8E_F_LR_SVM

January 28, 2020

9E and 9F: Finding the Probability P(Y==1|X)

9E: Implementing Decision Function of SVM RBF Kernel

After we train a kernel SVM model, we will be getting support vectors and their corresponsing coefficients α_i

Check the documentation for better understanding of these attributes:

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

As a part of this assignment you will be implementing the `decision_function()` of kernel SVM, here decision_function() means based on the value return by `decision_function()` model will classify the data point either as positive or negative

Ex 1: In logistic regression After traning the models with the optimal weights w we get, we will find the value $\frac{1}{1+\exp(-(wx+b))}$, if this value comes out to be < 0.5 we will mark it as negative class, else its positive class

Ex 2: In Linear SVM After traning the models with the optimal weights w we get, we will find the value of $sign(wx+b)$, if this value comes out to be -ve we will mark it as negative class, else its positive class.

Similarly in Kernel SVM After traning the models with the coefficients α_i we get, we will find the value of $sign(\sum_{i=1}^{n}(y_i\alpha_i K(x_i, x_q)) + intercept)$, here $K(x_i, x_q)$ is the RBF kernel. If this value comes out to be -ve we will mark x_q as negative class, else its positive class.

RBF kernel is defined as: $K(x_i, x_q) = exp(-\gamma||x_i - x_q||^2)$

For better understanding check this link: https://scikit-learn.org/stable/modules/svm.html#svm-mathematical-formulation

0.1 Task E

1. Split the data into X_{train}(60), X_{cv}(20), X_{test}(20)
2. Train $SVC(gamma = 0.001, C = 100.)$ on the (X_{train}, y_{train})
3. Get the decision boundry values f_{cv} on the X_{cv} data i.e. f_{cv} = `decision_function`(X_{cv}) you need to implement this decision_function()

```
In [1]: import numpy as np
        import pandas as pd
        from sklearn.datasets import make_classification
        from sklearn.model_selection import train_test_split
        from sklearn.svm import SVC
```

```
In [2]: X, y = make_classification(n_samples=5000, n_features=5, n_redundant=2,
                                   n_classes=2, weights=[0.7], class_sep=0.7, random_state=15)
```

0.1.1 Pseudo code

clf = SVC(gamma=0.001, C=100.) clf.fit(Xtrain, ytrain)

def decision_function(Xcv, ...): #use appropriate parameters for a data point x_q in Xcv: #write code to implement $(\sum_{i=1}^{\text{all the support vectors}}(y_i\alpha_i K(x_i, x_q)) + intercept)$, here the values y_i, α_i, and *intercept* can be obtained from the trained model return # the decision_function output for all the data points in the Xcv

fcv = decision_function(Xcv, ...) # based on your requirement you can pass any other parameters

Note: Make sure the values you get as fcv, should be equal to outputs of clf.decision_function(Xcv)

In [3]:
```
# you can write your code here
data = np.hstack((X, y.reshape(5000,1)))
data.shape
```

Out[3]:
```
(5000, 6)
```

In [4]:
```
X_train,X_rem,Y_train,Y_rem = train_test_split(X,y,test_size=0.40,random_state=16)
X_cv,X_test,Y_cv,Y_test = train_test_split(X_rem,Y_rem,test_size=0.50,random_state=16)

X_train.shape,X_cv.shape,X_test.shape
```

Out[4]:
```
((3000, 5), (1000, 5), (1000, 5))
```

In [5]:
```
model = SVC(gamma = 0.001, C = 100)
model.fit(X_train,Y_train)
```

Out[5]:
```
SVC(C=100, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

In [6]:
```
f_cv = model.decision_function(X_cv)
# f_cv
```

In [7]:
```
def calculate_rbf(X_i,X_q):
#     k = np.exp(-model.gamma * np.linalg.norm(X_i-X_q))
    k = np.exp(-model.gamma * np.dot(X_i-X_q, X_i-X_q))
    return k
```

In [8]:
```
from tqdm import tqdm

def custom_decision_function(Xcv): #use appropriate parameters
    ycv = np.zeros(Xcv.shape[0]).reshape((Xcv.shape[0],1))
    for i in tqdm(range(Xcv.shape[0])):
        sum = 0
        for j in range(model.support_vectors_.shape[0]):
            kernal = calculate_rbf(model.support_vectors_[j],Xcv[i])
            sum += kernal*model.dual_coef_[0][j]
        ycv[i] = sum + model.intercept_
    return ycv
```

```
In [9]: f_cv_custom = custom_decision_function(X_cv)
        f_test_custom = custom_decision_function(X_test)

100%|| 1000/1000 [00:05<00:00, 186.36it/s]
100%|| 1000/1000 [00:04<00:00, 206.61it/s]


