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

import warnings
warnings.filterwarnings("ignore")
```

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)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=15)

X_test, X_cv, y_test, y_cv = train_test_split(X_test, y_test, test_size=0.5, random_state=15)
```

In [3]:
```
def decision_function(X_cv, support_vectors, dual_coef, intercept, gamma):
    dec_fun = []

    for xq in X_cv:

        dual_form = 0
        for i in range(len(support_vectors)):
            dual_form += dual_coef[i] * np.exp(-gamma * (np.linalg.norm(support_vectors[i] - xq) **
2))

        dual_form = dual_form + intercept

        if dual_form > 0:
            dec_fun.append(1)
        else:
            dec_fun.append(-1)

    return dec_fun
```

In [4]:
```
gamma = 0.001

clf = SVC(C= 100,gamma= gamma)
clf.fit(X_train, y_train)

f_cv = decision_function(X_cv, clf.support_vectors_, clf.dual_coef_[0], clf.intercept_, gamma)
print('----------Custom Decision Function----------')
```

```python
print(f_cv)

N_positive = len(list(filter(lambda a: (a < 0), f_cv)))
N_negative = len(list(filter(lambda a: (a > 0), f_cv)))

print('----------Custom Decision Function Positive and Negative Count----------')
print(N_positive)
print(N_negative)


print('----------Classifier\'s Decision Function----------')
clf_f_cv = clf.decision_function(X_cv)
print(clf_f_cv)

print('----------Classifier\'s Decision Function Positive and Negative Count----------')
clf_N_positive = len(list(filter(lambda a: (a <= 0), clf_f_cv)))
clf_N_negative = len(list(filter(lambda a: (a > 0), clf_f_cv)))

print(clf_N_positive)
print(clf_N_negative)
```

```
----------Custom Decision Function----------
[-1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1,
-1, -1, 1, 1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1,
1, 1, -1, 1, 1, -1, 1, 1, 1, -1, -1, -1, 1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -
1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -
1, 1, 1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -
1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, 1, -1, 1, -1, -1, -1,

-1, -1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1,
-1, 1, -1, 1, -1, 1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, -1, -1, -1,
1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -
1, -1, -1, 1, 1, -1, -1, -1, -1, 1, -1, -1, -1, 1, 1, -1, -1, -1, -1, 1, -1, -1, -1, 1, 1, -1, -
1, -1, 1, -1, -1, -1, 1, 1, 1, -1, 1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
-1, -1, -1, 1, -1, 1, 1, -1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
-1, -1, -1, -1, -1, -1, 1, -1, 1, 1, 1, -1, -1, -1, 1, -1, 1, -1, -1, -1, 1, -1, 1, 1, -1, 1, -
1, -1, 1, -1, -1, -1, -1, 1, 1, 1, -1, 1, -1, -1, -1, 1, 1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -
1, 1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1,
1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, 1, -1, 1, -1, 1, -1,
-1, 1, 1, -1, 1, -1, -1, -1, 1, 1, -1, -1, 1, -1, -1, 1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -
1, -1, 1, 1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1, 1, 1, -1, -1, -1, -1, -1, 1, -1, -1, 1, 1, -1, -
1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
-1, -1, 1, 1, 1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1, -
1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -
1, -1, -1, -1, -1, 1, -1, -1, 1, 1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1,
-1, -1, -1, -1, -1, -1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
-1, -1, 1, -1, 1, -1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, 1, 1,
-1, -1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, 1, -1, 1, -1, -1, -1, 1, -1, 1, -1, 1, -1, 1, -1,
-1, -1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, 1, -1, 1, -1, -1, -1, 1, -1, 1, -1, 1, -1, 1, -1,
1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, 1, 1, -
1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -
1, -1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, 1,
-1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, 1, -1,
-1, -1, -1, -1, -1, -1, 1, -1, 1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1,
1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, 1, -1, -1, -1, 1, 1, 1,
1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, -1, -1, 1, 1, 1, -1, 1, 1,
-1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, -1, -1, -
1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -
1, -1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, -1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -
1, -1, 1, -1, -1, 1, -1, -1, -1, -1, 1, 1, 1, -1, 1, 1, 1, 1, -1, 1, 1, 1, -1, -1, 1, 1, -1, -1,
1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, 1, -1, -1, -1,
1, -1, 1, -1, 1, -1, -1, 1, -1, 1, -1, -1, -1, -1, 1, -1, -1, 1, 1, -1, 1, 1, -1, 1, -1, -1, -1,
-1, -1, 1, -1, 1, -1, 1, -1, -1, 1, -1, -1, 1, -1, -1]
----------Custom Decision Function Positive and Negative Count----------
702
298
----------Classifier's Decision Function----------
[-2.15980218 -2.63156431 -3.61056706  1.92540143 -0.983995   -1.84103624
 -3.0473973  -0.91080547 -1.07386141 -2.84948753 -0.24358518 -1.45335998
  1.91554997  1.96395045 -4.2875512  -1.96907276 -3.07109402 -0.63542208
 -2.11742229 -2.66989748 -2.33079579 -2.29643313  1.35137917  1.65469048
 -3.31402092 -1.48759379 -2.56103889  1.95781844  1.42575421 -3.15650408
 -1.78262728 -3.8510279   0.36278684  1.37387541  1.02021079 -2.51766924
 -1.99092502 -2.79077121 -2.55543383 -2.60288777 -2.94227384 -2.45524026
 -2.20947219 -1.85628265  1.35625834 -0.21635295 -3.2076807   1.24322851
 -2.67163654 -4.04416999 -2.97301845  0.56742762  2.02126299 -3.12793741
  1.85785685  0.76082574 -1.78027253  0.10260672  0.79058505  2.16755997
 -2.3809447  -2.63048338 -0.33608667  0.8128311  -3.35588898 -1.78973346
 -2.68757312  1.94906137 -1.8825106   0.64979623 -2.69724423 -0.81592694
 -2.48190921 -1.46426242 -2.29823709 -1.78005635 -1.47700104 -3.0912695
 -0.22899215 -0.12787648 -2.17724863 -1.45951845 -3.16849987  0.83678511
  1.42923867 -2.94919111 -1.47607856  1.4419966  -1.88341072 -1.26260517
 -2.44879424  1.17365042 -2.61264794 -0.19452622 -0.1757905  -2.45715487
 -1.02869514 -2.46505783 -2.22591715 -2.19360318 -2.81086472 -3.70556298
 -2.30713711  0.33887917  1.40706541 -1.6686049  -1.87965386  1.96320471
 -0.95697405 -0.89307499 -3.29506317  0.64202639 -2.56244327 -2.57164914
 -2.29006499 -3.02552808 -3.15718587  2.42515727 -1.33501655 -3.8716577
 -2.43426245 -4.05549613 -2.54395732 -2.45362881 -1.47674614 -3.28685719
 -2.73146113 -1.60889081 -3.80160411 -2.85481918 -0.94331514 -1.85535315
 -2.37182631 -2.98238703 -3.69302377 -2.33112437 -1.93180889 -1.72943827
