import the libraries required

In [1]:

```
import numpy as np
import pandas as pd
```

read the csv file

In [2]:

```
a_csv=pd.read_csv('5_b.csv')
print(a_csv.head(10))
a_csv.shape
```

```
     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
5  0.0  0.190260
6  0.0  0.320328
7  0.0  0.435013
8  0.0  0.284849
9  0.0  0.427919
```

Out[2]:

```
(10100, 2)
```

change the prob values to binary values.using thresholds.

In [3]:

```
df=a_csv['proba']
```

In [4]:

```
print(len(df))
```

```
10100
```

convert probabilistic values to either 0 or 1 based on threshold

In [5]:

```
count=0
count1=0
for i in range(0,len(df)):
    if df.loc[i]>0.5:
        count+=1
        df=df.replace(df.loc[i],1)
    elif df.loc[i]<=0.5:
        count1+=1
        df=df.replace(df.loc[i],0)
print(df.head(10))
print('positive points count',count)
print('positive points count',count1)
```

```
0    0.0
1    0.0
2    0.0
3    0.0
4    0.0
5    0.0
6    0.0
7    0.0
8    0.0
9    0.0
Name: proba, dtype: float64
positive points count 294
positive points count 9806
```

In [6]:

```
df.describe()
```

Out[6]:

```
count    10100.000000
mean         0.029109
std          0.168120
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          1.000000
Name: proba, dtype: float64
```

create column in the dataset to store yhat values

In [7]:

```
a_csv['yhat']=df.values
```

In [8]:

```
a_csv.head(10)
```

Out[8]:

	y	proba	yhat
0	0.0	0.281035	0.0
1	0.0	0.465152	0.0
2	0.0	0.352793	0.0
3	0.0	0.157818	0.0
4	0.0	0.276648	0.0
5	0.0	0.190260	0.0
6	0.0	0.320328	0.0
7	0.0	0.435013	0.0
8	0.0	0.284849	0.0
9	0.0	0.427919	0.0

calculate total number of positive and negative points in the dataset

In [9]:

```
pos=0
neg=0
def sudhi():
    global pos
    global neg
    for i in range(0,len(a_csv)):
        if a_csv['y'].loc[i]==1:
            pos+=1

        elif a_csv['y'].loc[i]==0:
            neg+=1
    return pos,neg
sudhi()
print('positive points count(p)',pos)
print('negative points count(n)',neg)
```

```
positive points count(p) 100
negative points count(n) 10000
```

In [10]:

```
print(pos)
print(neg)
```

```
100
10000
```

CONFUSION MATRIX CALCULATION

```
In [12]:
```

```
confusion_matrix=[]
TN=0
FN=0
FP=0
TP=0
for i in range(0,len(a_csv)):
    global TN
    global TP
    global FP
    global FN
    if a_csv['y'].loc[i]==a_csv['yhat'].loc[i]:

        if a_csv['y'].loc[i]==0:

            TN+=1

        elif a_csv['y'].loc[i]==1:

            TP+=1

    elif a_csv['y'].loc[i]!=a_csv['yhat'].loc[i]:

        if a_csv['y'].loc[i]==0 and a_csv['yhat'].loc[i]==1:
            FP+=1
        elif a_csv['y'].loc[i]==1 and a_csv['yhat'].loc[i]==0:
            FN+=1

print('TP',TP)
print('TN',TN)
print('FP',FP)
print('FN',FN)
confusion_matrix.append(TN)
confusion_matrix.append(FN)
confusion_matrix.append(FP)
confusion_matrix.append(TP)

x=np.reshape(confusion_matrix,(2, 2))
#print(confusion_matrix)
print('CONFUSION MATRIX \n',x)
```

```
TP 55
TN 9761
FP 239
FN 45
CONFUSION MATRIX
 [[9761   45]
 [ 239   55]]
```

PRECISION , RECALL , F1-SCORE calculation

In [13]:

```
precision=((TP)/(TP+FP))
recall=((TP)/(TP+FN))
print('precision \n',precision)
print('recall \n',recall)
F1_score=2*(precision*recall)/(precision+recall)
print('F1-score \n',F1_score)
```

```
precision
 0.1870748299319728
recall
 0.55
F1-score
 0.2791878172588833
```

ACCURACY SCORE

In [14]:

```
accuracy_score=(TP+TN)/(TP+FP+FN+TN)
print('accuracy score \n',accuracy_score)
```

```
accuracy score
 0.9718811881188119
```

COMPUTE AUC by considering each probability value in yhat as threshold and compute TPR and FPR

In [15]:

```
tpr_lst=[]
fpr_lst=[]
fn_lst=[]
tn_lst=[]
from tqdm import tqdm
a_csv=pd.read_csv('5_b.csv')
#a_csv['proba']=sorted(a_csv['proba'])
sorted_data=a_csv.sort_values('proba',ascending=True)
for threshold in tqdm(sorted_data['proba']):
    y_hat=[]
    for value in sorted_data['proba']:
        if (value<=threshold):
            y_hat.append(0.0)
        else:
            y_hat.append(1.0)
    #print(y_hat[:10])
    sorted_data['y_pred']=y_hat
    #print(a_csv.head(10))
    for k in a_csv:
        tp = (((sorted_data['y'])==1.0) & ((sorted_data['y_pred']) == 1.0)).sum()
        fp = (((sorted_data['y'])==0.0) & ((sorted_data['y_pred']) == 1.0)).sum()
        tn=(((sorted_data['y'])==0.0) & ((sorted_data['y_pred']) == 0.0)).sum()
        fn=(((sorted_data['y'])==1.0) & ((sorted_data['y_pred']) == 0.0)).sum()

    tpr_lst.append(tp/(tp+fn))
    fpr_lst.append(fp/(fp+tn))
    #print('TPR\n',tpr_lst[:])

    #print('FPR\n',fpr_lst[:])

    tn_lst.append(tn)
    fn_lst.append(fn)
x=sorted(tpr_lst)
y=sorted(fpr_lst)
auc = np.trapz(x,y)

print('AUC score is : ',auc)
print('TP',tp)
print('FP',fp)
print('TN',tn)
print('FN',fn)
```

```
100%|██████████████████████████████████████████████████████████████████████
██████| 10100/10100 [01:29<00:00, 112.42it/s]

AUC score is :  0.9376570000000001
TP 0
FP 0
TN 10000
FN 100
```