Exercise for Logistic Regression

Download employee retention dataset from here: https://www.kaggle.com/giripujar/hr-analytics (https://www.kaggle.com/giripujar/hr-analytics).

1. Now do some exploratory data analysis to figure out which variables have direct and clear impact on employee retention by using groupby (i.e. whether they leave the company or continue to work)
2. Plot bar charts showing impact of employee salaries on retention
3. Plot bar charts showing correlation between department and employee retention
4. Now build logistic regression model using variables that were narrowed down in step 1
5. Measure the accuracy of the model

In [2]:
```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

In [6]:
```
 = pd.read_csv('D:/Data_Science/My Github/Machine-Learning-with-Python/7. logistic
.head()
```

Out[6]:

ı	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5y
ȝ	2	157	3	0	1	
ȝ	5	262	6	0	1	
ȝ	7	272	4	0	1	
'	5	223	5	0	1	
ȝ	2	159	3	0	1	

Data exploration and visualization

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)

Average numbers for all columns

In [9]:
```
df.groupby('left').mean()
```

Out[9]:

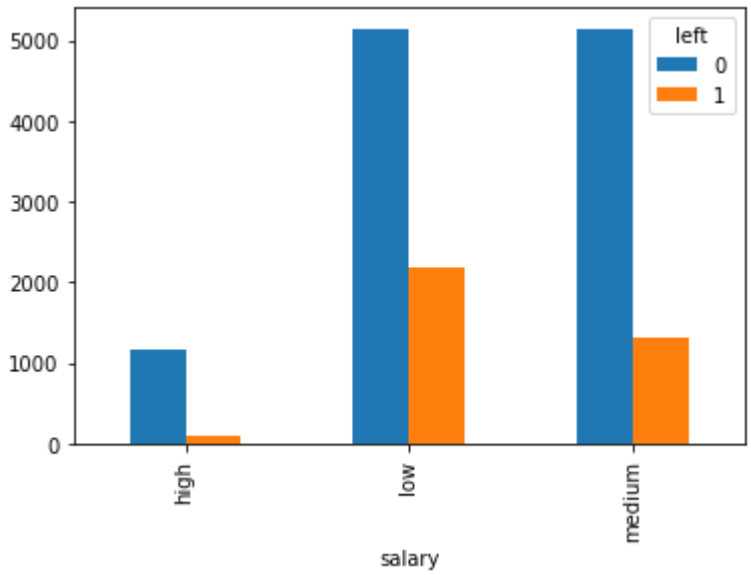
left	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company
0	0.666810	0.715473	3.786664	199.060203	3.380032
1	0.440098	0.718113	3.855503	207.419210	3.876505

From above table we can draw following conclusions:
1.**Satisfaction Level**: Satisfaction level seems to be relatively low (0.44) in employees leaving the firm vs the retained ones (0.66)
2.**Average Monthly Hours**: Average monthly hours are higher in employees leaving the firm (207 vs 199)
3.**Promotion Last 5 Years**: Employees who are given promotion are likely to be retained at firm

Impact of salary on employee retention with pd.crosstab

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

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

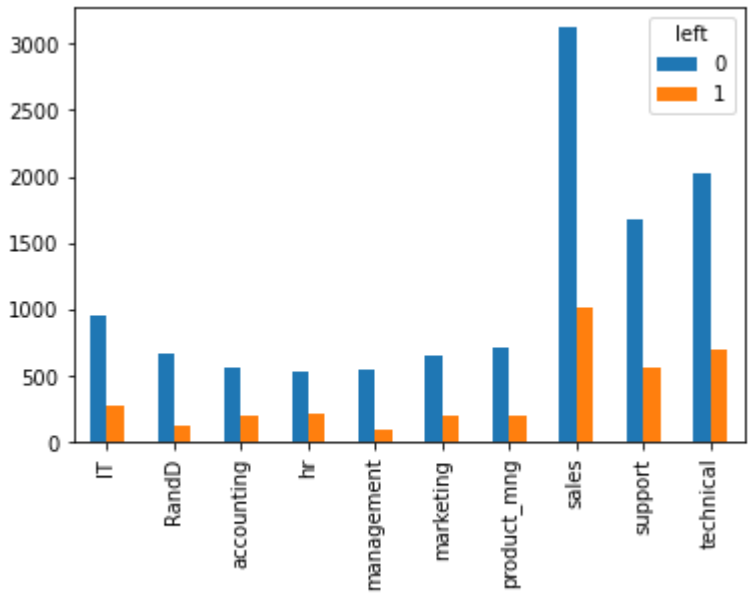


Above bar chart shows employees with high salaries are likely to not leave the company

Department wise employee retention rate

In [11]:
```
pd.crosstab(df.Department,df.left).plot(kind='bar')
```

Out[11]:
```
<AxesSubplot:xlabel='Department'>
```



From above chart there seem to be some impact of department on employee retention but it is not major hence we will ignore department in our analysis

From the data analysis so far we can conclude that we will use following variables as independant variables in our model

1.**Satisfaction Level**
2.**Average Monthly Hours**
3.**Promotion Last 5 Years**
4.**Salary**

In [13]:
```
subdf = df[['satisfaction_level','average_montly_hours','promotion_last_5years','s
subdf.head()
```

Out[13]:

	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

Tackle salary dummy variable

Salary has all text data. It needs to be converted to numbers and we will use dummy variable for that.

In [14]:
```
salary_dummies = pd.get_dummies(subdf.salary, prefix="salary")
```

In [15]:
```
df_with_dummies = pd.concat([subdf,salary_dummies],axis='columns')
```

In [16]:
```
df_with_dummies.head()
```

Out[16]:

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

Now we need to remove salary column which is text data. It is already replaced by dummy variables so we can safely remove it

In [17]:
```
df_with_dummies.drop('salary',axis='columns',inplace=True)
df_with_dummies.head()
```

Out[17]:

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

In [18]:
```
X = df_with_dummies
X.head()
```

Out[18]:

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

In [19]:
```
y = df.left
```

In [20]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,train_size=0.8)
```

In [21]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

In [22]:
```
model.fit(X_train, y_train)
```

Out[22]:
```
LogisticRegression()
```

In [23]:
```
model.predict(X_test)
```

Out[23]:
```
array([0, 1, 0, ..., 0, 0, 0], dtype=int64)
```

Accuracy of the model

In [24]:
```
model.score(X_test,y_test)
```

Out[24]:
```
0.7716666666666666
```

Date	Author
2021-09-02	Ehsan Zia