pract-mlp

June 7, 2023

```
[15]: import numpy as np
      import pandas as pd
      import matplotlib.pyplot as plt
      from sklearn.metrics import classification_report,confusion_matrix
      from sklearn import preprocessing
```

```
[2]: data=pd.read_csv('HR_comma_sep.csv')
```

```
[3]: print(data.head())
```

```
   satisfaction_level  last_evaluation  number_project  average_montly_hours  \
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

   time_spend_company  Work_accident  left  promotion_last_5years  sales  \
0                   3              0     1                      0  sales
1                   6              0     1                      0  sales
2                   4              0     1                      0  sales
3                   5              0     1                      0  sales
4                   3              0     1                      0  sales

   salary
0     low
1  medium
2  medium
3     low
4     low
```

```
[4]: print("Descriptive Stats:\n",data.describe())
```

```
Descriptive Stats:
        satisfaction_level  last_evaluation  number_project  \
count         14999.000000     14999.000000    14999.000000
mean              0.612834         0.716102        3.803054
std               0.248631         0.171169        1.232592
```

```
min              0.090000         0.360000        2.000000
25%              0.440000         0.560000        3.000000
50%              0.640000         0.720000        4.000000
75%              0.820000         0.870000        5.000000
max              1.000000         1.000000        7.000000

       average_montly_hours  time_spend_company  Work_accident          left  \
count          14999.000000        14999.000000   14999.000000  14999.000000
mean             201.050337            3.498233       0.144610      0.238083
std               49.943099            1.460136       0.351719      0.425924
min               96.000000            2.000000       0.000000      0.000000
25%              156.000000            3.000000       0.000000      0.000000
50%              200.000000            3.000000       0.000000      0.000000
75%              245.000000            4.000000       0.000000      0.000000
max              310.000000           10.000000       1.000000      1.000000

       promotion_last_5years
count           14999.000000
mean                0.021268
std                 0.144281
min                 0.000000
25%                 0.000000
50%                 0.000000
75%                 0.000000
max                 1.000000
```

[5]:

```
data.head()
```

[5]:

```
   satisfaction_level  last_evaluation  number_project  average_montly_hours  \
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

   time_spend_company  Work_accident  left  promotion_last_5years  sales  \
0                   3              0     1                      0  sales
1                   6              0     1                      0  sales
2                   4              0     1                      0  sales
3                   5              0     1                      0  sales
4                   3              0     1                      0  sales

   salary
0     low
1  medium
2  medium
3     low
```

```
4     low
```

[6]:
```
le=preprocessing.LabelEncoder()
data['sales']=le.fit_transform(data['sales'])
data['salary']=le.fit_transform(data['salary'])
```

[7]:
```
data.head()
```

[7]:
```
   satisfaction_level  last_evaluation  number_project  average_montly_hours  \
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

   time_spend_company  Work_accident  left  promotion_last_5years  sales  \
0                   3              0     1                      0      7
1                   6              0     1                      0      7
2                   4              0     1                      0      7
3                   5              0     1                      0      7
4                   3              0     1                      0      7

   salary
0       1
1       2
2       2
3       1
4       1
```

[8]:
```
x=data.drop('left',axis=1)
y=data['left']
```

[9]:
```
from sklearn.model_selection import train_test_split
x_test,x_train,y_test,y_train=train_test_split(x,y,test_size=0.
 ↪3,random_state=42)
```

