node: When a sub-node does not further split into additional sub-nodes; represents possible outcomes.
- Pruning: The process of removing sub-nodes of a decision tree.
- Branch: A subsection of the decision tree consisting of multiple nodes.

A decision tree resembles, well, a tree. The base of the tree is the root node. From the root node flows a series of decision nodes that depict decisions to be made. From the decision nodes are leaf nodes that represent the consequences of those decisions. Each decision node represents a question or split point, and the leaf nodes that stem from a decision node represent the possible answers. Leaf nodes sprout from decision nodes similar to how a leaf sprouts on a tree branch. This is why we call each subsection of a decision tree a "branch." Let's take a look at an example for this. You're a golfer, and a consistent one at that. On any given day you want to predict where your score will be in two buckets: below par or over par.



There are two main types of decision treesExternal link:
Categorical Variable Decision Tree

In a categorical variable decision tree, the answer neatly fits into one category or another. Was the coin toss heads or tails? Is the animal a reptile or mammal? In this type of decision tree, data is placed into a single category based on the decisions at the nodes throughout the tree.

Continuous Variable Decision Tree

A continuous variable decision tree is one where there is not a simple yes or no answer. It's also known as a regression tree because the decision or outcome variable depends on other decisions farther up the tree or the type of choice involved in the decision. The benefit of a continuous variable decision tree is that the outcome can be predicted based on multiple variables rather than on a single variable as in a categorical variable decision tree. Continuous variable decision trees are used to create predictions. The system can be used for both linear and non-linear relationships if the correct algorithm is selected.

The algorithm selection is also based on the type of target variables. Let us look at some algorithms used in Decision Trees:

ID3 → (extension of D3)
C4.5 → (successor of ID3)
CART → (Classification And Regression Tree)
CHAID → (Chi-square automatic interaction detection Performs multi-level splits when computing classification trees)
MARS → (multivariate adaptive regression splines)

The ID3 algorithm builds decision trees using a top-down greedy search approach through the space of possible branches with no backtracking. A greedy algorithm, as the name suggests, always makes the choice that seems to be the best at that moment.

Steps in ID3 algorithm:

1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates Entropy(H) and Information gain(IG) of this attribute. 3. It then selects the attribute which has the smallest Entropy or Largest Information gain. 4. The set S is then split by the selected attribute to produce a subset of the data. 5. The algorithm continues to recur on each subset, considering only attributes never selected before.

For solving this attribute selection problem, researchers worked and devised some solutions. They suggested using some criteria like :

Entropy,

Information gain,

Gini index,

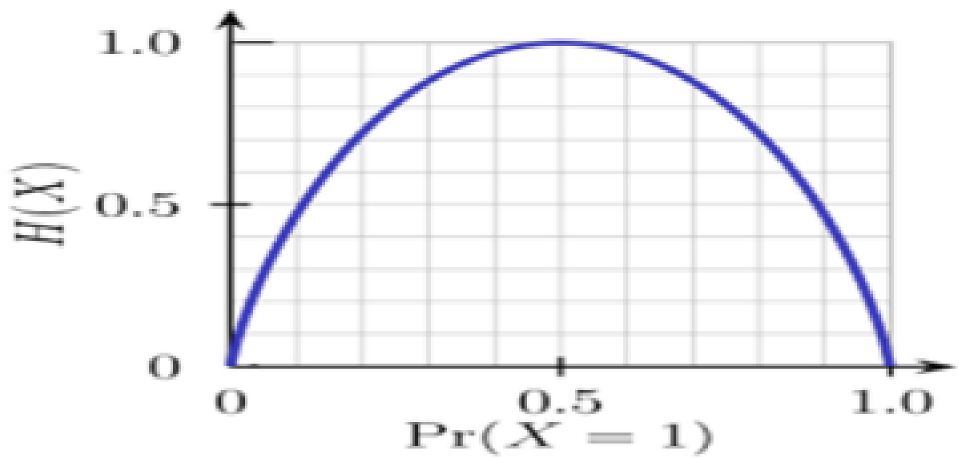
Gain Ratio,

Reduction in Variance

Chi-Square

Entropy

Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information. Flipping a coin is an example of an action that provides information that is random.



ID3 follows the rule — A branch with an entropy of zero is a leaf node and A brach with entropy more than zero needs further splitting.

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

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

Mathematically Entropy for 1 attribute is represented as:

Gini Index

You can understand the Gini index as a cost function used to evaluate splits in the dataset. It is calculated by subtracting the sum of the squared probabilities of each class from one. It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

we have a dataset that has users and their movie genre preferences based on variables like gender, group of age, rating, blah, blah. With the help of information gain, you split at 'Gender' (assuming it has the highest information gain) and now the variables 'Group of Age' and 'Rating' could be equally important and with the help of gain ratio, it will penalize a variable with more distinct values which will help us decide the split at the next level.

$$Gain\ Ratio = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy\ (before) - \sum_{j=1}^{K} Entropy(j,\ after)}{\sum_{j=1}^{K} w_j \log_2 w_j}$$

PROGRAM CODE:

OUTPUT (SCREENSHOT):

