Practical - 14

AIM

Implement a classification/ logistic regression problem.

PROBLEM

Based on a dataset predict whether an employee will leave a company or not.

CODE & OUTPUT

In [2]:

```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

In [4]:

```
df = pd.read_csv("employee_retention.csv")
df.head()
```

Out[4]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	le
0	0.38	0.53	2	157	3	0	
1	0.80	0.86	5	262	6	0	
2	0.11	0.88	7	272	4	0	
3	0.72	0.87	5	223	5	0	
4	0.37	0.52	2	159	3	0	

In [5]:

```
# exploring data to see which attributes have direct impact on employee retention
df.groupby('left').mean()
```

Out[5]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident
left						
0	0.666810	0.715473	3.786664	199.060203	3.380032	0.175009
1	0.440098	0.718113	3.855503	207.419210	3.876505	0.047326

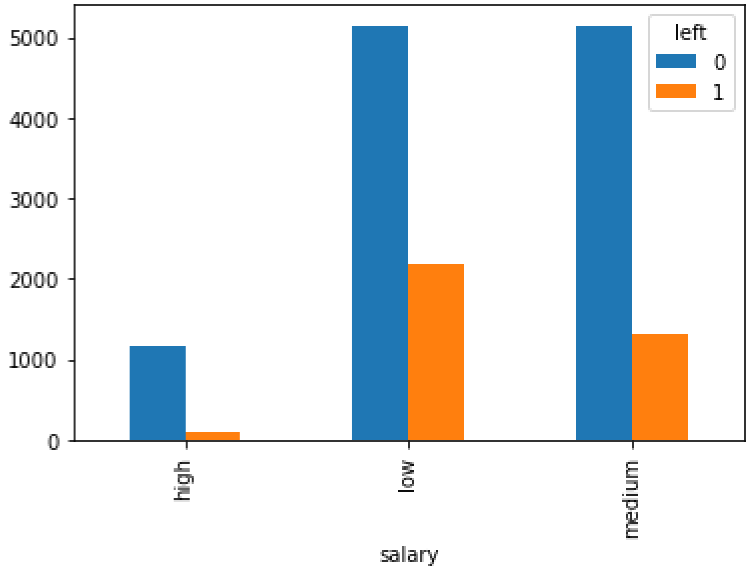
In [6]:

```
# From the above evidence, we can say that attributes that affect an employee leaving or i
# 1. Satisfaction Level (satisfaction level of employees leaving is relatively low that th
# 2. Average Monthly Hours (employees who left worked more on an average that the employee
# 3. Promotion Last 5 Years (employees who got a promotion in the last 5 years are more li

# Now we see the impact of Salary on the employees leaving the firm
```

Out[6]:

```
<AxesSubplot:xlabel='salary'>
```



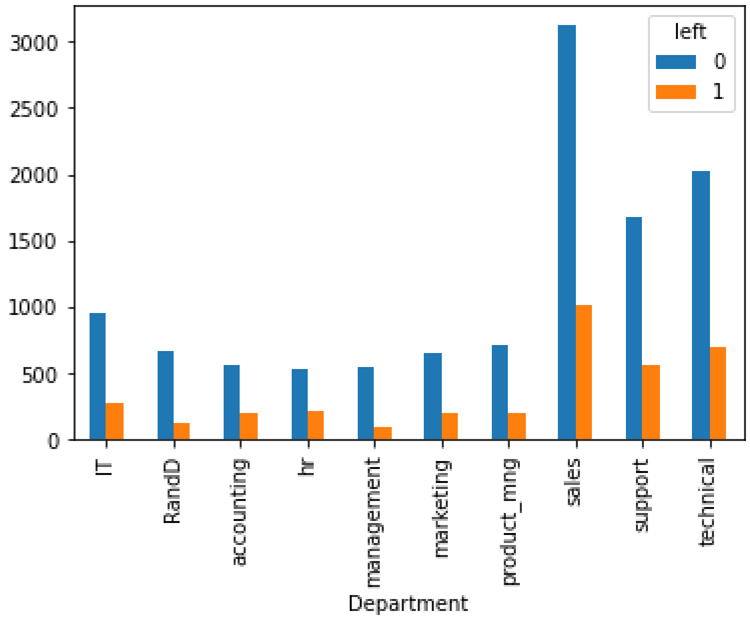
In [7]:

```
# We can see that employees with high salary are less likely to leave the company

# Now we see the department wise retention rate
pd.crosstab(df.Department,df.left).plot(kind='bar')
```

Out[7]:

```
<AxesSubplot:xlabel='Department'>
```



In [9]:

```
# From the above evidence, department have some impact on employee retention but not hat r
# consider department as important feature

# Thus, we have 4 independent variables for our model:
# 1. Satisfaction Level
# 2. Average Monthly Hours
# 3. Promotion Last 5 Years
# 4. Salary

subset_df = df[['satisfaction_level', 'average_montly_hours', 'promotion_last_5years', 'sa
subset_df.head()
```

Out[9]:

	satisfaction_level	average_montly_hours	promotion_last_5years	salary
0	0.38	157	0	low

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

In [12]:

```
# we have to convert Salary into a numeric field to be able to do the analysis
salary_dummies = pd.get_dummies(subset_df.salary, prefix = "salary") # get the salary_high
df_with_dummies = pd.concat([subset_df, salary_dummies], axis="columns") # concatenate the
df_with_dummies.drop("salary", axis="columns", inplace=True) # drop the original salary fi
df_with_dummies.head()
```

Out[12]:

	satisfaction_level	average_montly_hours	promotion_last_5years	salary_high	salary_low	salary_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 [13]:

```
X = df_with_dummies
y = df.left
```

In [19]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.3)
```

In [20]:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
```

Out[20]:

```
LogisticRegression()
```

In [21]:

```
y_pred = model.predict(X_test)
```

In [22]:

```
print(y_pred)
```

```
[0 0 0 ... 0 0 0]
```

In [23]:

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

Out[23]:

```
0.7764761904761904
```