[12]:
```
from sklearn.neural_network import MLPClassifier
model=MLPClassifier(hidden_layer_sizes=(6,5),random_state=5,verbose=True,learning_rate_init=0.
 ↪01)
model.fit(x_train,y_train)
```

```
Iteration 1, loss = 0.66595201
Iteration 2, loss = 0.58677261
Iteration 3, loss = 0.57998329
Iteration 4, loss = 0.57311768
Iteration 5, loss = 0.56793123
Iteration 6, loss = 0.55674154
Iteration 7, loss = 0.55032400
Iteration 8, loss = 0.53818180
Iteration 9, loss = 0.52460225
Iteration 10, loss = 0.51554002
Iteration 11, loss = 0.50400107
Iteration 12, loss = 0.49904158
Iteration 13, loss = 0.49314475
Iteration 14, loss = 0.48229725
Iteration 15, loss = 0.47636624
Iteration 16, loss = 0.47590910
Iteration 17, loss = 0.48135799
Iteration 18, loss = 0.49589865
Iteration 19, loss = 0.46484377
Iteration 20, loss = 0.48635463
Iteration 21, loss = 0.46497716
Iteration 22, loss = 0.46379125
Iteration 23, loss = 0.46816840
Iteration 24, loss = 0.46153633
Iteration 25, loss = 0.46347898
Iteration 26, loss = 0.46163912
Iteration 27, loss = 0.45668105
Iteration 28, loss = 0.46873997
Iteration 29, loss = 0.46360303
Iteration 30, loss = 0.45689111
Iteration 31, loss = 0.47150452
Iteration 32, loss = 0.47122201
Iteration 33, loss = 0.47003000
Iteration 34, loss = 0.45802161
Iteration 35, loss = 0.45870173
Iteration 36, loss = 0.46343473
Iteration 37, loss = 0.46003687
Iteration 38, loss = 0.47448930
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs.
Stopping.
```

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

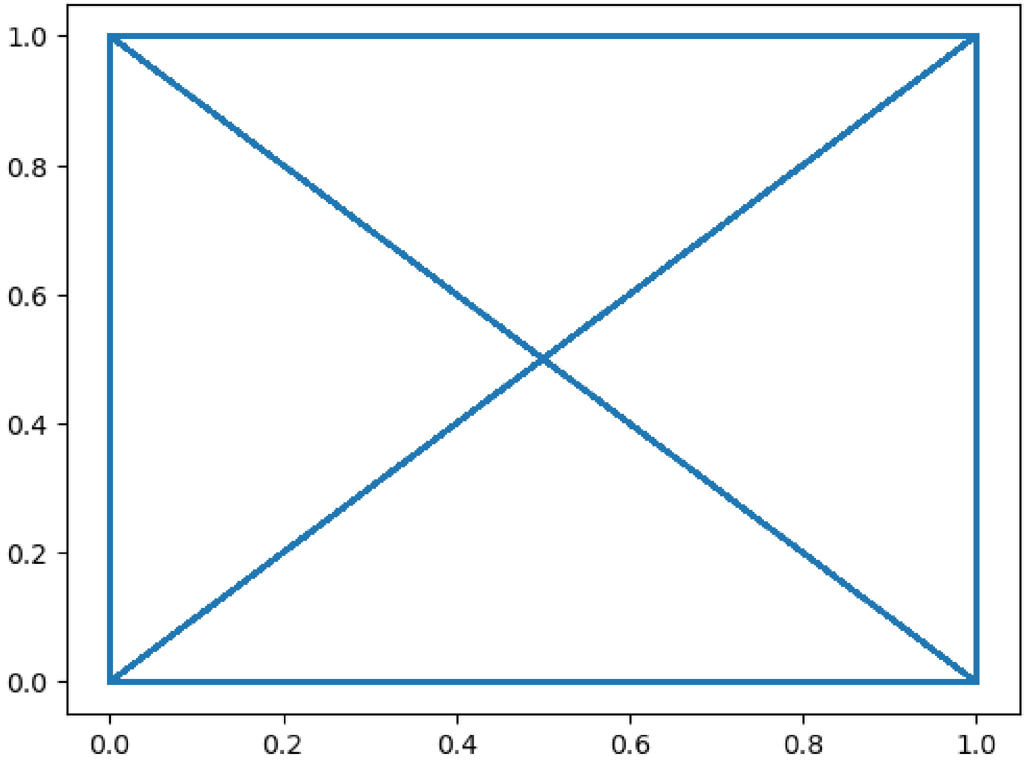
```
[14]: y_pred=model.predict(x_test)
      print("Classification Results:/n",classification_report(y_test,y_pred))
```

```
Classification Results:/n                precision    recall  f1-score   support

           0       0.80      0.94      0.86      8000
           1       0.56      0.24      0.34      2499

    accuracy                           0.77     10499
   macro avg       0.68      0.59      0.60     10499
```

```
weighted avg       0.74      0.77      0.74     10499
```