 -0.35416286 -0.27988187 -2.0141241  -2.76530052 -2.63476831 -1.85006939
  0.78919404 -3.21556378  0.52861162 -4.60658906  1.85013265 -2.42873331
 -0.38356448 -3.14438097 -4.16158651 -2.17053734  1.24857961  2.25101163
 -2.56703238  1.7621164  -1.5688125  -2.77427088 -3.66978294  1.23947709
 -2.71634149 -0.1945509  -2.39531858  1.94684564 -1.22119781 -1.70997025
  0.20128025 -2.56710578  0.62118073 -3.72713147  0.45947394 -3.2822854
 -3.48839716 -3.09148402 -1.70106455 -2.083701   -2.63906678  0.70057693
  3.48859716  3.09148462  1.70100455  2.085701    2.65900078  0.70057695
 -2.49142816  1.5156687  -2.69553691  0.43932627 -1.96248844 -2.04496831
  1.62174458 -3.0642911  -2.4640089  -1.19130086 -2.87226123  1.71991839
 -2.92076307 -1.7753934   0.24190522  1.56095099 -1.34968329  0.79356986
 -1.93461786  0.31774122  0.12098802  2.0468341  -1.81138025 -0.38810669
 -4.63093689  1.73351701 -0.69388184 -1.83112441  2.60205141 -2.70860004
 -0.59676479 -3.43076375  1.42638139 -3.88179416 -2.40078572 -1.5852617
 -1.6629571  -3.1661962  -1.14298282 -2.405439    0.9513465  -2.77647212
 -2.08838809 -2.93129672  0.82369474 -2.65414493 -0.88902852 -0.24242697
 -3.72675762 -1.57788551 -2.33955819 -3.04756277 -2.40232568  0.95296644
  2.20550042 -2.06506247 -0.32375952 -4.01439429 -1.42072037  0.23947107
 -1.60413591 -2.43402956 -3.80629624  0.13418211  1.48163106 -1.46756562
 -2.20184571 -2.24870171 -3.12964223  2.44334959 -1.48509016 -2.33157765
 -3.14396565  1.39347802  2.05468679 -0.67163599 -1.65671151 -4.1149448
  0.03170109 -2.07094525 -0.01437598 -3.54308803  1.7234411   1.62576041
  2.06024866 -1.14615317  0.98120738 -0.89721897  1.48317888  0.66877933
 -2.55389652 -0.76236997 -1.89547025 -2.99066685 -3.67960064 -4.26256466
 -2.81246726 -2.72715334 -1.02660909 -1.13781664 -3.96572096 -2.84878608
 -3.33847607 -4.41010174 -1.03977967  1.74013518 -1.67107701  1.7243984
  1.94690945 -2.81488166 -1.41034137 -6.39579626  2.32771782  1.38153911
 -2.26675953 -1.61010132  2.3727565   1.25860156 -1.12905226 -1.04946381
 -0.56749724 -1.61471626 -2.38224992 -0.72407294 -3.27027432 -1.52152755
 -2.64084731 -0.44127203 -0.60886299 -3.72364233 -2.51488462 -1.97451261
 -3.23580562 -0.47429097  1.67793733 -3.180322    1.49792628  1.71596281
  2.2411538  -2.9735897  -3.35159715 -1.53804781  2.32945132 -0.60428045
  1.79969984 -0.68255575 -2.93021856 -1.8612763   1.03435797 -2.53978082
  1.53980707  1.79531995 -2.00954178  0.12169172 -2.95602356 -2.61946372
  1.04153989 -1.70781107 -3.03509516 -2.6609708  -2.14195421  2.18860009
  0.54642573  1.32386969 -4.38666731  1.40433428 -2.17601983 -3.62688758
 -1.60314212  1.19668499  1.52321905 -1.50574634 -3.64487569 -3.31281217
  1.17002256 -1.94128883  1.47075955 -0.20001308 -2.43410951 -2.09132221
 -2.31145385  1.72014463  2.19965463 -3.62751359 -2.97898187 -3.28294057
  1.71276777 -2.35287723 -0.09212771 -2.23464743  1.55086989  2.03283166
 -1.26840949 -2.08185371 -2.81463571 -3.53503016 -2.70573659 -2.31137105
 -2.93743257  0.44697806 -0.94587578 -2.41643104 -2.81195108 -2.22653596
 -2.36997722 -2.54536346  1.93145322  1.70906467 -2.81176377  1.69537175
 -3.92364572 -2.71312804 -1.87701795 -1.20742789 -3.40374364 -2.14001954
 -1.62558592 -2.85609622 -3.25572582 -3.17337819 -4.04836394 -2.24827386
  1.79349266 -2.6319856  -2.28260078 -4.88354336  1.69721836 -1.85897828
  0.34542681 -1.98639633  0.72883073 -1.74658687 -2.18743969  1.53051114
  1.41858136 -1.44297037  0.38431313 -1.97769971 -3.12286583 -1.3503888
  0.77092584  0.32353664 -3.28458229 -3.22643897  1.41980417 -2.91718458
 -0.83241382  1.6366444   2.40164025 -2.12922055 -0.20679871 -2.68607595
 -0.55507727 -2.7865579   0.31178432  1.90975401 -1.16512444 -2.00326225
 -2.47994961 -3.53068546  0.3518022   0.76854564 -2.46100953  1.52288079
 -0.59253147  0.0647706  -1.86101666 -2.07169046  1.3335099   1.76461231
  1.55482017 -1.90734332  2.2340956   1.2349462  -2.50827706 -4.47414772
 -3.04657454 -4.45683606 -4.16856748  1.91862494 -2.16906493 -2.82132138
  0.94069656  1.80383908 -3.13149678 -3.67942967 -2.49095344  1.21828214
```

```
  0.94009656  1.80383908 -3.13149078 -3.07942967  2.49095544  1.21828214
  0.01432633  0.14202389 -2.16148369 -2.58467402 -0.22579788 -3.09519938
 -2.24952792 -1.11205997  2.19496914 -2.03385667 -2.7590884   0.05559613
 -2.84159482  0.59671851 -0.93506139 -2.80238331 -2.58346544 -3.71731953
 -3.68517351 -3.52730079 -2.293103   -2.28506329 -1.15304833 -1.61429866
 -0.69914543  0.89398308  1.55620082  2.25620076 -2.33863221 -0.16902652
 -2.47349584  1.89303168 -1.69249391 -2.61288462 -2.51327293  1.33897332
 -2.86942374 -0.06402366 -3.21187008 -2.30705807 -2.83395614  0.6992984
 -1.62840335 -2.58431333  1.90523352 -3.09891654 -1.35683654  0.49242965
  0.69465375 -2.54830123 -3.34855589 -2.97736198 -1.53306203  1.60534204
 -3.46889089 -2.38978224  0.80663775 -3.10182342 -0.61655844 -3.40378099
  0.2561897  -0.88020665 -1.06090848 -2.57090783 -1.29990198 -2.26140212
  0.88318773 -1.95012627 -3.23228229 -3.97773118  2.05021314 -2.09675087
 -2.5175608  -2.73216436 -3.26107512 -0.30944441 -1.65209485 -1.98835357
 -2.95094813  0.6612601  -2.60979922 -3.31384818  0.42256809  0.33516186
 -2.64488165 -2.48607376 -1.9103341  -3.43248702  0.19425709  1.88208605
 -0.20338178 -1.18198128 -3.48643767 -3.21081454 -2.52552191  0.45511322
 -2.13002192 -0.14940011  1.6780305  -1.91319714 -2.99635864 -3.86989111
 -1.03215174 -1.3327501  -2.36478739 -2.62946013  2.01460628  1.10872694
 -2.48946758  0.14082528 -2.54135874 -2.00102952 -1.92131567 -0.42640764
 -3.03184236 -1.91683345 -1.4113323  -3.49006219 -4.04893848 -1.68765914
 -1.89241193 -1.94038831 -3.05105537 -0.00787114 -4.19762266 -1.37093457
 -1.76324571  1.05240804 -4.67712474  0.98042669 -1.7718004  -2.9429454
 -1.97680251 -1.36330317  1.60542827 -0.30225942 -0.40910623 -2.4613492
  0.98463774  1.97743101  2.77011395  1.58702113 -4.35327656  1.0814428
 -0.47782772 -0.47038216 -4.53346065  1.0680813  -2.66348232 -3.8665378
  1.228343    0.96692308 -1.60183748 -2.57459878 -2.80814918 -0.45721292
  1.40014055 -3.1641464  -2.27500273 -3.76875737 -2.45602972  1.96151345
 -3.34331197 -2.74896277  1.50701426 -2.89186325  2.37242778 -0.67194692
 -3.08096963 -2.85435647  1.4785375  -2.90104048  0.71382988 -1.06804546
  0.8249903  -1.64674306  0.43278271 -2.40402416  1.44483616 -1.15541322
 -4.08211557 -0.7174186  -3.53693671 -2.55744604 -2.37008034 -0.44369822
  0.74838698 -2.28592702 -1.81122432  0.33078696 -3.35132945 -2.84014244
 -2.38632576 -3.08077551 -2.2411865  -2.96942114  2.16793969 -2.67077059
 -3.50145285 -3.51467728 -3.09152981  1.52314457  0.74988939 -1.58890615
  1.75052207 -3.95963639 -1.81158617  1.42959203 -0.29093612 -2.85905831