Practical 3 (ML).ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
#Shrutika Pradeep Bagdi (CS22130)
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
#Shrutika Pradeep Bagdi (CS22130)
import warnings
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
#Shrutika Pradeep Bagdi (CS22130)
path="/content/daily_weather (1).csv"
df=pd.read_csv(path)
df.head()
```

	number	air_pressure_9am	air_temp_9am	avg_wind_direction_9am	avg_wind_speed_9am	max_wind_direction_9am	max_wind_speed_9am	rain_accumulation_9am	rain_duration_9am	relative_hur
0	0	918.060000	74.822000	271.100000	2.080354	295.400000	2.863283	0.0	0.0	
1	1	917.347688	71.403843	101.935179	2.443009	140.471548	3.533324	0.0	0.0	
2	2	923.040000	60.638000	51.000000	17.067852	63.700000	22.100967	0.0	20.0	
3	3	920.502751	70.138895	198.832133	4.337363	211.203341	5.190045	0.0	0.0	
4	4	921.160000	44.294000	277.800000	1.856660	136.500000	2.863283	8.9	14730.0	

Practical 3 (ML).ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
#Shrutika Pradeep Bagdi (CS22130)
df.isnull().sum()
```

	0
number	0
air_pressure_9am	3
air_temp_9am	5
avg_wind_direction_9am	4
avg_wind_speed_9am	3
max_wind_direction_9am	3
max_wind_speed_9am	4
rain_accumulation_9am	6
rain_duration_9am	3
relative_humidity_9am	0
relative_humidity_3pm	0

dtype: int64

```
#Shrutika Pradeep Bagdi (CS22130)
df.shape
```

```
(1095, 11)
```

Practical 3 (ML).ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
#Shrutika Pradeep Bagdi (CS22130)
df=df.dropna()
```

```
#Shrutika Pradeep Bagdi (CS22130)
df.shape
```

```
(1064, 11)
```

```
#Shrutika Pradeep Bagdi (CS22130)
df['relative_humidity_3pm'] = df['relative_humidity_3pm'].apply(lambda x: 1 if x > 24.99 else 0)
```

```
#Shrutika Pradeep Bagdi (CS22130)
df.head()
```

	number	air_pressure_9am	air_temp_9am	avg_wind_direction_9am	avg_wind_speed_9am	max_wind_direction_9am	max_wind_speed_9am	rain_accumulation_9am	rain_duration_9am	relative_hur
0	0	918.060000	74.822000	271.100000	2.080354	295.400000	2.863283	0.0	0.0	
1	1	917.347688	71.403843	101.935179	2.443009	140.471548	3.533324	0.0	0.0	
2	2	923.040000	60.638000	51.000000	17.067852	63.700000	22.100967	0.0	20.0	
3	3	920.502751	70.138895	198.832133	4.337363	211.203341	5.190045	0.0	0.0	
4	4	921.160000	44.294000	277.800000	1.856660	136.500000	2.863283	8.9	14730.0	

Practical 3 (ML).ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
#Shrutika Pradeep Bagdi (CS22130)
df = df.drop(columns=['number','relative_humidity_9am'])
```

```
#Shrutika Pradeep Bagdi (CS22130)
df.head()
```

	air_pressure_9am	air_temp_9am	avg_wind_direction_9am	avg_wind_speed_9am	max_wind_direction_9am	max_wind_speed_9am	rain_accumulation_9am	rain_duration_9am	relative_humidity_3
0	918.060000	74.822000	271.100000	2.080354	295.400000	2.863283	0.0	0.0	
1	917.347688	71.403843	101.935179	2.443009	140.471548	3.533324	0.0	0.0	
2	923.040000	60.638000	51.000000	17.067852	63.700000	22.100967	0.0	20.0	
3	920.502751	70.138895	198.832133	4.337363	211.203341	5.190045	0.0	0.0	
4	921.160000	44.294000	277.800000	1.856660	136.500000	2.863283	8.9	14730.0	

```
#Shrutika Pradeep Bagdi (CS22130)
X=df.iloc[:,:-1]
print(X)
```

```
      air_pressure_9am  air_temp_9am  avg_wind_direction_9am  \
0           918.060000     74.822000              271.100000
1           917.347688     71.403843              101.935179
2           923.040000     60.638000               51.000000
3           920.502751     70.138895              198.832133
4           921.160000     44.294000              277.800000
...                ...           ...                     ...
1090        918.900000     63.104000              192.900000
1091        918.710000     49.568000              241.600000
```

Practical 3 (ML).ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
#Shrutika Pradeep Bagdi (CS22130)
y=df.iloc[:,-1]
print(y)
```

```
0       1
1       0
2       0
3       0
4       1
       ..
1090    1
1091    1
1092    1
1093    1
1094    0
Name: relative_humidity_3pm, Length: 1064, dtype: int64
```

```
#Shrutika Pradeep Bagdi (CS22130)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, test_size = 0.25, random_state = 42)
```

```
#Shrutika Pradeep Bagdi (CS22130)
from sklearn.tree import DecisionTreeClassifier # Import DecisionTreeClassifier
humidity_classifier= DecisionTreeClassifier(max_leaf_nodes=10,random_state=0)
humidity_classifier.fit(X_train,y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(max_leaf_nodes=10, random_state=0)
```

+ Code + Text