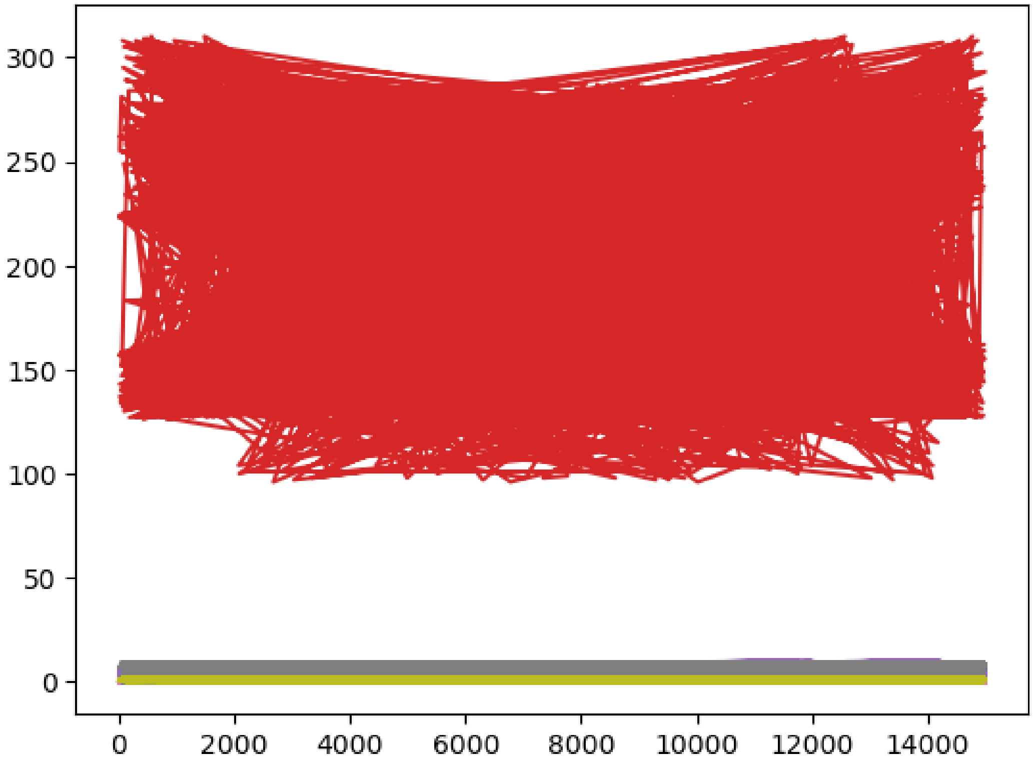
[16]:
```
plt.plot(y_test,y_pred)
plt.show
```

[16]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



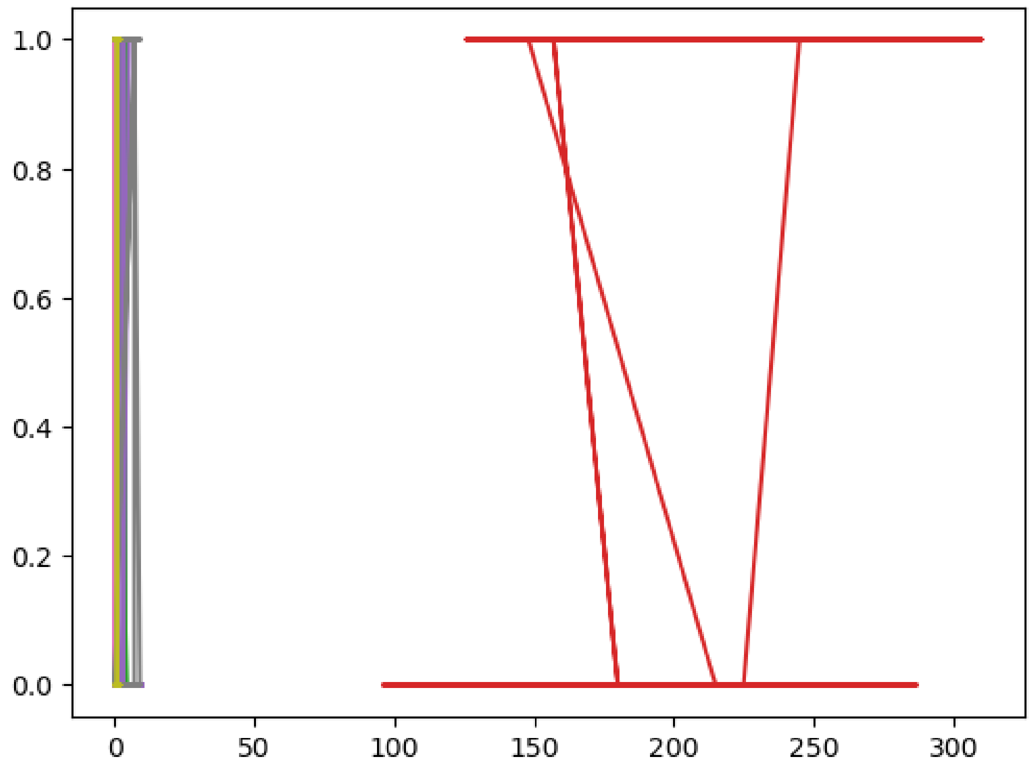
[17]:
```
plt.plot(x_train)
plt.show
```

