

In [8]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [9]:

```
data = pd.read_csv('pred_data.csv')
data.shape
```

Out[9]:

```
(10100, 2)
```

In [10]:

```
data.head()
```

Out[10]:

	y	proba
0	0.0	0.281035
1	0.0	0.465152
2	0.0	0.352793
3	0.0	0.157818
4	0.0	0.276648

In [11]:

```
data['y'].value_counts()
```

Out[11]:

```
0.0    10000
1.0      100
Name: y, dtype: int64
```

In [13]:

```
data['y_pred'] = data['proba'].apply(lambda p: 1 if p>=0.5 else 0)
```

In [14]:

```
data.head()
```

Out[14]:

	y	proba	y_pred
0	0.0	0.281035	0
1	0.0	0.465152	0
2	0.0	0.352793	0
3	0.0	0.157818	0
4	0.0	0.276648	0

In [16]:

```
tp = ((data['y'] ==1) & (data['y_pred']==1)).sum()  
tp  
  
# hw. -> fp, tn, tn
```

Out[16]:

55

In [17]:

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

In [18]:

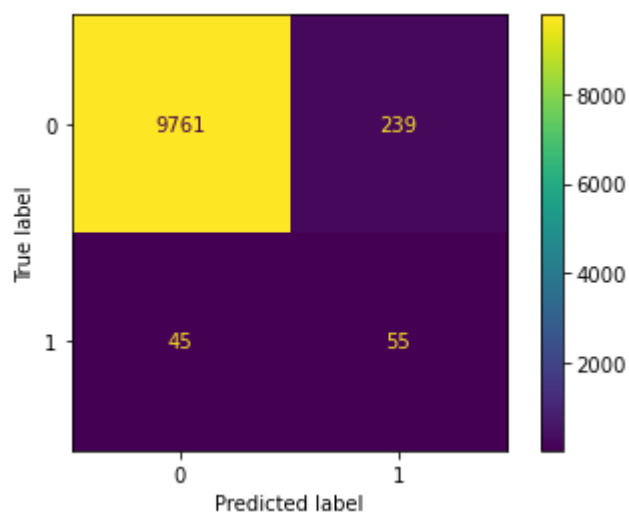
```
cm = confusion_matrix(data['y'], data['y_pred'])  
cm
```

Out[18]:

```
array([[9761, 239],  
       [ 45, 55]])
```

In [19]:

```
ConfusionMatrixDisplay(cm).plot();
```



In [22]:

```
cm.diagonal().sum()/cm.sum()
```

Out[22]:

0.9718811881188119

Precision

In [23]:

```
55 / (55+239)
```

Out[23]:

0.1870748299319728

In [24]:

```
from sklearn.metrics import precision_score
```

In [25]:

```
precision_score(data['y'], data['y_pred'])
```

Out[25]:

0.1870748299319728

In []:

Recall

In [26]:

```
55 / (45 + 55)
```

Out[26]:

0.55

In [27]:

```
from sklearn.metrics import recall_score
```

In [28]:

```
recall_score(data['y'], data['y_pred'])
```

Out[28]:

0.55

In []:

In []:

In []:

F1 score

In [29]:

```
0.7*0.7*2/(0.7+0.7)
```

Out[29]:

0.7

In [30]:

```
2*0.9*0.5/(0.9+0.5)
```

Out[30]:

0.6428571428571429

In [32]:

```
re = recall_score(data['y'], data['y_pred'])
pr = precision_score(data['y'], data['y_pred'])

f1 = 2*pr*re/(pr+re)
print(f1.round(2))
```

0.28

In [33]:

```
from sklearn.metrics import f1_score
```

In [35]:

```
f1_score(data['y'], data['y_pred']).round(2)
```

Out[35]:

0.28

In []:

In []:

ROC

In [49]:

```
np.random.randint(0, 2, 100)
```

Out[49]:

```
array([1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1,
1,
      0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1,
1,
      0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1,
1,
      1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0,
0,
      0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1])
```

In [51]:

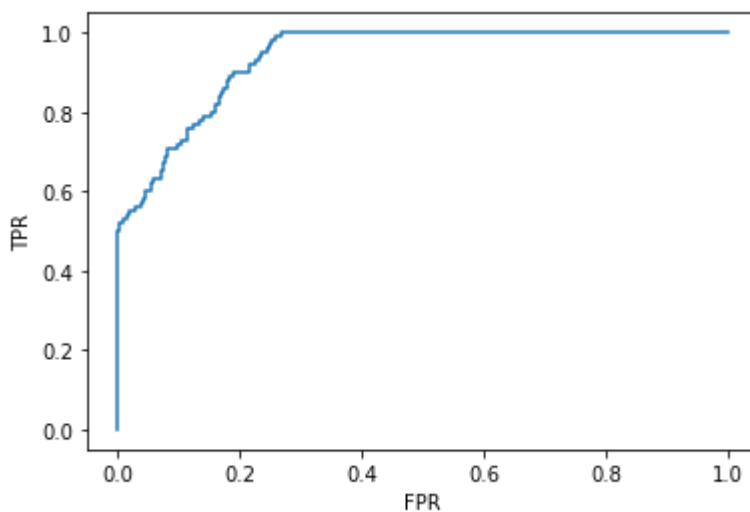
```
from sklearn.metrics import roc_curve, roc_auc_score
```

In [52]:

```
fpr, tpr, thres = roc_curve(data['y'], data['proba'])
```

In [55]:

```
plt.xlabel("FPR")
plt.ylabel("TPR")
plt.plot(fpr, tpr)
plt.show()
```



In [56]:

```
roc_auc_score(data['y'], data['proba'])
```

Out[56]:

```
0.9377570000000001
```

In [58]:

```
# thres
```

In [59]:

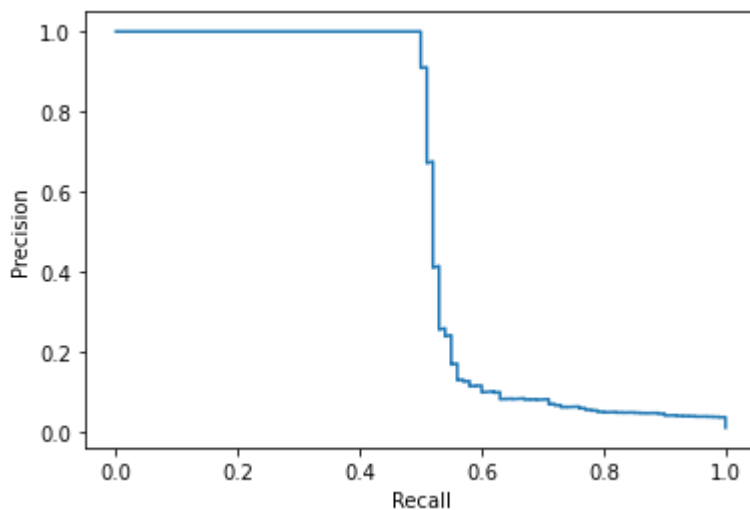
```
from sklearn.metrics import precision_recall_curve, auc
```

In [60]:

```
pr, re, thres = precision_recall_curve(data['y'], data['proba'])
```

In [62]:

```
plt.xlabel("Recall")  
plt.ylabel("Precision")  
plt.plot(re, pr)  
plt.show()
```



In [63]:

```
auc(re, pr)
```

Out[63]:

```
0.554765146410734
```

In [67]:

```
data['random_prob'] = np.random.rand(10100)
```

In [68]:

```
data.head()
```

Out[68]:

	y	proba	y_pred	random_prob
0	0.0	0.281035	0	0.088962
1	0.0	0.465152	0	0.768248
2	0.0	0.352793	0	0.038002
3	0.0	0.157818	0	0.737514
4	0.0	0.276648	0	0.372516

In [72]:

```
# np.array(list((zip(re, pr, thres))))
```

Out[72]:

```
array([[1.          , 0.00990099, 0.10000141],
       [1.          , 0.00990197, 0.1001608 ],
       [1.          , 0.00990295, 0.10016508],
       ...,
       [0.03        , 1.          , 0.59219787],
       [0.02        , 1.          , 0.5948084 ],
       [0.01        , 1.          , 0.59529418]])
```

In []: