In [51]:
```
import pandas as pd
import numpy as np
```

In [52]:
```
df=pd.read_csv('DATA/HR_comma_sep.csv')
df.head()
```

Out[52]:

| | satisfaction_level | last_evaluation | number_project | average_montly_hours | time_spend_company | Wo |
|---|---|---|---|---|---|---|
| **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 | |

In [ ]:
```
#left is output variable
```

In [53]:
```
from sklearn.preprocessing import LabelEncoder
```

In [54]:
```
le = LabelEncoder()
```

In [55]:
```
#convert the categorical into integer
df['Department'] = le.fit_transform(df['Department'])
df['salary'] = le.fit_transform(df['salary'])
```

In [56]:
```
df.head()
```

Out[56]:

| satisfaction_level | last_evaluation | number_project | average_montly_hours | time_spend_company | Work_ |
|---|---|---|---|---|---|
| 0.38 | 0.53 | 2 | 157 | 3 | |
| 0.80 | 0.86 | 5 | 262 | 6 | |
| 0.11 | 0.88 | 7 | 272 | 4 | |
| 0.72 | 0.87 | 5 | 223 | 5 | |
| 0.37 | 0.52 | 2 | 159 | 3 | |

In [59]:
```
X = df[['satisfaction_level',
        'last_evaluation',
        'number_project',
        'average_montly_hours',
        'time_spend_company',
        'Work_accident',
        'promotion_last_5years',
        'Department',
        'salary']]
y = df['left']
```

In [60]:
```
df.shape
```

Out[60]: (14999, 10)

In [61]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(X,y,test_size=0.25,random_state=1)
```

In [62]:
```
xtrain.shape
```

Out[62]: (11249, 9)

In [63]:
```
xtest.shape
```

Out[63]: (3750, 9)

In [65]:
```
from sklearn.neural_network import MLPClassifier
```

In [66]:
```
clf = MLPClassifier(hidden_layer_sizes=(6,5),random_state=5,verbose=True,learning_
#two hidden layer1 with 6 neurons then after that hidden layer2 with 5 neurons
#verbose is for detail
```

In [67]:
```
clf.fit(xtrain,ytrain)
```

```
Iteration 1, loss = 0.61173563
Iteration 2, loss = 0.56610705
Iteration 3, loss = 0.53379109
Iteration 4, loss = 0.50526290
Iteration 5, loss = 0.49305592
Iteration 6, loss = 0.49090774
Iteration 7, loss = 0.44190654
Iteration 8, loss = 0.39038250
Iteration 9, loss = 0.35495330
Iteration 10, loss = 0.34197132
Iteration 11, loss = 0.32906195
Iteration 12, loss = 0.31401783
Iteration 13, loss = 0.30434400
Iteration 14, loss = 0.30497620
Iteration 15, loss = 0.31951860
Iteration 16, loss = 0.29290617
Iteration 17, loss = 0.29172226
Iteration 18, loss = 0.30686365
Iteration 19, loss = 0.30512820
Iteration 20, loss = 0.28846803
Iteration 21, loss = 0.32059201
Iteration 22, loss = 0.29769396
Iteration 23, loss = 0.28881736
Iteration 24, loss = 0.28540207
Iteration 25, loss = 0.27706584
Iteration 26, loss = 0.28757267
Iteration 27, loss = 0.29055327
Iteration 28, loss = 0.28974985
Iteration 29, loss = 0.29017120
Iteration 30, loss = 0.27737603
Iteration 31, loss = 0.28531902
Iteration 32, loss = 0.28806741
Iteration 33, loss = 0.27121425
Iteration 34, loss = 0.28533661
Iteration 35, loss = 0.28440099
Iteration 36, loss = 0.27690615
Iteration 37, loss = 0.27832869
Iteration 38, loss = 0.26980603
Iteration 39, loss = 0.26234432
Iteration 40, loss = 0.28249958
Iteration 41, loss = 0.26612812
Iteration 42, loss = 0.26467398
Iteration 43, loss = 0.26874733
Iteration 44, loss = 0.25919635
Iteration 45, loss = 0.26742460
Iteration 46, loss = 0.27422672
Iteration 47, loss = 0.25815315
Iteration 48, loss = 0.26078831
Iteration 49, loss = 0.26390934
Iteration 50, loss = 0.25259498
Iteration 51, loss = 0.24276198
Iteration 52, loss = 0.25159396
Iteration 53, loss = 0.25434856
Iteration 54, loss = 0.25362808
Iteration 55, loss = 0.24292661
Iteration 56, loss = 0.24618863
Iteration 57, loss = 0.24236200
Iteration 58, loss = 0.24161522
```

```
Iteration 59, loss = 0.25516133
Iteration 60, loss = 0.26422890
Iteration 61, loss = 0.25954281
Iteration 62, loss = 0.25083415
Iteration 63, loss = 0.26494829
Iteration 64, loss = 0.26258953
Iteration 65, loss = 0.24764567
Iteration 66, loss = 0.26467657
Iteration 67, loss = 0.26376716
Iteration 68, loss = 0.23958408
Iteration 69, loss = 0.25631213
Iteration 70, loss = 0.26436764
Iteration 71, loss = 0.26443774
Iteration 72, loss = 0.25124572
Iteration 73, loss = 0.25405088
Iteration 74, loss = 0.25266038
Iteration 75, loss = 0.23740123
Iteration 76, loss = 0.23772879
Iteration 77, loss = 0.23547121
Iteration 78, loss = 0.24382808
Iteration 79, loss = 0.24006399
Iteration 80, loss = 0.26110504
Iteration 81, loss = 0.24146286
Iteration 82, loss = 0.25332370
Iteration 83, loss = 0.25696777
Iteration 84, loss = 0.24211336
Iteration 85, loss = 0.25107179
Iteration 86, loss = 0.24027004
Iteration 87, loss = 0.24241551
Iteration 88, loss = 0.23653858
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. S
topping.
```

Out[67]:
```
MLPClassifier(hidden_layer_sizes=(6, 5), learning_rate_init=0.01,
              random_state=5, verbose=True)
```

In [68]:
```
predictions = clf.predict(xtest)
```

In [70]:
```
from sklearn.metrics import accuracy_score
print('Accuracy of the multilayer perceptron is',accuracy_score(ytest,predictions)
```

```
Accuracy of the multilayer perceptron is 0.9117333333333333
```

In [ ]:
```

```