[17]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



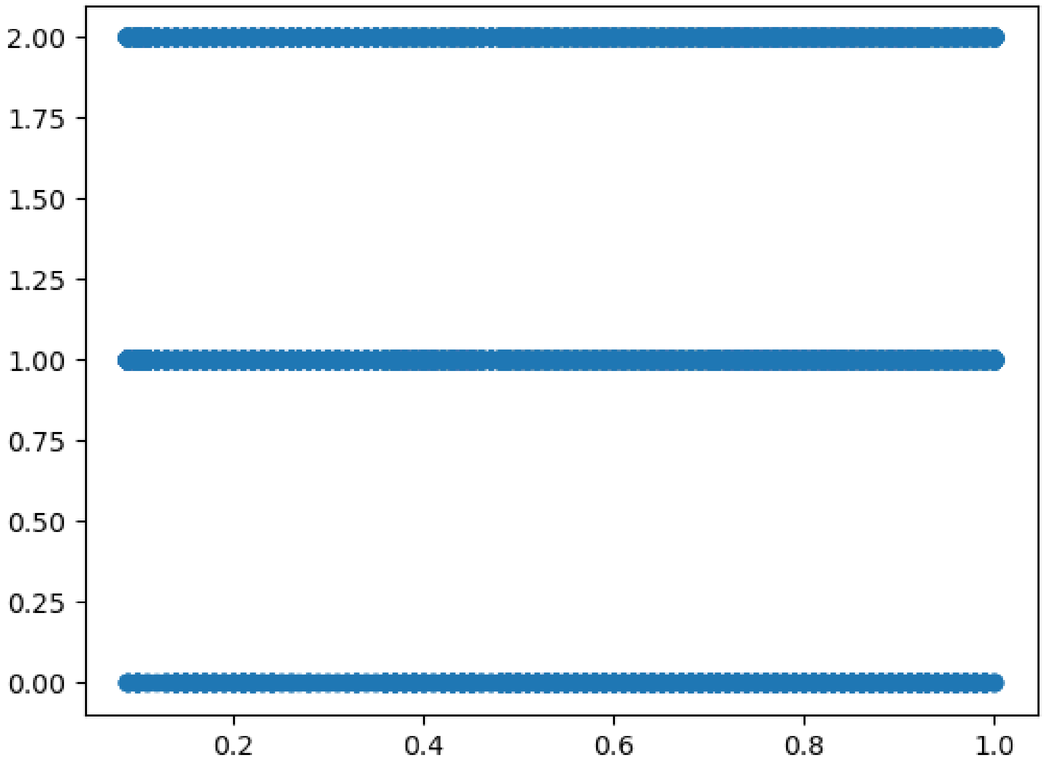
[18]:
```
plt.plot(x,y)
plt.show
```

[18]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
[22]: plt.scatter(x['satisfaction_level'],x['salary'])
      plt.show
```

```
[22]: <function matplotlib.pyplot.show(close=None, block=None)>
```


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.2
0.4
0.6
0.8
1.0

[ ]:

[ ]:

[ ]: