In [1]:
```
import pandas as pd
import seaborn as sns
```

In [12]:
```
df=pd.read_csv("salaries.csv")
```

In [13]:
```
df.head()
```

Out[13]:

	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 [14]:
```
df = df.rename(columns={'salary_more_then_100k':'salary>100k'})  #change the column name
```

In [15]:
```
df
```

Out[15]:

	company	job	degree	salary>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
5	google	computer programmer	masters	1
6	abc pharma	sales executive	masters	0
7	abc pharma	computer programmer	bachelors	0
8	abc pharma	business manager	bachelors	0
9	abc pharma	business manager	masters	1
10	facebook	sales executive	bachelors	1
11	facebook	sales executive	masters	1
12	facebook	business manager	bachelors	1
13	facebook	business manager	masters	1
14	facebook	computer programmer	bachelors	1
15	facebook	computer programmer	masters	1

In [17]:
```
independ=df.drop("salary>100k",axis=1)
```

In [18]:
```
depend=df['salary>100k']
```

In [19]:
```
depend
```

Out[19]:
```
0     0
1     0
2     1
3     1
4     0
5     1
6     0
7     0
8     0
9     1
10    1
11    1
12    1
13    1
14    1
15    1
Name: salary>100k, dtype: int64
```

In [20]:
```
independ
```

Out[20]:

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

In [21]:
```
from sklearn.preprocessing import LabelEncoder
```

In [22]:
```
lacompany=LabelEncoder()
lajob=LabelEncoder()                    #here we create the object
ladegree=LabelEncoder()
```

In [24]:
```
independ
```

Out[24]:

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

In [27]:
```
independ['companynum']=lacompany.fit_transform(independ.company) # add new column companynum based on old columns
independ['jobnum']=lacompany.fit_transform(independ.job)
independ['degreenum']=lacompany.fit_transform(independ.degree)
```

In [28]:
```
independ
```

Out[28]:

	company	job	degree	companynum	jobnum	degreenum
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 [31]:
```
independ=independ.drop(['company','job','degree'],axis=1)   #now we dont need old columns so we delete them.
```

In [32]:
```
independ
#here the
'''
google=2, abc=0, facebook=1
sale=2,business manager=0,computer programming=1
bachelor =0, master=1
'''
```

Out[32]:

	companynum	jobnum	degreenum
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 [34]:
```
from sklearn import tree
```

In [35]:
```
reg=tree.DecisionTreeClassifier()  #this is model
```

In [39]:
```
from sklearn.model_selection import train_test_split
```

In [55]:
```
x_train,x_test,y_train,y_test=train_test_split(independ,depend,test_size=0.5)
```

In [56]:
```
reg.fit(x_train,y_train)  #here we train
```

Out[56]:
```
DecisionTreeClassifier
DecisionTreeClassifier()
```

In [57]:
```
reg.predict([[2,2,0]])  # google ma sale ko kam garney bachelor degree padeko manche ko salary less than 100K
```

```
C:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

Out[57]:
```
array([0], dtype=int64)
```

In [58]:
```
reg.score(x_test,y_test) # accuracy of my model
```

Out[58]:
```
0.875
```

In [ ]:

In [ ]: