Sarvesh Karanjkar (20210812002) BDA LAB05 (DECISION TREE)

In [1]:

```
## IMPORTING DATA AND CREATING DATAFRAMES
import pandas as pd
df = pd.read_csv("salaries.csv")
df.head()
```

Out[1]:

	company	job	degree	salary_more_then_100k
0	google	sales executive	bachelors	0
1	google	sales executive	masters	0
2	google	business manager	bachelors	1
3	google	business manager	masters	1
4	google	computer programmer	bachelors	0

In [3]:

```
## DROPPING TARGET COLUMS
inputs = df.drop('salary_more_then_100k',axis='columns')
target = df['salary_more_then_100k']
```

In [6]:

```
## CONVERTING CATEGORICAL VALUES INTO NUMERICAL VALUES
from sklearn.preprocessing import LabelEncoder
le_company = LabelEncoder()
le_job = LabelEncoder()
le_degree = LabelEncoder()


inputs['company_n'] = le_company.fit_transform(inputs['company'])
inputs['job_n'] = le_job.fit_transform(inputs['job'])
inputs['degree_n'] = le_degree.fit_transform(inputs['degree'])

inputs
```

Out[6]:

	company	job	degree	company_n	job_n	degree_n
0	google	sales executive	bachelors	2	2	0
1	google	sales executive	masters	2	2	1
2	google	business manager	bachelors	2	0	0
3	google	business manager	masters	2	0	1
4	google	computer programmer	bachelors	2	1	0
5	google	computer programmer	masters	2	1	1
6	abc pharma	sales executive	masters	0	2	1
7	abc pharma	computer programmer	bachelors	0	1	0
8	abc pharma	business manager	bachelors	0	0	0
9	abc pharma	business manager	masters	0	0	1
10	facebook	sales executive	bachelors	1	2	0
11	facebook	sales executive	masters	1	2	1
12	facebook	business manager	bachelors	1	0	0
13	facebook	business manager	masters	1	0	1
14	facebook	computer programmer	bachelors	1	1	0
15	facebook	computer programmer	masters	1	1	1

In [7]:

```
inputs_n = inputs.drop(['company','job','degree'],axis='columns')
inputs_n
```

Out[7]:

	company_n	job_n	degree_n
0	2	2	0
1	2	2	1
2	2	0	0
3	2	0	1
4	2	1	0
5	2	1	1
6	0	2	1
7	0	1	0
8	0	0	0
9	0	0	1
10	1	2	0
11	1	2	1
12	1	0	0
13	1	0	1
14	1	1	0
15	1	1	1

In [10]:

```
######## BUILDING THE TREE

from sklearn import tree
model = tree.DecisionTreeClassifier()

###### TRAINING MODEL
model.fit(inputs_n, target)

#### CHECKING SCORE
model.score(inputs_n,target)
```

Out[10]:

```
1.0
```

In [11]:

```
######### salary of Google, Computer Engineer, Bachelors degree > 100 k ?

model.predict([[2,1,0]])
```

```
C:\Users\sarvesh\anaconda3\lib\site-packages\sklearn\base.py:439: UserWarn
ing: X does not have valid feature names, but DecisionTreeClassifier was f
itted with feature names
  warnings.warn(
```

Out[11]:

```
array([0], dtype=int64)
```

Conclusion - Successfully build decision tree using python on salaries Dataset