

MACHINE LEARNING LAB

EXERCISE :: 3

NAME:: Saptarshi Datta

REG NO:: 19BAI1041

Decision Tree Classification ::

The dataset is about adult income based on their detail. The target attribute is Income which is categorized into ' $\leq 50K$ ' & ' $> 50K$ '.

The dataset used is :- <https://www.kaggle.com/uciml/adult-census-income>

(Few attributes were eliminated).

age	workclass	fnlwgt	education	marital.s	race	sex	capital.g	hours.pe	income
90	?	77053	9	Widowed	White	Female	0	40	$\leq 50K$
82	Private	132870	9	Widowed	White	Female	0	18	$\leq 50K$
66	?	186061	10	Widowed	Black	Female	0	40	$\leq 50K$
54	Private	140359	4	Divorced	White	Female	0	40	$\leq 50K$
41	Private	264663	10	Separate	White	Female	0	40	$\leq 50K$
34	Private	216864	9	Divorced	White	Female	0	45	$\leq 50K$
38	Private	150601	6	Separate	White	Male	0	40	$\leq 50K$
74	State-gov	88638	16	Never-m	White	Female	0	20	$> 50K$
68	Federal-g	422013	9	Divorced	White	Female	0	40	$\leq 50K$
41	Private	70037	10	Never-m	White	Male	0	60	$> 50K$
45	Private	172274	16	Divorced	Black	Female	0	35	$> 50K$
38	Self-emp	164526	15	Never-m	White	Male	0	45	$> 50K$
52	Private	129177	13	Widowed	White	Female	0	20	$> 50K$
32	Private	136204	14	Separate	White	Male	0	55	$> 50K$
51	?	172175	16	Never-m	White	Male	0	40	$> 50K$
46	Private	45363	15	Divorced	White	Male	0	40	$> 50K$
45	Private	172822	7	Divorced	White	Male	0	76	$> 50K$
57	Private	317847	14	Divorced	White	Male	0	50	$> 50K$
22	Private	119592	12	Never-m	Black	Male	0	40	$> 50K$
34	Private	203034	13	Separate	White	Male	0	50	$> 50K$
37	Private	188774	13	Never-m	White	Male	0	40	$> 50K$
29	Private	77009	7	Separate	White	Female	0	42	$\leq 50K$
61	Private	29059	9	Divorced	White	Female	0	25	$\leq 50K$
51	Private	153870	10	Married-	White	Male	0	40	$\leq 50K$

Code ::

```
import pandas as pd
```

```
import numpy as np
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.tree import export_graphviz
```

```
from six import StringIO
```

```
from IPython.display import Image
```

```
import pydotplus
```

```
df = pd.read_csv(r"C:\Users\SAPTARSHI\Desktop\ML\Decision Tree\data\adult.csv");
```

```
df.head()
```

	age	workclass	fnlwgt	education.num	marital.status	race	sex	capital.gain	hours.per.week	income
0	90	?	77053	9	Widowed	White	Female	0	40	<=50K
1	82	Private	132870	9	Widowed	White	Female	0	18	<=50K
2	66	?	186061	10	Widowed	Black	Female	0	40	<=50K
3	54	Private	140359	4	Divorced	White	Female	0	40	<=50K
4	41	Private	264663	10	Separated	White	Female	0	40	<=50K

#Replacing string to integer. All attributes with String Value is replaced with a integer value.

```
salary = {
```

```
    '<=50K' : 0,
```

```
    '>50K' : 1,
```

```
}
```

```
work = {
```

```
    '?' : 0,
```

```
    'Private' : 1,
```

```
    'State-gov' : 2,
```

```
    'Federal-gov' : 3,
```

```
    'Self-emp-not-inc' : 4,
```

```
    'Self-emp-inc' : 5,
```

```

'Local-gov' : 6,
'Federal-gov' : 7,
'Without-pay' : 8,
'Never-worked' : 9
}

marry = {
    'Widowed' : 0,
    'Divorced' : 1,
    'Separated' : 2,
    'Never-married' : 3,
    'Married-civ-spouse': 4,
    'Married-spouse-absent': 5,
    'Married-AF-spouse' : 6
}

race = {
    'White': 0,
    'Black' : 1,
    'Asian-Pac-Islander' : 2,
    'Other' : 3,
    'Amer-Indian-Eskimo' : 4
}

sex = {
    'Female' : 0,
    'Male' : 1,
}

df = df.replace({'income':salary});
df = df.replace({'workclass':work});
df = df.replace({'education':edu});
df = df.replace({'marital.status':marry});
df = df.replace({'race':race});
df = df.replace({'sex':sex});
df.head()

```

	age	workclass	fnlwgt	education.num	marital.status	race	sex	capital.gain	hours.per.week	income
0	90	0	77053	9	0	0	0	0	40	0
1	82	1	132870	9	0	0	0	0	18	0
2	66	0	186061	10	0	1	0	0	40	0
3	54	1	140359	4	1	0	0	0	40	0
4	41	1	264663	10	2	0	0	0	40	0

```
x = df.iloc[:, :9]
```

```
y = df.iloc[:, 9]
```

```
x = np.array(x)
```

```
y = np.array(y)
```

```
df.shape
```

```
(59, 10)
```

```
tree = DecisionTreeClassifier(criterion = 'entropy')
```

```
tree.fit(x,y)
```

```
DecisionTreeClassifier(criterion='entropy')
```

```
dot_data = StringIO()
```

```
export_graphviz(tree, out_file=dot_data,
```

```
    filled=True, rounded=True, special_characters=True)
```

```
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
```

```
Image(graph.create_png(), width=400, height=600)
```