```
#Shrutika Pradeep Bagdi (CS22130)
y_pred = humidity_classifier.predict(X_test)
```

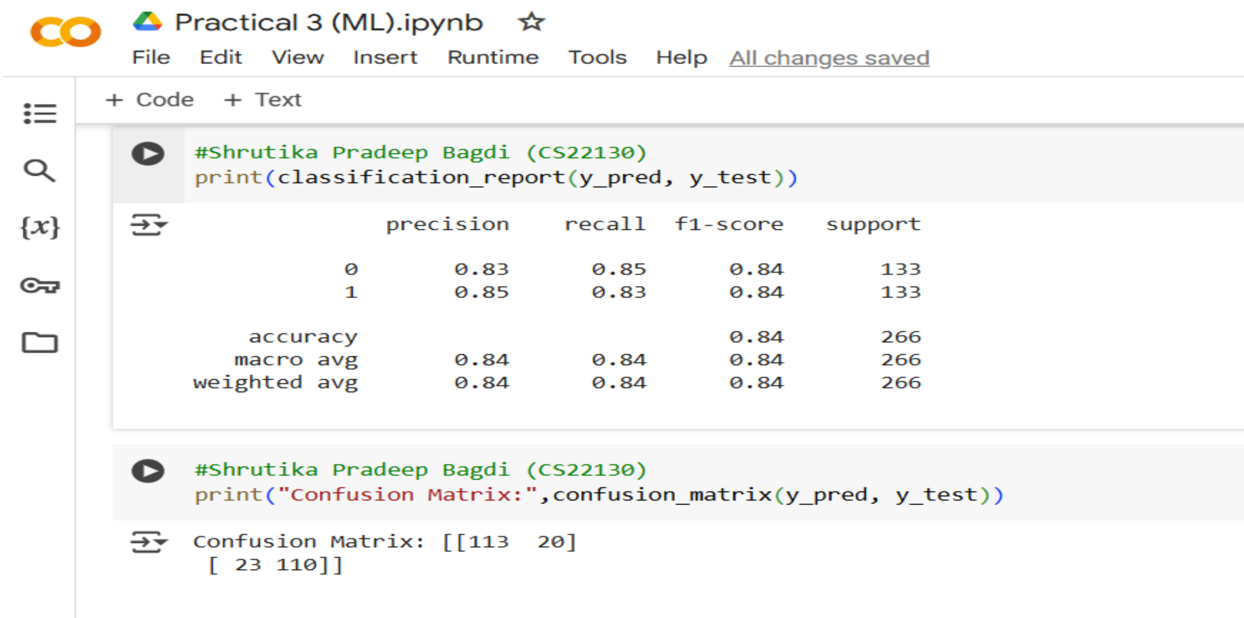
```
#Shrutika Pradeep Bagdi (CS22130)
print(y_pred)
```

```
[1 0 0 1 0 1 1 1 0 1 1 1 0 1 0 0 1 1 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1
 0 1 0 1 1 1 1 1 1 0 0 1 1 1 0 1 1 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 1 0 1 1 1
 1 1 0 0 1 1 1 0 1 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 0
 1 1 1 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0 0 1 1 1 0 0 1 0 0 1 1 1 0
 1 0 0 0 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 1 0 0 0
 0 1 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 0 0 1 0
 0 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 1
 0 1 1 0 1 0 1]
```

```
#Shrutika Pradeep Bagdi (CS22130)
from sklearn.metrics import accuracy_score, classification_report,confusion_matrix
```

```
#Shrutika Pradeep Bagdi (CS22130)
score = accuracy_score(y_pred, y_test)
print(score)
```

```
0.8383458646616542
```


Practical 3 (ML).ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
#Shrutika Pradeep Bagdi (CS22130)
print(classification_report(y_pred, y_test))
```

```
              precision    recall  f1-score   support

           0       0.83      0.85      0.84       133
           1       0.85      0.83      0.84       133

    accuracy                           0.84       266
   macro avg       0.84      0.84      0.84       266
weighted avg       0.84      0.84      0.84       266
```

```
#Shrutika Pradeep Bagdi (CS22130)
print("Confusion Matrix:",confusion_matrix(y_pred, y_test))
```

```
Confusion Matrix: [[113  20]
 [ 23 110]]
```

CONCLUSION:

DISCUSSION AND VIVA VOCE:

- What is decision tree in machine learning interview questions?
- What are the issues in decision tree in machine learning?
- What is entropy?
- What is information gain?
- How are entropy and information gain related decision trees?
- How do you calculate the entropy of children nodes after the split based on a feature?

REFERENCE

- https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision tree/#:~:text=A%20decision%20tree%20is%20a,that%20contains%20the%20desired%20categorization.
- https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
- https://www.google.com/search?q=viva+questions+on+decision+tree+in+machine+learning&oq=viva+questions+on+decision+tree+in+m&aqs=chrome.1.69i57j33i160j33i22i29i30l3.16746j0j7&sourceid=chrome&ie=UTF-8