In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [5]:
```
df = pd.read_csv('HR_comma_sep.csv')
df.head(25)
```

Out[5]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company
0	0.38	0.53	2	157	3
1	0.80	0.86	5	262	6
2	0.11	0.88	7	272	4
3	0.72	0.87	5	223	5
4	0.37	0.52	2	159	3
5	0.41	0.50	2	153	3
6	0.10	0.77	6	247	4
7	0.92	0.85	5	259	5
8	0.89	1.00	5	224	5
9	0.42	0.53	2	142	3
10	0.45	0.54	2	135	3
11	0.11	0.81	6	305	4
12	0.84	0.92	4	234	5
13	0.41	0.55	2	148	3
14	0.36	0.56	2	137	3
15	0.38	0.54	2	143	3
16	0.45	0.47	2	160	3
17	0.78	0.99	4	255	6
18	0.45	0.51	2	160	3
19	0.76	0.89	5	262	5
20	0.11	0.83	6	282	4
21	0.38	0.55	2	147	3
22	0.09	0.95	6	304	4
23	0.46	0.57	2	139	3
24	0.40	0.53	2	158	3

In [6]:
```
df.Department.unique()
```

Out[6]:
```
array(['sales', 'accounting', 'hr', 'technical', 'support', 'management',
       'IT', 'product_mng', 'marketing', 'RandD'], dtype=object)
```

In [7]:
```
left = df[df.left == 1]
left.shape
```

Out[7]:
```
(3571, 10)
```

In [8]:
```
retained = df [df.left == 0]
retained.shape
```

Out[8]:
```
(11428, 10)
```

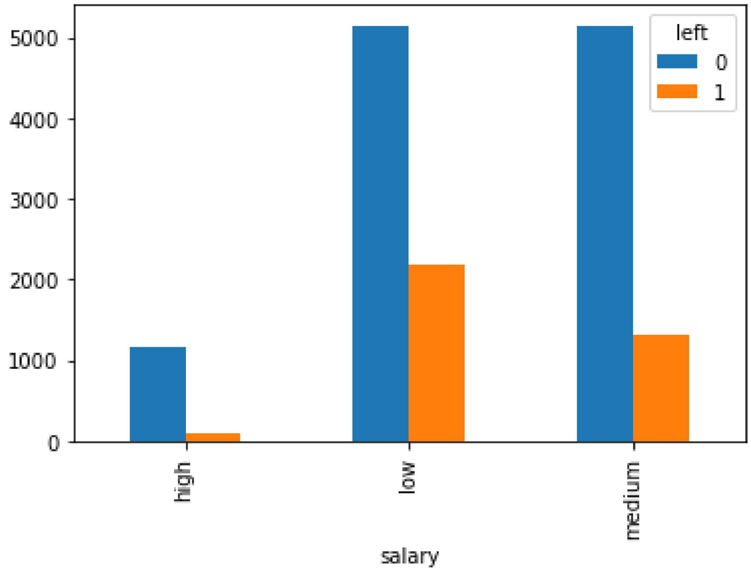
In [35]:
```
df.groupby('left').mean()
```

Out[35]:

left	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company
0	0.666810	0.715473	3.786664	199.060203	3.380032
1	0.440098	0.718113	3.855503	207.419210	3.876505

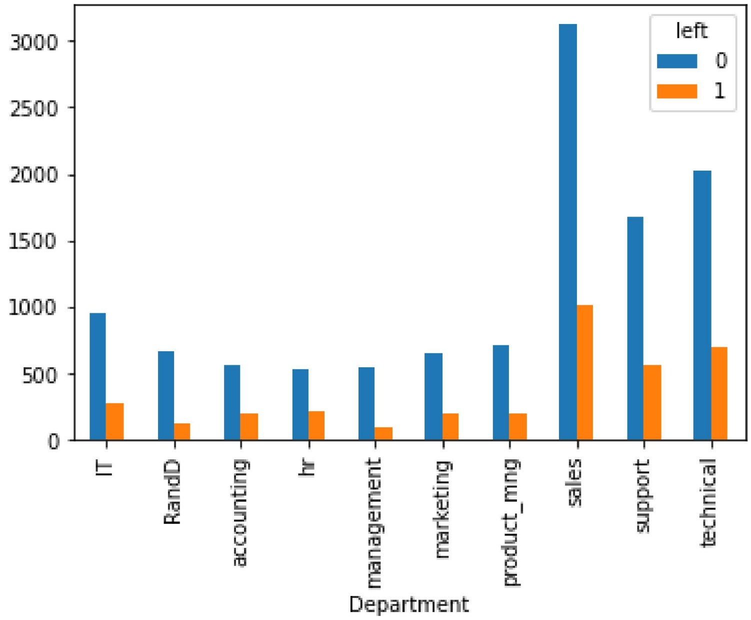
In [9]:
```
pd.crosstab(df.salary , df.left).plot(kind = 'bar')
```

Out[9]:
```
<AxesSubplot:xlabel='salary'>
```



In [10]:
```
pd.crosstab(df.Department , df.left).plot(kind = 'bar')
```

Out[10]:
```
<AxesSubplot:xlabel='Department'>
```



In [21]:
```
new_df = pd.get_dummies(df.salary   )
new_df.head()
```

Out[21]:

	high	low	medium
0	0	1	0
1	0	0	1
2	0	0	1
3	0	1	0
4	0	1	0

In [22]:
```
new_df = pd.concat([df , new_df] , axis = 'columns' )
new_df.head()
```

Out[22]:

_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident
0.38	0.53	2	157	3	0
0.80	0.86	5	262	6	0
0.11	0.88	7	272	4	0
0.72	0.87	5	223	5	0
0.37	0.52	2	159	3	0

In [42]:
```
subdf = new_df[['satisfaction_level' , 'average_montly_hours','promotion_last_5ye
subdf.head()
```

Out[42]:

	satisfaction_level	average_montly_hours	promotion_last_5years	salary	high	low	medium
0	0.38	157	0	low	0	1	0
1	0.80	262	0	medium	0	0	1
2	0.11	272	0	medium	0	0	1
3	0.72	223	0	low	0	1	0
4	0.37	159	0	low	0	1	0

In [45]:
```
dff = subdf.drop('salary' , axis = 'columns' )
dff.head()
```

Out[45]:

	satisfaction_level	average_montly_hours	promotion_last_5years	high	low	medium
0	0.38	157	0	0	1	0
1	0.80	262	0	0	0	1
2	0.11	272	0	0	0	1
3	0.72	223	0	0	1	0
4	0.37	159	0	0	1	0

In [46]:
```
from sklearn.model_selection import train_test_split
```

In [47]:
```
x = dff
x.head()
```

Out[47]:

	satisfaction_level	average_montly_hours	promotion_last_5years	high	low	medium
0	0.38	157	0	0	1	0
1	0.80	262	0	0	0	1
2	0.11	272	0	0	0	1
3	0.72	223	0	0	1	0
4	0.37	159	0	0	1	0

In [49]:
```
y = df.left
```

In [58]:
```
x_train , x_test , y_train , y_test = train_test_split(x ,y ,train_size = 0.3)
```

In [59]:
```
len(x_train)
```

Out[59]:
```
4499
```

```
In [65]: from sklearn.linear_model import LogisticRegression

In [66]: model = LogisticRegression()

In [67]: model.fit(x_train , y_train)

Out[67]: LogisticRegression()

In [68]: model.predict(x_test)

Out[68]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)

In [69]: model.score(x_test , y_test)

Out[69]: 0.7777142857142857

In [65]:

In [ ]:

In [ ]:
```