logistic regression

June 17, 2022

[1]:
```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

[2]:
```
df=pd.read_csv("HR_comma_sep.csv")
df.head()
```

[2]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	Department	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

1 Data Exploration and visualization

[4]:
```
left=df[df.left==1]
left.shape
```

[4]: (3571, 10)

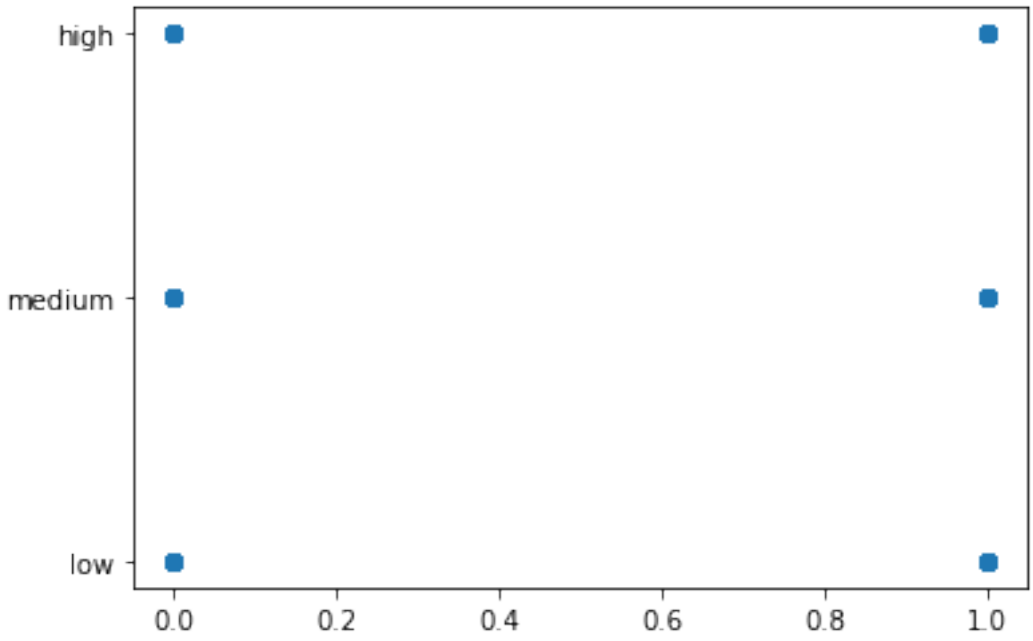
[73]:
```
retained=df[df.left==0]
retained.shape
```

[73]: (11428, 10)

Average number for all cplumns

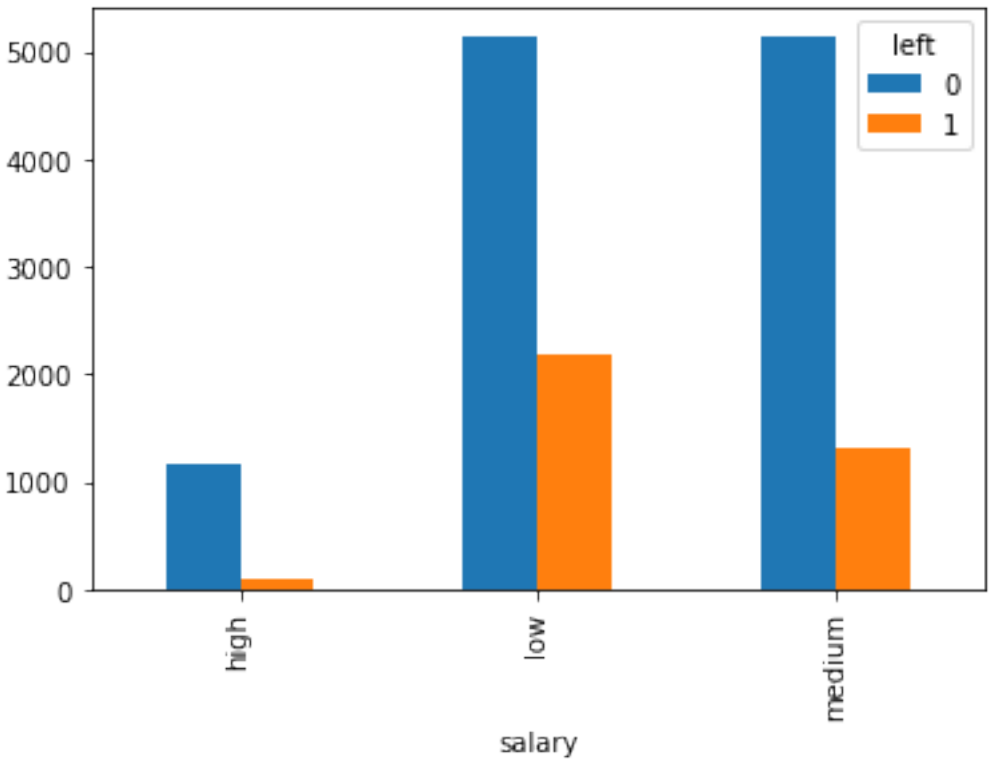
[75]:
```
plt.scatter(df.left,df.salary)
```

[75]: <matplotlib.collections.PathCollection at 0x198656d44f0>



[8]:
```
pd.crosstab(df.salary,df.left).plot(kind='bar')
```

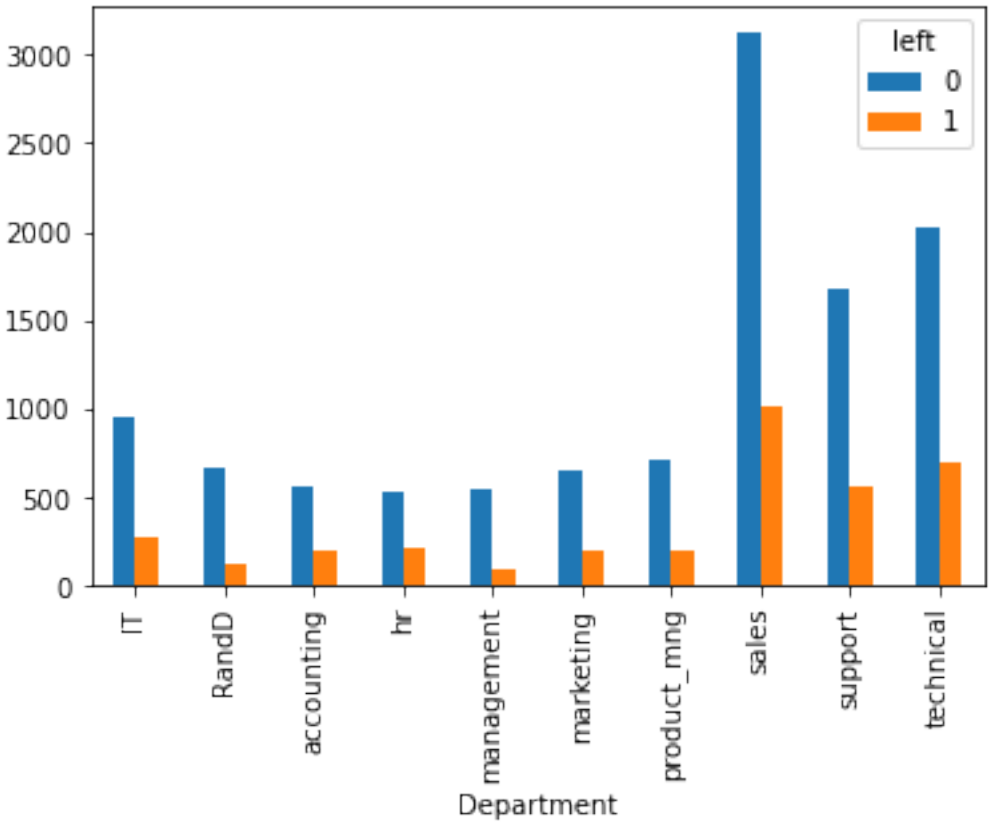
[8]: <AxesSubplot:xlabel='salary'>



[ ]:

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

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



[58]:
```
subdf=df[['satisfaction_level','average_montly_hours','promotion_last_5years','salary']]
subdf.head()
```

[58]:

	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

[18]:
```
salary_dummies=pd.get_dummies(subdf.salary,prefix="salary")
```

[21]:
```
df_with_dummies=pd.concat([subdf,salary_dummies],axis='columns')
```

[24]:
```
df_with_dummies.head()
```

[24]:

	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_high  salary_low  salary_medium
0            0           1              0
1            0           0              1
2            0           0              1
3            0           1              0
4            0           1              0
```