In [10]: f_cv_custom - f_cv.reshape(1000,1)

Out[10]: array([[ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
                [ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[-1.13686838e-13],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 4.26325641e-14],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 1.42108547e-14],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 5.68434189e-14],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 3.63797881e-12],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[-4.54747351e-13],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 3.63797881e-12],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[-2.27373675e-13],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[-1.42108547e-14],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 1.42108547e-14],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 5.68434189e-14],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 4.54747351e-13],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 1.42108547e-14],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 2.27373675e-13],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
[ 0.00000000e+00],
```

```
[ 0.00000000e+00]])
```

9F: Implementing Platt Scaling to find P(Y==1|X)
Check this PDF

0.2 TASK F

4. Apply SGD algorithm with (f_{cv}, y_{cv}) and find the weight W intercept b `Note: here our data is of one dimensional so we will have a one dimensional weight vector i.e W.shape (1,)`

Note1: Don't forget to change the values of y_{cv} as mentioned in the above image. you will calculate y+, y- based on data points in train data

Note2: the Sklearn's SGD algorithm doesn't support the real valued outputs, you need to use the code that was done in the `'Logistic Regression with SGD and L2'` Assignment after modifying loss function, and use same parameters that used in that assignment. if Y[i] is 1, it will be replaced with y+ value else it will replaced with y- value

5. For a given data point from X_{test}, $P(Y=1|X) = \frac{1}{1+exp(-(W*f_{test}+b))}$ where f_{test} `= decision_function(`X_{test}`)`, W and b will be learned as metioned in the above step

Note: in the above algorithm, the steps 2, 4 might need hyper parameter tuning, To reduce the complexity of the assignment we are excluding the hyerparameter tuning part, but intrested students can try that

If any one wants to try other calibration algorithm istonic regression also please check these tutorials

1. http://fa.bianp.net/blog/tag/scikit-learn.html#fn:1
2. https://drive.google.com/open?id=1MzmA7QaP58RDzocB0RBmRiWfl7Co_VJ7
3. https://drive.google.com/open?id=133odBinMOIVb_rh_GQxxsyMRyW-Zts7a
4. https://stat.fandom.com/wiki/Isotonic_regression#Pool_Adjacent_Violators_Algorithm

In [11]:
```
N_positive = 0
N_negative = 0
for i in range(len(Y_cv)):
    if Y_cv[i] > 0:
        N_positive +=1
    else:
        N_negative +=1
y_positive = (N_positive + 1) / (N_positive + 2)
y_negative = 1/(N_negative + 2)

N_positive,N_negative,y_positive,y_negative
```

Out[11]:
```
(305, 695, 0.996742671009772, 0.0014347202295552368)
```

In [12]: f_cv_custom[0]

Out[12]: array([-1.80554548])

In [13]:
```
import math

def sigmoid(x,w,b):
    a = np.dot(x,w)+b
    return 1/(1 + math.exp(-a))
```

In [14]:
```
class CustomSGDClassifier:
    def __init__(self,alpha = 0.0001,N = len(X_train),b = 0,w = np.zeros_like(X_train
        self.alpha = alpha
        self.N = N
        self.b = b
        self.w = w
        self.eta0 = eta0
        self.ephocs = ephocs
        self.train_loss = []
        self.test_loss = []
        self.result = []

    def calculateLoss(self,w,b,X,Y):
        loss = []
        for i in np.arange(0,len(X)):
            sig = sigmoid(X[i],w,b)
            if Y[i] > 0:
                loss.append(-y_positive*np.log(sig) - (1-y_positive)*np.log(1-sig) + s
            else:
                loss.append(-y_negative*np.log(sig) - (1-y_negative)*np.log(1-sig) + s
        return np.mean(loss)

    def fit(self,X,Y):
        initial_loss = self.calculateLoss(w,b,X,Y)
#         print('Initial Loss:',initial_loss)
        for ep in range(self.ephocs):
            for i in range(N):
                sig = sigmoid(X[i],self.w,self.b)
                w_new = (1- (self.alpha*self.eta0)/self.N)*self.w + self.eta0*X[i]*(Y
                b_new = self.b + self.eta0*(Y[i]-sig)
                self.w = w_new
                self.b = b_new
            next_loss = self.calculateLoss(self.w,self.b,X,Y)
            self.train_loss.append(next_loss)
            print('-- Epoch: {}, Avg. Train Loss: {}'.format(ep+1,next_loss))
            if (next_loss < initial_loss) & ((initial_loss-next_loss)<0.0001):
                break
            initial_loss = next_loss
```

```
    def getProbability(self,X):
        res = []
        for i in range(len(X)):
            prob = 1/(1+math.exp(-(self.w*X[i]+self.b)))
            res.append(prob)
        return res
```

In [17]:
```
w = np.zeros_like(f_cv_custom[0])
b = 0
eta0  = 0.0001
alpha = 0.0001
N = len(f_cv_custom)
# f_test_custom = custom_decision_function(X_test)

model = CustomSGDClassifier(alpha,N,b,w,eta0,ephocs=30)
%time model.fit(f_cv_custom,Y_cv)
```

```
-- Epoch: 1, Avg. Train Loss: 0.6079155706248737
-- Epoch: 2, Avg. Train Loss: 0.5432836790075528
-- Epoch: 3, Avg. Train Loss: 0.49358708840014864
-- Epoch: 4, Avg. Train Loss: 0.4547003818858166
-- Epoch: 5, Avg. Train Loss: 0.42371468946471313
-- Epoch: 6, Avg. Train Loss: 0.39859299715826124
-- Epoch: 7, Avg. Train Loss: 0.3778994410869025
-- Epoch: 8, Avg. Train Loss: 0.3606085361894483
-- Epoch: 9, Avg. Train Loss: 0.3459761556072111
-- Epoch: 10, Avg. Train Loss: 0.33345327076465753
-- Epoch: 11, Avg. Train Loss: 0.3226281158110312
-- Epoch: 12, Avg. Train Loss: 0.3131870226728602
-- Epoch: 13, Avg. Train Loss: 0.30488754386278194
-- Epoch: 14, Avg. Train Loss: 0.2975397344274488
-- Epoch: 15, Avg. Train Loss: 0.29099291865545984
-- Epoch: 16, Avg. Train Loss: 0.28512619511616644
-- Epoch: 17, Avg. Train Loss: 0.2798415264456194
-- Epoch: 18, Avg. Train Loss: 0.27505864189224627
-- Epoch: 19, Avg. Train Loss: 0.2707112289054954
-- Epoch: 20, Avg. Train Loss: 0.2667440535831429
-- Epoch: 21, Avg. Train Loss: 0.2631107589204572
-- Epoch: 22, Avg. Train Loss: 0.2597721635902968
-- Epoch: 23, Avg. Train Loss: 0.2566949345238134
-- Epoch: 24, Avg. Train Loss: 0.2538505416185785
-- Epoch: 25, Avg. Train Loss: 0.2512144275145971
-- Epoch: 26, Avg. Train Loss: 0.248765342861617
-- Epoch: 27, Avg. Train Loss: 0.24648481005728792
-- Epoch: 28, Avg. Train Loss: 0.24435668754887577
-- Epoch: 29, Avg. Train Loss: 0.2423668134719499
-- Epoch: 30, Avg. Train Loss: 0.2405027123436499
```

```
Wall time: 1.88 s
```

```
In [18]: model.w,model.b
```

```
Out[18]: (array([0.98423654]), -0.11499080123800826)
```

```
In [23]: prob_test = model.getProbability(f_test_custom)
         prob_test
```

```
Out[23]: [0.025769892876980352,
          0.6063125935736543,
          0.8153664847849584,
          0.862127047261603,
          0.07381803964467702,
          0.07326255570560314,
          0.2543876718566009,
          0.03781039153996189,
          0.022470991228961575,
          0.22615581445987729,
          0.82948801248056,
          0.7898723817747162,
          0.02283995407183305,
          0.11485329640673592,
          0.08889038861217811,
          0.9058214960505804,
          0.033456980293837306,
          0.19696706839609343,
          0.016010862663409724,
          0.03962311926166313,
          0.21505618473928134,
          0.04797513392139256,
          0.7825259865349374,
          0.10157678963523618,
          0.015094013557797957,
          0.01361107151376025,
          0.888370991223175,
          0.08143916671247538,
          0.025496461090779403,
          0.020289799469849247,
          0.02161272545420333,
          0.7299841772231987,
          0.6113036934954857,
          0.17906084191982194,
          0.3107140826982653,
          0.02711271312676181,
          0.09496424035772999,
          0.09794894286015264,
          0.2936545223150206,
```

0.11817600447096488,
0.08096917324445167,
0.024253678434862144,
0.755007913810675,
0.811532655110211,
0.9680506984114012,
0.7734803030331603,
0.044650996523089376,
0.04859609101734506,
0.04505869754163358,
0.8676518885210662,
0.9031441591765481,
0.8770742930019888,
0.2864343158015732,
0.7755487537747409,
0.7285266458849635,
0.06709359146693347,
0.12284563636155725,
0.13448717936939136,
0.5334249243957178,
0.2691331272541635,
0.8299615648674172,
0.34620587489953203,
0.8163213784820774,
0.007527021693307151,
0.04435495717533687,
0.02036937982860176,
0.27739491784827636,
0.9128165858107548,
0.7609563719434288,
0.03152506619087605,
0.043482100838084584,
0.7422684462744691,
0.0629928435932503,
0.8441642630859207,
0.04681807460890803,
0.14681944486129062,
0.8638487265257342,
0.10069769860915798,
0.13117409611448805,
0.6142114459239668,
0.011337235335523783,
0.020602916168835646,
0.8276275413889307,
0.05695758714012357,
0.06114533496310733,
0.033591843895207994,
0.2837238257570668,

0.11633136456154283,
0.24321683894981988,
0.045607566770868815,
0.03815265208009564,
0.8692992670699718,
0.039855065659986956,
0.8153790522382445,
0.9537798790739953,
0.03655419467578019,
0.7538197764339274,
0.06335538480136264,
0.041262694986520146,
0.04145712760761284,
0.599771869831537,
0.26779544181109455,
0.8846422305966182,
0.02368465703430353,
0.14066613390045238,
0.7715795193592605,
0.21882277973749065,
0.04413584267631312,
0.8507903472660545,
0.10236194823717196,
0.11090673140997839,
0.7807443521840622,
0.030568669379454277,
0.009137456629200793,
0.7028249429579392,
0.027493455429084986,
0.05336822625497851,
0.0110725593561745,
0.7824648221948152,
0.1887416432630242,
0.8693393362408884,
0.4011465724670709,
0.8290560244523463,
0.057725087670817056,
0.07493274842402478,
0.08286301569901448,
0.8420151477723956,
0.027889297582758148,
0.039341577333877685,
0.14638935074752094,
0.08606278988441399,
0.015071391679288393,
0.10143100098579047,
0.8568618008621945,
0.05155222794126176,

```
0.19955978237884883,
0.16356994532638247,
0.07950212545365988,
0.5341674590534641,
0.7879886563150711,
0.9528970553197801,
0.7974414245026443,
0.8559656648764629,
0.0936556767526703,
0.02760551991348087,
0.40045158570751205,
0.5802635316270608,
0.07767768723944986,
0.02377739975121511,
0.7889437518866975,
0.15056306600518896,
0.02581113806024562,
0.8004220745535258,
0.7963747647859792,
0.10892781537122254,
0.8005881201055141,
0.0973241596715372,
0.35317010372779367,
0.021741869555717837,
0.06580136891128044,
0.04955997857469003,
0.4321971594448727,
0.03667753600676889,
0.7758222966635024,
0.0847952000773815,
0.12171623004459818,
0.6833061015031306,
0.20549577704970157,
0.34764917841385884,
0.7811244324988511,
0.03567010555373079,
0.02381296889727196,
0.6788636451557128,
0.028540358787706843,
0.1798352684620513,
0.7863092843010405,
0.8752904525182209,
0.03628267995709873,
0.6598849188355773,
0.01840888438606852,
0.07837848275819584,
0.7333031196060654,
0.08637705935929636,
```

```
0.03623151129272842,
0.22876974433991706,
0.7463822782343573,
0.021338297861177007,
0.5279326893348864,
0.18457757266949965,
0.14996241792803586,
0.04627889199956972,
0.03980638618646278,
0.77720293210833,
0.06840482621578932,
0.17933229381802498,
0.8626735064916035,
0.06670245759012589,
0.11671863367092344,
0.7935450133436706,
0.687557722331818,
0.8039261972540337,
0.8292461072994293,
0.07102134565661154,
0.03715822160933788,
0.03879407952411825,
0.8488501802428586,
0.00839240181278266,
0.04692481506395889,
0.09103597354421783,
0.7912375282817715,
0.02624133612983703,
0.6896693146324954,
0.8390779964146741,
0.8124700453909213,
0.8355826591519104,
0.09759615683053811,
0.02463090613252483,
0.5671602682279104,
0.94148363282358,
0.16969277530422247,
0.5006140405795321,
0.6459717196452374,
0.5627922174285582,
0.09355401208607081,
0.04545515818383187,
0.027541705368324614,
0.0336374590157086,
0.8720437635013186,
0.6504660424898902,
0.023024813753959376,
0.046263987462113916,
```

```
0.5031406592344232,
0.15052879023604201,
0.08283787845769025,
0.05859952915933266,
0.043961147823288624,
0.8102628159693269,
0.21853403548927758,
0.1139009053911997,
0.8175518807782358,
0.40072628158410467,
0.09241431028785349,
0.07236306401752986,
0.8078803337079259,
0.013826304011974969,
0.1621020485557297,
0.02361623569521956,
0.12658539222810858,
0.0387941586894237,
0.055331229416749034,
0.16782047647205403,
0.0955379633166281,
0.04995873321359266,
0.054247997901874546,
0.2496233242730385,
0.06967692593147395,
0.1625649211045757,
0.0650401987355196,
0.17274097262090984,
0.09529209237363868,
0.04422364131477578,
0.882087853042569,
0.11797214608038017,
0.8533097309874634,
0.19029203740184017,
0.030976834230369968,
0.06021285250120027,
0.019470821504067964,
0.8931768368544019,
0.032353306858577735,
0.8567543048218778,
0.09181308563221578,
0.2812703836829187,
0.15436306401542924,
0.8846272935054955,
0.862591611783309,
0.03656246113977437,
0.22428053277308532,
0.0878350769107389,
```

```
0.04188266949609989,
0.03836088905989063,
0.04137852345218488,
0.25019672385707864,
0.05482263369579996,
0.9254294200368814,
0.019284337217646816,
0.07012210661088815,
0.05480933530659869,
0.014927617037906845,
0.045353900606247353,
0.04384248220106566,
0.02406858241701164,
0.09104319392678482,
0.17273261680621882,
0.8397260259671073,
0.8234232354950243,
0.052372825499459326,
0.10346524648709855,
0.5224808901168403,
0.03155119157111125,
0.13306347988437484,
0.6544355487538466,
0.04930004692536584,
0.0271877427166508,
0.8026730270561805,
0.9556120234786132,
0.050308968032709615,
0.046018933413284786,
0.7884440772790569,
0.11791212241046188,
0.05444552667106873,
0.03014004853084326,
0.0827822341436555,
0.015378696440866731,
0.07359801611102401,
0.13697970481849053,
0.05316862274937021,
0.8332240617954272,
0.03644856102212062,
0.08193109082975258,
0.1480908477177682,
0.09339215428918399,
0.530993384927245,
0.008983191465793822,
0.10000764724389531,
0.761297828158388,
0.8263969454614255,
```

```
0.923520964241546,
0.18290271330192223,
0.8973916749825147,
0.12034060465225635,
0.12324509977391401,
0.6155557156030513,
0.7523921405422875,
0.11292121322640403,
0.8010687788746912,
0.1418763459408232,
0.7728865904344532,
0.14628785536277586,
0.9175445023186896,
0.43088352887729964,
0.030940031983187462,
0.03336651695779738,
0.05790454262774589,
0.02601923836669508,
0.7788338754236918,
0.07640920457314654,
0.060642891735890075,
0.06169737812606349,
0.7188502031522721,
0.07424133578060023,
0.04293924627405292,
0.03813925933194923,
0.016631453270186523,
0.7958239031512985,
0.8564702223291554,
0.7809459772450746,
0.32620120544051784,
0.06263653651474213,
0.8469903960097092,
0.08679788414704095,
0.7755547906121109,
0.599239035315371,
0.09358735746069503,
0.30765594153271947,
0.17696828356780675,
0.04675235969342776,
0.6943387604195029,
0.09216415461723998,
0.05735958015676505,
0.016498584197939804,
0.08879571782034305,
0.042669432055539976,
0.9006044954417968,
0.10271833373981666,
```

0.013046286351281437,
0.8417004145439028,
0.05702070206547915,
0.27735659189996575,
0.5040709922805695,
0.03167534791873332,
0.042629412053432195,
0.5728597942298648,
0.05059979391923468,
0.7160271112460042,
0.4485584176560649,
0.8048133623448269,
0.11545663950101923,
0.8298897811009756,
0.0395772379397425,
0.0784573714213766,
0.00669470051712216,
0.8646183719524186,
0.8975996305156583,
0.04527800082678335,
0.03513891993027055,
0.033716488342019,
0.024430131603633887,
0.7859285115151049,
0.17396760734240801,
0.0981300297804891,
0.7588966571830569,
0.08662358041653767,
0.8180351614179336,
0.1592480959606271,
0.027689724120617466,
0.08000535549324164,
0.06830581526467698,
0.10391982586839613,
0.23179526033188938,
0.12532647495937582,
0.01747946796125643,
0.162498127754935,
0.01294427284845027,
0.05444804668549733,
0.12417020537032587,
0.040615421753990875,
0.14786792723564793,
0.0334975359595367,
0.11605242684908508,
0.08790479307825107,
0.09407030037306756,
0.41026257638009356,

