**Div: B**
**Roll No.: 422004**
**Gr. No.: 21810939**
**Name of Student: Aakanksha Bhondve**
**subject: Advanced Machine Learning**

**Assignment No. 1**

**Title:** Write a program to do : A dataset 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 variable, to help in buying lip-sticks in the future. Find the root node of decision tree. According to the decision tree you have made from previous training data set, what is the decision for the test data: [Age < 21, Income = Low, Gender = Female, Marital Status = Married]?

| ID | Age | Income | Gender | Marital Status | Buys |
|---|---|---|---|---|---|
| 1 | < 21 | High | Male | Single | No |
| 2 | < 21 | High | Male | Married | No |
| 3 | 21-35 | High | Male | Single | Yes |
| 4 | >35 | Medium | Male | Single | Yes |
| 5 | >35 | Low | Female | Single | Yes |
| 6 | >35 | Low | Female | Married | No |
| 7 | 21-35 | Low | Female | Married | Yes |
| 8 | < 21 | Medium | Male | Single | No |
| 9 | <21 | Low | Female | Married | Yes |
| 10 | > 35 | Medium | Female | Single | Yes |
| 11 | < 21 | Medium | Female | Married | Yes |
| 12 | 21-35 | Medium | Male | Married | Yes |
| 13 | 21-35 | High | Female | Single | Yes |
| 14 | > 35 | Medium | Male | Married | No |

# Theory:

## Machine Learning:

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

## Types of Machine Learning Algorithms:

1. **Supervised Learning**

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not. The goal of supervised learning is to map input data with the output data.

Supervised learning can be grouped further in two categories of algorithms:

- Classification
- Regression

2. **Unsupervised Learning**

   Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data.

   It can be further classified into two categories of algorithms:

   - Clustering
   - Association

3. **Reinforcement Learning**

   Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance. Supervised Machine Learning: Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y). Supervised learning can be further divided into two types of problems: 1. Regression Regression algorithms are used if there is a relationship between the input variable and the output variable. It is

used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc.

Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

**2. Classification**

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Spam Filtering,
- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

**Decision Tree Working**

- **Step 1:**

Begin the tree with the root node, says S, which contains the complete dataset.

- **Step 2:**

Find the best attribute in the dataset using Attribute Selection Measure (ASM).

- **Step 3:**

Divide the S into subsets that contain possible values for the best attributes.

