In [1]:
```
import pandas as pd
df=pd.read_csv("salaries.csv")
print(df)
```

```
        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
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 [2]:

```
x=df.drop("salary_more_then_100k",axis="columns")
y=df["salary_more_then_100k"]
print(x)
print(y)
```

```
       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
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_more_then_100k, dtype: int64
```

In [3]:

```
x.head()
y.head()
```

Out[3]:

```
0    0
1    0
2    1
3    1
4    0
Name: salary_more_then_100k, dtype: int64
```

In [4]:
```
from sklearn.preprocessing import LabelEncoder
s=LabelEncoder()
x["new_company"]=s.fit_transform(df["company"])
print(x)
```

```
       company                  job     degree  new_company
0       google      sales executive  bachelors            2
1       google      sales executive    masters            2
2       google     business manager  bachelors            2
3       google     business manager    masters            2
4       google  computer programmer  bachelors            2
5       google  computer programmer    masters            2
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            0
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 [5]:
```
x["new_jobs"]=s.fit_transform(df["job"])
print(x)
```

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

In [6]:

```
x["new_degree"]=s.fit_transform(df["degree"])
x
```

Out[6]:

	company	job	degree	new_company	new_jobs	new_degree
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]:

```
a=df.drop(["company","job","degree"],axis="columns")
print(a)
```

```
    salary_more_then_100k
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
```

In [8]:

```
from sklearn.tree import DecisionTreeClassifier
model=DecisionTreeClassifier()
model.fit(a.values,y)
```

Out[8]:

```
DecisionTreeClassifier()
```

In [9]:

```
print(model.score(a,y))
```

```
1.0

D:\anaconda\lib\site-packages\sklearn\base.py:443: UserWarning: X has feature
names, but DecisionTreeClassifier was fitted without feature names
  warnings.warn(
```

In [10]:

```
import pandas as pd
df=pd.read_csv("salaries.csv")
df
```

Out[10]:

	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
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 [11]:

```
x=df.drop("salary_more_then_100k",axis="columns")
y=df["salary_more_then_100k"]
x
y
```

Out[11]:

```
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_more_then_100k, dtype: int64
```

In [12]:

```
from sklearn.preprocessing import LabelEncoder
z=LabelEncoder()
x["company_new"]=z.fit_transform(df["company"])
x
```

Out[12]:

	company	job	degree	company_new
0	google	sales executive	bachelors	2
1	google	sales executive	masters	2
2	google	business manager	bachelors	2
3	google	business manager	masters	2
4	google	computer programmer	bachelors	2
5	google	computer programmer	masters	2
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	0
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 [13]:

```
x["job_new"]=z.fit_transform(df["job"])
x
```

Out[13]:

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

In [14]:

```
x["degree_new"]=z.fit_transform(df["degree"])
x
```

Out[14]:

	company	job	degree	company_new	job_new	degree_new
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 [15]:

```
s=x.drop(["company","job","degree"],axis="columns")
s
```

Out[15]:

	company_new	job_new	degree_new
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 [16]:

```
from sklearn.tree import DecisionTreeClassifier
c=DecisionTreeClassifier()
c.fit(s,y)
```

Out[16]:

```
DecisionTreeClassifier()
```

In [17]:

```
print(c.predict([[2,1,0]]))
```

```
[0]

D:\anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not ha
ve valid feature names, but DecisionTreeClassifier was fitted with feature na
mes
  warnings.warn(
```

In [18]:

```
print(c.score(s,y))
```

```
1.0
```

randomforest classifier

In [19]:
```
from sklearn.datasets import load_iris
iris=load_iris()
print(dir(iris))
```

```
['DESCR', 'data', 'data_module', 'feature_names', 'filename', 'frame', 'targe
t', 'target_names']
```

In [20]:
```
iris.data
```

Out[20]:
```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
       [5.1, 3.8, 1.5, 0.3],
```

In [21]:
```
iris.data_module
```

Out[21]:
```
'sklearn.datasets.data'
```

In [22]:
```
iris.feature_names
```

Out[22]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [23]:
```
iris.filename
```

Out[23]:
```
'iris.csv'
```

In [24]:
```
iris.frame
```

In [25]:
```
iris.target
```

Out[25]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [26]:
```
iris.target_names
```

Out[26]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [27]:
```
print(len(iris.data))
```

```
150
```

In [28]:
```
df=pd.DataFrame(iris.data,columns=iris.feature_names)
df
```

Out[28]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [29]:
```
df["target"]=iris.target
df
```

Out[29]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

In [30]:
```
print(df[df.target==0].head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2   
1                4.9               3.0                1.4               0.2   
2                4.7               3.2                1.3               0.2   
3                4.6               3.1                1.5               0.2   
4                5.0               3.6                1.4               0.2   

   target  
0       0  
1       0  
2       0  
3       0  
4       0  
```

In [31]:
```
print(len(df[df.target==0]))
```

```
50
```

In [32]:
```
len(df[df.target==1])
```

Out[32]:
```
50
```

In [33]:
```
len(df[df.target==2])
```

Out[33]:
```
50
```

In [34]:
```
a=lambda x:iris.target_names[x]
df["flower_name"]=df.target.apply(a)
df
```

Out[34]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa
...	...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2	virginica
146	6.3	2.5	5.0	1.9	2	virginica
147	6.5	3.0	5.2	2.0	2	virginica
148	6.2	3.4	5.4	2.3	2	virginica
149	5.9	3.0	5.1	1.8	2	virginica

150 rows × 6 columns

In [35]:
```
a=lambda x:iris.target_names[x]
df["flower_name"]=df.target.apply(a)
df
```

Out[35]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa
...	...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2	virginica
146	6.3	2.5	5.0	1.9	2	virginica
147	6.5	3.0	5.2	2.0	2	virginica
148	6.2	3.4	5.4	2.3	2	virginica
149	5.9	3.0	5.1	1.8	2	virginica

150 rows × 6 columns

In [36]:
```
setosa_50=df[0:50]
setosa_50.head()
```

Out[36]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa

In [37]:
```
versicolor_50=df[50:100]
versicolor_50.head()
```

Out[37]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
50	7.0	3.2	4.7	1.4	1	versicolor
51	6.4	3.2	4.5	1.5	1	versicolor
52	6.9	3.1	4.9	1.5	1	versicolor
53	5.5	2.3	4.0	1.3	1	versicolor
54	6.5	2.8	4.6	1.5	1	versicolor

In [38]:
```
virginica_50=df[100:150]
virginica_50.head()
```

Out[38]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
100	6.3	3.3	6.0	2.5	2	virginica
101	5.8	2.7	5.1	1.9	2	virginica
102	7.1	3.0	5.9	2.1	2	virginica
103	6.3	2.9	5.6	1.8	2	virginica
104	6.5	3.0	5.8	2.2	2	virginica

In [39]:
```
from sklearn.model_selection import train_test_split
X=df.drop(["target","flower_name"],axis="columns")
y=df.target
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

In [40]:
```
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier(n_estimators=40)
model.fit(X_train,y_train)
```

Out[40]:
```
RandomForestClassifier(n_estimators=40)
```

```
In [41]: print(model.score(X_test,y_test))
```

```
0.9333333333333333
```

```
In [42]: print(model.predict([[4.8,3.0,1.5,0.3]]))
```

```
[0]

D:\anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not ha
ve valid feature names, but RandomForestClassifier was fitted with feature na
mes
  warnings.warn(
```

```
In [ ]:
```