```
0.8070169607185232,
0.07085016411119359,
0.060926077988763906,
0.08819011702746357,
0.0424007569562946,
0.33024264120959845,
0.39440249695332374,
0.08605633777072877,
0.07689532969183596,
0.12205898454973997,
0.8163024520921964,
0.051611634922671755,
0.15728390768803543,
0.01707038692043271,
0.8437278972431573,
0.2661651266786621,
0.09849342126003768,
0.8195547916477743,
0.13916227630115174,
0.13531144286243368,
0.05377903189057956,
0.30454007764574637,
0.47901267212774445,
0.46332589979083033,
0.4139735083749796,
0.8649830259197147,
0.054747060609273104,
0.14112069870330618,
0.026330237873227826,
0.06285418755415417,
0.7996600022552053,
0.8114252404546033,
0.01767915030126871,
0.07612241322277613,
0.042150850214459086,
0.09712759196439565,
0.2378205904452341,
0.1672314362563881,
0.011189923775826977,
0.08970425578216479,
0.7027635270331927,
0.06256620130080569,
0.030448622148822737,
0.03319874604784034,
0.2607128638277222,
0.024720351716812927,
0.04420191616788702,
0.741372333637274,
```

0.09079992330641322,
0.559757231964132,
0.009438887024809972,
0.09905327050079506,
0.18490892030090886,
0.22154047570738802,
0.5157659486358659,
0.3786160700109712,
0.8065015811144612,
0.05427355679920139,
0.31047550471479823,
0.9050631788581239,
0.5041501042933492,
0.14516528727495373,
0.24294434551692246,
0.194514592903572,
0.09871125501584226,
0.04180652493961712,
0.07132544888928267,
0.036529687372418795,
0.24277075391796435,
0.06239414719097323,
0.03425687355001947,
0.015412983102981816,
0.30596227706314916,
0.1383803836095333,
0.15472297545267313,
0.031452675584392506,
0.08979170920946469,
0.018792405903886124,
0.09193088572906938,
0.06031434310396201,
0.8602461958630135,
0.16666619799881693,
0.04312251041644884,
0.20910096337067666,
0.1072977092917533,
0.4960993630274483,
0.0391840479479662,
0.8178058147572451,
0.12956035910966476,
0.025013874234398454,
0.8058666413875342,
0.847945644754972,
0.052327942746561926,
0.34848135014586973,
0.025661089802193122,
0.09324652753891872,

0.06751486331638967,
0.08463834718956291,
0.2368881057993093,
0.036708321145359074,
0.7912339275355355,
0.014002815578484849,
0.050282342697645074,
0.06660703533881977,
0.5167262934471054,
0.038826065183847965,
0.5962496147325245,
0.6520119473419266,
0.06618537302023633,
0.9072332919519367,
0.3132302890978979,
0.09000397938759115,
0.8569712400081394,
0.07341549145815231,
0.674448572022239,
0.19716945602847188,
0.2655803415805162,
0.14059400399969826,
0.8719135156417538,
0.7877595036191375,
0.20230462193699067,
0.8789356454575946,
0.02298004751873121,
0.10729184908601241,
0.01750472611545504,
0.00848210237083871,
0.03136520396987811,
0.5629886541733882,
0.7218780408724171,
0.17050713976384846,
0.07848473235075433,
0.016301504446446086,
0.09117018143486745,
0.016935746547005057,
0.016677166324064772,
0.8727014505543053,
0.1010758844255709,
0.4726539883094523,
0.17853092835943854,
0.20306922382000683,
0.8253488820032806,
0.1899450382850318,
0.02811437314657239,
0.8314516069308707,

```
0.8640795812260991,
0.033012726342809405,
0.11392828109782682,
0.1451208484902095,
0.8163981218594141,
0.9146420416340777,
0.7278960107281329,
0.020981790919531437,
0.23529337112537616,
0.8745521738322487,
0.3693493902553767,
0.12998069899253129,
0.04090177767169734,
0.37533059157182463,
0.04978964457428612,
0.15041055676270582,
0.1251554885794592,
0.07621827497965791,
0.012909629296260753,
0.10839592127737183,
0.7149531879361645,
0.26552079013891833,
0.8818358168671535,
0.19224313222946943,
0.6928786048820793,
0.8187420827303413,
0.027054604326946594,
0.1854359415176701,
0.020106774993846027,
0.5962791972144762,
0.10023782538385924,
0.01772255582506145,
0.826479988403443,
0.09090093052098341,
0.6124110915096985,
0.48459062318782925,
0.8394673537081994,
0.3953303205812789,
0.012480644450657179,
0.5571338930975369,
0.03692083664982146,
0.5105266848547164,
0.04582506872753681,
0.25540169022520554,
0.08605247640342523,
0.11485575636120597,
0.7681341053459239,
0.026116303125293592,
```

0.005507690009221867,
0.09153375711021863,
0.08814409325701075,
0.5190429545326438,
0.07068226541931853,
0.8909888908131346,
0.3697143504345854,
0.35057259378220207,
0.19451102920141855,
0.08121466662105295,
0.052856713880709144,
0.7461624383245818,
0.1333240209327791,
0.6805577855007545,
0.061377108452974415,
0.06701588239846394,
0.19374182755593652,
0.8648755272211538,
0.011951752982085067,
0.5481129776740673,
0.015611020738202868,
0.08042439118370501,
0.017975017158475933,
0.7691375573981964,
0.11879116292490315,
0.506476885663402,
0.015791230450737097,
0.3738375715196388,
0.04902049477629754,
0.7337184354173099,
0.6707959001913668,
0.04639053127352356,
0.14729499691444062,
0.7644768692616538,
0.04511410145930406,
0.8351084476189347,
0.02443524908078581,
0.058564151665768484,
0.8382042402641897,
0.019360715591837073,
0.8580902129905812,
0.08726433398997903,
0.025912722995818645,
0.1329963253958255,
0.791777700790401,
0.04561580641560926,
0.7081866539724477,
0.03266940389569734,