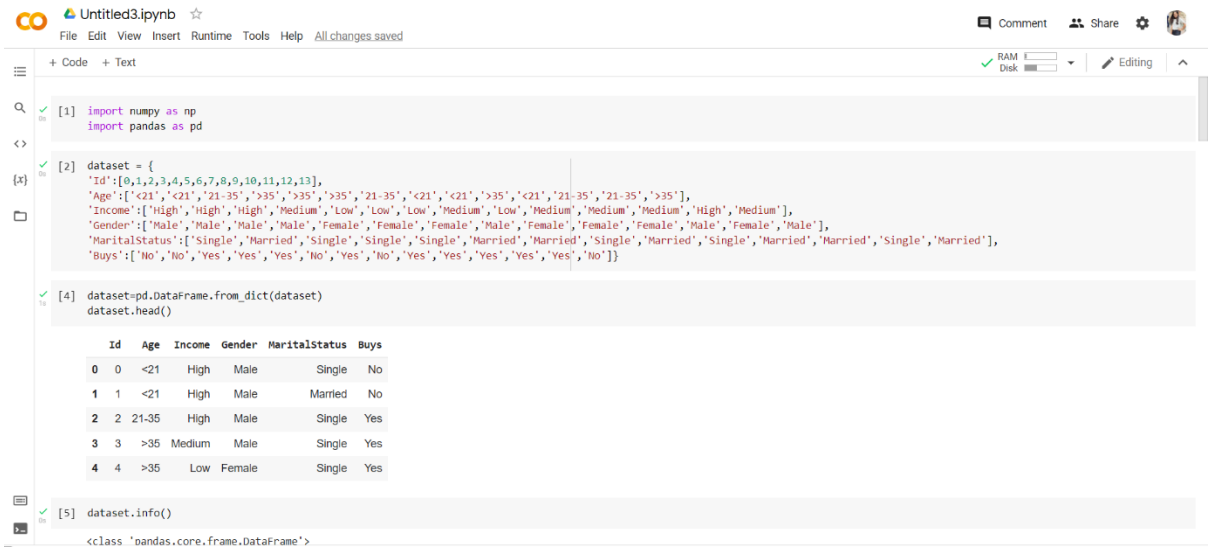
- **Step 4:**

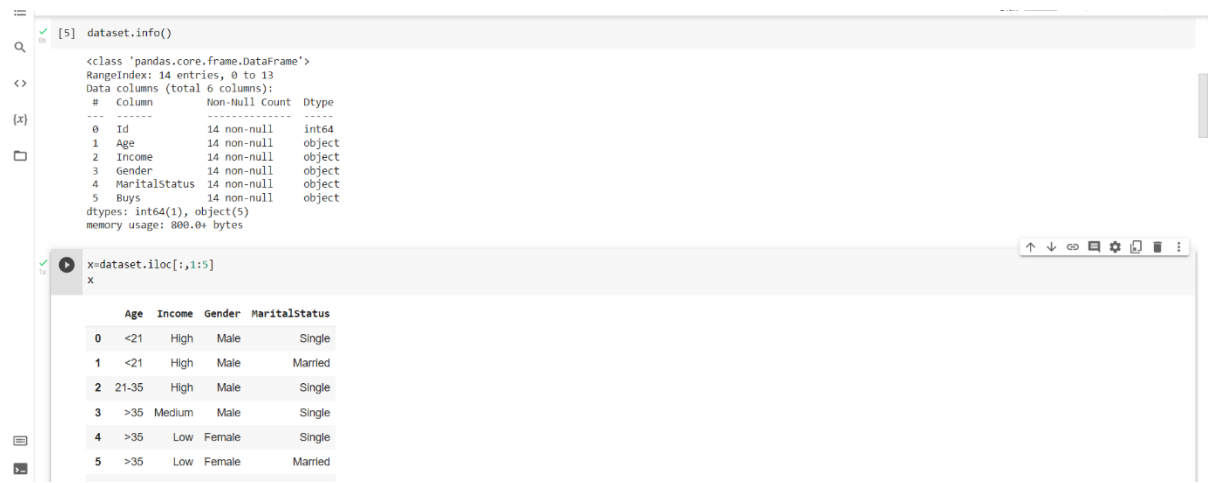
Generate the decision tree node, which contains the best attribute.

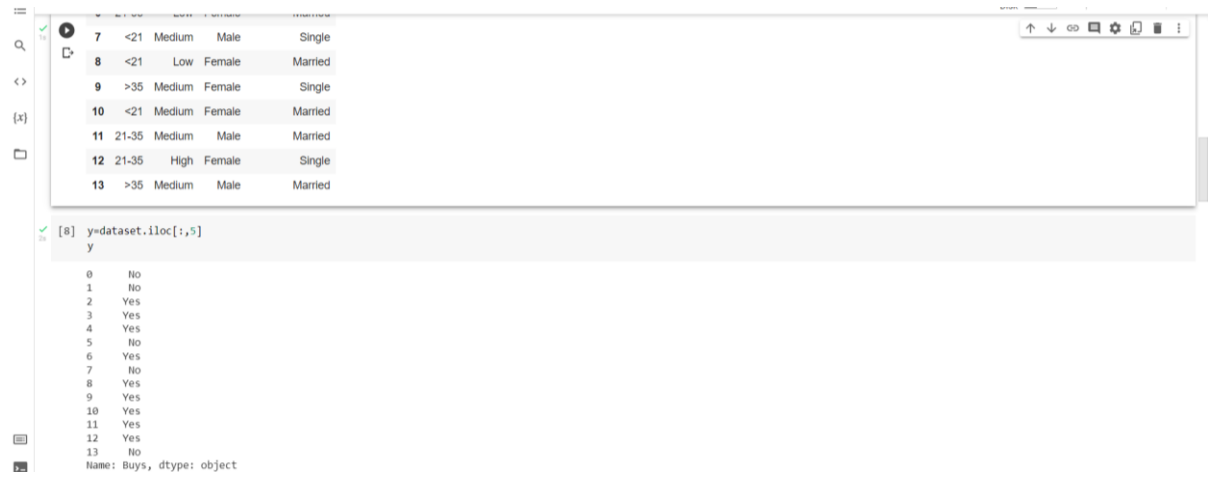
- **Step 5:**

Recursively make new decision trees using the subsets of the dataset created in step 3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

# Code & Output:


Untitled3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
[1] import numpy as np
import pandas as pd
[2] dataset = {
'Id':[0,1,2,3,4,5,6,7,8,9,10,11,12,13],
'Age':['<21','<21','21-35','>35','>35','>35','21-35','<21','<21','>35','<21','21-35','21-35','>35'],
'Income':['High','High','High','Medium','Low','Low','Low','Medium','Low','Medium','Medium','Medium','High','Medium'],
'Gender':['Male','Male','Male','Male','Female','Female','Female','Male','Female','Female','Female','Male','Female','Male'],
'MaritalStatus':['Single','Married','Single','Single','Single','Married','Married','Single','Married','Single','Married','Married','Single','Married'],
'Buys':['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Yes','No']}
[4] dataset=pd.DataFrame.from_dict(dataset)
dataset.head()
Id Age Income Gender MaritalStatus Buys
0 0 <21 High Male Single No
1 1 <21 High Male Married No
2 2 21-35 High Male Single Yes
3 3 >35 Medium Male Single Yes
4 4 >35 Low Female Single Yes
[5] dataset.info()
<class 'pandas.core.frame.DataFrame'>


