```
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
%matplotlib inline
```

```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/HR_comma_sep.csv')
df.head()
```

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spen
0	0.38	0.53	2	157	
1	0.80	0.86	5	262	
2	0.11	0.88	7	272	
3	0.72	0.87	5	223	
4	0.37	0.52	2	159	

```
left = df[df.left==1]
left.shape
```

```
(3571, 10)
```

```
retained = df[df.left==0]
retained.shape
```

```
(11428, 10)
```

```
df.groupby('left').mean()
```

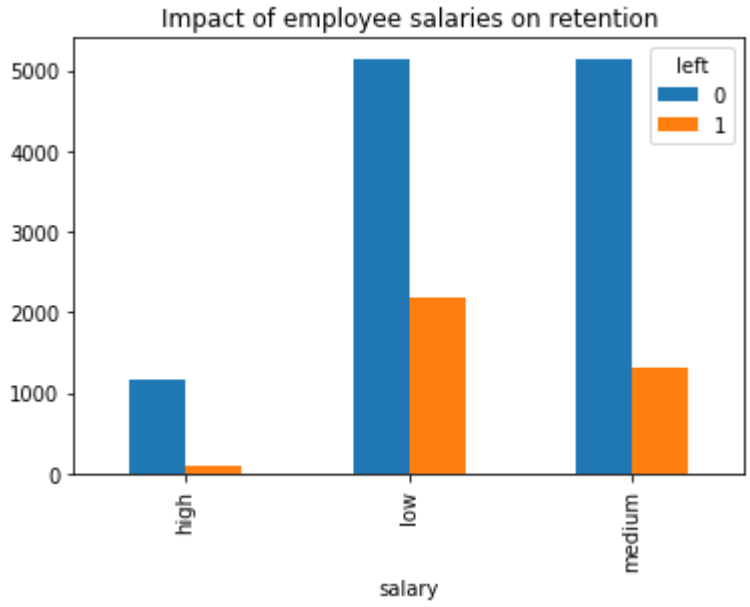
left	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_s
0	0.666810	0.715473	3.786664	199.060203	
1	0.440098	0.718113	3.855503	207.419210	

```
df.groupby('salary').mean()
```

salary	satisfaction_level	last_evaluation	number_project	average_montly_hours	time
high	0.637470	0.704325	3.767179	199.867421	

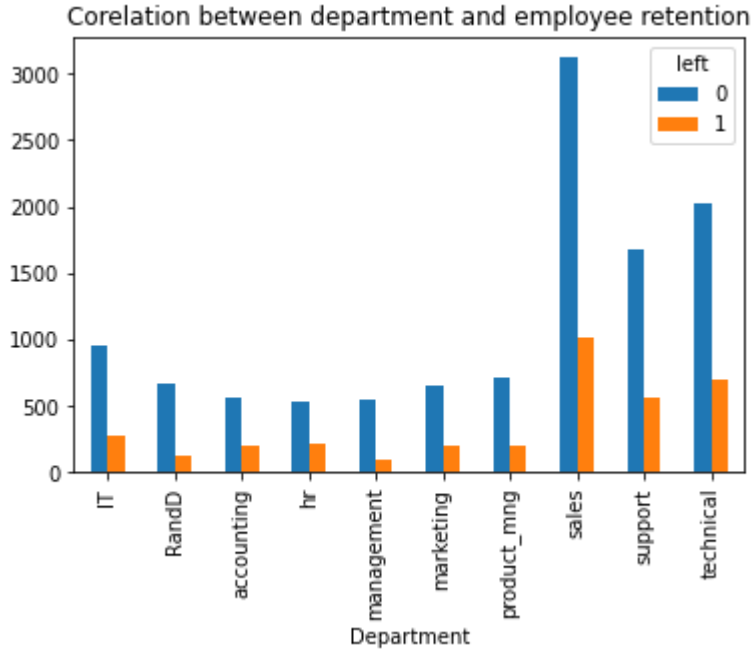
```
pd.crosstab(df.salary,df.left).plot(kind='bar',title='Impact of employee salaries on retentio
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37cb4adb90>
```




```
pd.crosstab(df.Department,df.left).plot(kind='bar', title='Corelation between department and
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37cb40ba90>
```




```
dflr = df[['satisfaction_level','average_montly_hours','promotion_last_5years','salary']]
dflr.head()
```

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

```
salary_dummies = pd.get_dummies(dflr.salary,prefix='salary')
df_dummies = pd.concat([dflr,salary_dummies],axis='columns')
df_dummies.drop('salary',axis='columns',inplace=True)
df_dummies.head()
```

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

```
X = df_dummies
y = df.left


X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)


model = LogisticRegression()
model.fit(X_train,y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
model.predict(X_test)
```

```
array([0, 0, 0, ..., 0, 0, 1])
```

```
model.score(X_test,y_test)
```

```
0.7726666666666666
```

✓ 0s completed at 16:44