```
0.033577485419812175,
0.6587461235585887,
0.11842853125857446,
0.04257531537165338,
0.17679456810217975,
0.10986767387348684,
0.1807702232102295,
0.7127001432079956,
0.8693549636679774,
0.1449471012197448,
0.9087078540168781,
0.05379646698423287,
0.8026935871737277,
0.04802673335848018,
0.8463927791107303,
0.6381371188015582,
0.1132676905097955,
0.027188416022745507,
0.6540915973193826,
0.10774735622422453,
0.384087699772142,
0.8021472102966487,
0.7782541938783462,
0.076432022467698,
0.15537281877001963,
0.05060374224602149,
0.020274363673312045,
0.04971372589847311,
0.050642526538121575,
0.055066835452644076,
0.3085055326263972,
0.10444745759041221,
0.02411117001837626,
0.14416899759024102,
0.2378443692056544,
0.2498193246023654,
0.12108309429365974,
0.11417059980428497,
0.8273382338840495,
0.027429856258360207,
0.06376171196217252,
0.8207238452269936,
0.8432027672735175,
0.8718605262148087,
0.7012049906823101,
0.7760949017101292,
0.1071630282272484,
0.9771485817396284,
```

```
0.1443335234587046,
0.027289900034386253,
0.07440994363984603,
0.02625606607956406,
0.8471371983768172,
0.05694150065682453,
0.15561803725225654,
0.224432845367267,
0.0894278746014269,
0.28280432011731793,
0.14262997052597673,
0.014610780842402823,
0.1168043258328834,
0.07209133273939757,
0.06990128095157223,
0.019869357028332018,
0.036818475763713124,
0.016190586987173117,
0.6348363178762986,
0.05974066120863124,
0.022011306173546255,
0.0301205418196395,
0.07276237301652431,
0.04814171559895874,
0.10565574941008213,
0.08861309912931081,
0.1443072131933634,
0.8298861724421241,
0.6820219912244976,
0.8366532765442161,
0.06774466594564033,
0.051732342400913754,
0.29960453035201284,
0.33781545326071744,
0.020001798836236064,
0.6627110667135904,
0.3506147572851561,
0.03841395415046941,
0.01646162561519374,
0.08593308624667176,
0.8178455708200603,
0.06069436664259551,
0.49969719302937987,
0.1345621854812077,
0.058870141673713884,
0.03821015206657506,
0.3758863959960152,
0.3035148001822348,
```

0.02087069758800957,
0.9848451120498912,
0.16892326694106036,
0.8889206235356546,
0.3940868243505691,
0.7669251068798805,
0.8031871399633084,
0.4700076712890369,
0.10217073417014984,
0.5358999834059398,
0.1743617732058141,
0.8173213307951958,
0.004586840079114118,
0.8801634692149759,
0.14861422460785648,
0.29061294949151417,
0.17443276470972083,
0.006624526140101459,
0.05262736930413181,
0.05075106586355853,
0.3275299207455056,
0.8529094860868469,
0.042158333157801244,
0.05737240223273164,
0.04708385031914897,
0.7363992117325632,
0.25661793938849176,
0.037913409637181254,
0.15204954793634623,
0.7883869167540001,
0.03580012012494305,
0.570258038661736,
0.8125352300755633,
0.025054303945948507,
0.8626228996505616,
0.8467444110529008,
0.9127546494895802,
0.6507718051404439,
0.035091878303624376,
0.07387057612892307,
0.053604533981443506,
0.04251097585988816,
0.02694650659148859,
0.09259882495427561,
0.12949134626843833,
0.832139663632639,
0.8663707717190672,
0.8561683570053553,

```
0.05352207247530718,
0.8798537928691719,
0.059645015691907007,
0.08364992686755415,
0.02466869029535704,
0.018739800247672838,
0.30889220194208544,
0.268852186518342,
0.10418737274673891,
0.15205327727841614,
0.040793150160881496,
0.13916190481597587,
0.11274932474305066,
0.06570655505118166,
0.07542304644261129,
0.39443620277377084,
0.08839712857924648,
0.33763127159378703,
0.5708692031438775,
0.7647233809947978,
0.6519995153116291,
0.03952974242666412,
0.9497995363610334,
0.5113375576376776,
0.1041564441342916,
0.424234984764907,
0.3237938287257522,
0.07395046620622246,
0.9170569604801713,
0.19672958950642477,
0.6201424127900318,
0.08046302536538248,
0.8725124183526936,
0.6772781457048671,
0.7414542308329534,
0.2229791835331376,
0.4198054277318186,
0.06074489233833945,
0.3441862125195922,
0.026581736159743363,
0.03601968716622426,
0.05604155495700871,
0.12501844969536352,
0.019421697879762428,
0.03752930993484317,
0.194429614819136,
0.8065227057722287,
0.06546550207045906,
```

```
0.04820723537802458,
0.6507807509238611,
0.03468313270397625,
0.04457543385960478,
0.06483976790416503,
0.06928839550727621,
0.023893321344299098,
0.2727983562860259,
0.5506836923960652,
0.030538701131312452,
0.4413963786761411,
0.5931790685975541,
0.8177067058898627,
0.05903876346935514,
0.20500013043175985,
0.0767682659460059,
0.24213679782802994,
0.44205939996789106,
0.7720883204624684,
0.16098160296862019,
0.08310740099878308,
0.050697094197781394,
0.06292383531807072,
0.02187520492894252,
0.045592065803594595,
0.8470672134632028,
0.7667688794372166,
0.9213739735806233,
0.12742366279163272,
0.10570734793793597,
0.12229599850304372,
0.06866827018584498,
0.17238585005986692,
0.07536370944277031,
0.6200598685190173,
0.04418801945754518,
0.06400296401545083,
0.09146909703172723,
0.07182553453355545,
0.1275790049057183,
0.6493546145605391,
0.02747399998639164,
0.014814254039532926,
0.03219815353905554,
0.03656099659630037,
0.09228301191370403,
0.0736193092226458,
0.04071064382096032,
```

```
0.19596905885652233,
0.9120411380741692,
0.029038439770315665,
0.35655425736196233,
0.1937708828462201,
0.048494743782577975,
0.7475478970867389,
0.05531931976066045,
0.8433185103339759,
0.003985414029794187,
0.031070448624176076,
0.06995811795972076,
0.01931702845318745,
0.05139724869924567,
0.10063202512404314,
0.09664465175633007,
0.8387823436663052,
0.374050640962686,
0.036075649018592164,
0.3177477044702325,
0.8438316456252335,
0.017694457022158227,
0.9028013955877902,
0.15415006033537748,
0.7949533537948679,
0.8030823864287084,
0.03368291112426417,
0.8373651833100273,
0.07068071057458433,
0.17883821762309401,
0.8777265943902629,
0.7798402055260801,
0.013917211141785431,
0.8919391216329936,
0.07025486674338925,
0.03737150409409059,
0.8447018888398559,
0.015267455879329974,
0.18821311240991317,
0.11825925184028648,
0.06973941618755632,
0.13660662840676097,
0.835872236469137,
0.015154616987309471,
0.05117902861959036,
0.08591927841524308,
0.13983953272828165,
0.04778182765686723,
```

0.07829723737936498,
0.4351427411383397,
0.24889748172475257,
0.02252885191307236,
0.00948393891627035,
0.9230390398665884,
0.06405851825415312,
0.09994338641473573,
0.07492947745587333,
0.6763903707414245,
0.6552342677473902,
0.031403948005559054,
0.5791861511857757,
0.17668830356948265,
0.7555029761206378,
0.07429449336794385,
0.3502396231358732,
0.015303796138847704,
0.5333774848593359,
0.0638187701947497,
0.08168740851304152,
0.9021019834227169,
0.05519775825033845,
0.1236488396362066,
0.8750105402422903,
0.14596182034401456,
0.6777584263814719,
0.7132500857610701,
0.703473955035297,
0.8924699773996331,
0.033190931726194875,
0.018297973356399745,
0.10029944112970829,
0.6645076528820056,
0.129728458923436,
0.7747774264220161,
0.9047066815237239,
0.023180424106962555,
0.06107162928408595,
0.03757267916479918,
0.06855747149561905,
0.11398877952202187,
0.009826080280086746,
0.060913701901350946,
0.7993591648690951,
0.8047786187360438,
0.5664728687080021,
0.046127502156164436,

```
 0.8789188927249879]
```

In [31]:
```
correct = 0
for i in range(len(Y_test)):
    if (prob_test[i] > 0.5 and Y_test[i]==1) or (prob_test[i] < 0.5 and Y_test[i]==0)
        correct += 1

correct,len(Y_test)
```

Out[31]:
```
(932, 1000)
```