now we need to remove salary column which is text data . it is already replaced by dummy variable so we can safely remove it

[59]:

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-59-666563193c58> in <module>
----> 1 df_with_dummies.drop('salary',axis='columns',inplace=True)
      2 df_with_dummies.head()

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in drop(self,␣
↪labels, axis, index, columns, level, inplace, errors)
   4306                 weight  1.0     0.8
   4307         """
-> 4308         return super().drop(
   4309             labels=labels,
   4310             axis=axis,

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in drop(self,
↪labels, axis, index, columns, level, inplace, errors)
   4151         for axis, labels in axes.items():
   4152             if labels is not None:
-> 4153                 obj = obj._drop_axis(labels, axis, level=level,␣
↪errors=errors)
   4154
   4155         if inplace:

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in␣
↪_drop_axis(self, labels, axis, level, errors)
   4186                 new_axis = axis.drop(labels, level=level, errors=errors
   4187             else:
-> 4188                 new_axis = axis.drop(labels, errors=errors)
   4189             result = self.reindex(**{axis_name: new_axis})
   4190
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in␣
↪drop(self, labels, errors)
   5589         if mask.any():
   5590             if errors != "ignore":
-> 5591                 raise KeyError(f"{labels[mask]} not found in axis")
   5592             indexer = indexer[~mask]
   5593         return self.delete(indexer)

KeyError: "['salary'] not found in axis"
```

[63]:
```
X=df_with_dummies[['satisfaction_level','average_montly_hours','promotion_last_5years','salary
X.head()
```

[63]:
```
   satisfaction_level  average_montly_hours  promotion_last_5years  \
0                0.38                   157                      0
1                0.80                   262                      0
2                0.11                   272                      0
3                0.72                   223                      0
4                0.37                   159                      0

   salary_high  salary_low  salary_medium
0            0           1              0
1            0           0              1
2            0           0              1
3            0           1              0
4            0           1              0
```

[60]:
```
y=df.left
y.head()
```

[60]:
```
0    1
1    1
2    1
3    1
4    1
Name: left, dtype: int64
```

[61]:
```
from sklearn.linear_model import LogisticRegression
model= LogisticRegression()
```

[68]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train,y_test=train_test_split(X,y,test_size=0.3)
```

[69]:
```
model.fit(X_train,y_train)
```

[69]:
```
LogisticRegression()
```

[71]:
```
model.predict(X_test)
```

[71]:
```
array([0, 0, 0, …, 1, 0, 0], dtype=int64)
```

[72]:
```
model.score(X_test,y_test)
```

[72]:
```
0.7726666666666666
```