[5] dataset.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 6 columns):
# Column Non-Null Count Dtype
0 Id 14 non-null int64
1 Age 14 non-null object
2 Income 14 non-null object
3 Gender 14 non-null object
4 MaritalStatus 14 non-null object
5 Buys 14 non-null object
dtypes: int64(1), object(5)
memory usage: 800.0+ bytes
x=dataset.iloc[:,1:5]
x
Age Income Gender MaritalStatus
0 <21 High Male Single
1 <21 High Male Married
2 21-35 High Male Single
3 >35 Medium Male Single
4 >35 Low Female Single
5 >35 Low Female Married


7 <21 Medium Male Single
8 <21 Low Female Married
9 >35 Medium Female Single
10 <21 Medium Female Married
11 21-35 Medium Male Married
12 21-35 High Female Single
13 >35 Medium Male Married
[8] y=dataset.iloc[:,5]
y
0 No
1 No
2 Yes
3 Yes
4 Yes
5 No
6 Yes
7 No
8 Yes
9 Yes
10 Yes
11 Yes
12 Yes
13 No
Name: Buys, dtype: object

```
x
```

| | Age | Income | Gender | MaritalStatus |
|---|---|---|---|---|
| 0 | <21 | High | Male | Single |
| 1 | <21 | High | Male | Married |
| 2 | 21-35 | High | Male | Single |
| 3 | >35 | Medium | Male | Single |
| 4 | >35 | Low | Female | Single |
| 5 | >35 | Low | Female | Married |
| 6 | 21-35 | Low | Female | Married |
| 7 | <21 | Medium | Male | Single |
| 8 | <21 | Low | Female | Married |
| 9 | >35 | Medium | Female | Single |
| 10 | <21 | Medium | Female | Married |
| 11 | 21-35 | Medium | Male | Married |
| 12 | 21-35 | High | Female | Single |
| 13 | >35 | Medium | Male | Married |

```
[11] from sklearn.preprocessing import LabelEncoder
     enc=LabelEncoder()
```

0s completed at 22:23

```
[11] from sklearn.preprocessing import LabelEncoder
     enc=LabelEncoder()
     encoded_x=x
     for i in x.columns:
       encoded_x[i]=enc.fit_transform(x[i])
     print(encoded_x)
```

```
    Age  Income  Gender  MaritalStatus
0     1       0       1              1
1     1       0       1              0
2     0       0       1              1
3     2       2       1              1
4     2       1       0              1
5     2       1       0              0
6     0       1       0              0
7     1       2       1              1
8     1       1       0              0
9     2       2       0              1
10    1       2       0              0
11    0       2       1              0
12    0       0       0              1
13    2       2       1              0
```

```
[12] from sklearn.tree import DecisionTreeClassifier
     model = DecisionTreeClassifier()
     model.fit(x,y)
     y_pred=model.predict(x)
     print('Predicted values: ',y_pred)
     print('Actual Dependant valyes: ',y)
```

```
Predicted values:  ['No' 'No' 'Yes' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes'
 'No']
```

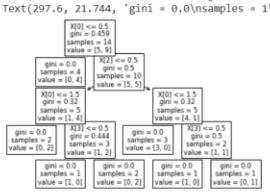
0s completed at 22:23

```
[12] Predicted values:  ['No' 'No' 'Yes' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes'
      'No']
     Actual Dependant valyes:  0      No
     1      No
     2     Yes
     3     Yes
     4     Yes
     5      No
     6     Yes
     7      No
     8     Yes
     9     Yes
     10    Yes
     11    Yes
     12    Yes
     13     No
     Name: Buys, dtype: object
```

```
[13] from sklearn.tree import plot_tree
     plot_tree(model)
```

```
[Text(111.60000000000001, 195.696, 'X[0] <= 0.5\ngini = 0.459\nsamples = 14\nvalue = [5, 9]'),
 Text(74.4, 152.208, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(148.8, 152.208, 'X[2] <= 0.5\ngini = 0.5\nsamples = 10\nvalue = [5, 5]'),
 Text(74.4, 108.72, 'X[0] <= 1.5\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(37.2, 65.232, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(111.60000000000001, 65.232, 'X[3] <= 0.5\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(74.4, 21.744, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(148.8, 21.744, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(223.20000000000002, 108.72, 'X[0] <= 1.5\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(186.0, 65.232, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(260.40000000000003, 65.232, 'X[3] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(223.20000000000002, 21.744, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
```

```
Text(148.8, 21.744, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(223.20000000000002, 108.72, 'X[0] <= 1.5\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
Text(186.0, 65.232, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(260.40000000000003, 65.232, 'X[3] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(223.20000000000002, 21.744, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(297.6, 21.744, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]')]
```




[14]
```
test = [[1,1,0,0]]
```

[15]
```
model.predict(test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:446: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  "X does not have valid feature names, but"
array(['Yes'], dtype=object)
```