 -2.09215298 -3.18956687 -1.97182953 -4.02019097 -2.77744157 -2.47775065
 -2.07670388 -1.67351053  0.73696639 -2.88605367 -2.26207219 -0.35239587
  2.06264035 -2.48960521 -3.81094587  1.88155814 -3.58239427 -2.27743544
 -2.67385708 -1.24574456 -2.52730127  1.99172183 -3.31666121 -1.57725713
 -4.34813284 -1.83425434 -1.54236265  1.48605761  1.62067393 -3.09671223
  0.9641951  -2.83706466 -1.01813522 -2.03723033 -3.16419486 -1.76802907
 -2.73169955 -2.55126285 -1.25354924 -1.4342827   0.43939114  1.48187108
 -3.47629413  0.28786473 -2.03297531 -1.98544852  1.00251398 -1.99505734
 -3.27792591 -0.10579637 -2.3889764  -1.41376948 -3.60673759 -4.45437328
 -1.59449637 -4.02151836 -2.71698927 -3.32530835 -1.77512819  1.47350828
  1.25168064 -2.21781915 -4.12147933 -0.40500486 -2.42039435 -2.79218287
 -3.12641735  0.70507578 -2.94779833 -3.06481894 -3.12906001 -0.21137441
 -3.12811074 -3.28740694 -2.45217249  1.20775072 -3.42596664  0.06765081
 -1.9198628   2.34355842  0.83575185 -3.31501017 -2.96375563 -3.04021592
 -2.7392466  -4.12923767 -3.26319186 -1.13756245 -3.39801432  1.84285138
 -1.55289968 -1.90774054 -2.19978457 -2.88021893  1.43716531  0.34759214
 -2.33413783 -2.12580765 -1.03710688 -2.88124882 -2.07072708 -3.18901296
 -3.19801304 -2.73678037 -4.10052551 -0.84894147 -3.35948899 -3.17152471
 -2.24344275 -2.71997207  0.57853771  1.64344749 -1.15940651  0.54870855
 -3.12917941 -2.83746983 -2.30372377  0.68757169  3.60066932  0.50742447
  1.75119335 -2.89546918 -2.8431726  -0.40152856 -2.08746321 -2.71354627
  1.00487695  1.28183754  1.93150597 -2.12760574 -3.06583782  0.46420881
 -1.60034843 -2.45722796 -0.36410775 -2.16703236  1.11415669 -2.39748488
 -2.02125315 -0.63043527 -3.43837092  1.60421018  1.60102555  1.44467886
 -2.8646173   1.44517369  0.72242205 -0.90251483  1.12962424 -1.95013101
 -2.66595681 -3.58259406 -2.38770052 -3.8852265  -3.78241459 -1.62824652
 -2.94354056 -0.51402289  0.16350674 -0.56036971 -2.75345155 -0.82282745
 -2.6917646   1.63878699 -2.60608879 -1.48540573 -2.68929027  1.27097416
  1.36503943  0.44289025 -1.96217213 -3.54434078 -3.40682105  1.96377098
  0.83572567 -2.49074615  1.87725192 -2.57294861 -0.18133062  1.80708255
 -0.78636667  1.32371314  2.092229   -3.82198796  1.87563444 -0.18433889
 -3.61932068  1.85167769 -1.29225887  1.13801028 -0.61289767 -1.49150583
  1.05632147 -1.80842883  1.19494635  1.95510123 -3.7779399   1.46357909
 -0.73804696 -1.8972708  -1.42008043 -3.3815173   0.13753235 -1.6669742
 -1.42670913 -2.23445028 -1.29519944 -2.8695093   0.58361849  1.09034815
 -0.7608024   1.38911476 -2.60759005  1.53263216  1.39907937  1.4000444
 -2.72923211 -5.27821077  1.43832987 -1.64638025  1.20672779 -4.08659398
 -2.81358064  0.88654953  0.97230045  1.63332843 -0.42784894 -3.93629667
  1.78434163 -2.2002323  -1.18019442  1.20079024 -3.02243228 -0.57589589
 -2.63289476 -1.66918112  1.98245219  0.65456163  1.51260437 -0.1104851
  1.8633443   1.77128512  1.56334123  1.11248602 -2.82255246  1.63141641
  1.94836947  1.42117293 -2.52752754 -1.0673132   0.28167302  1.78089906
 -3.49549604 -2.43163379  2.25522108  1.40303847 -0.65392807 -2.36341217
 -2.19911947  2.27610667 -2.66753926 -1.49552427 -1.67473047  1.66884805
  0.46345355 -2.62203341 -3.04806184 -1.86853662 -1.75882538  2.08299906
  1.23411847 -2.84601467 -2.25160773 -2.37568678 -2.32866    -1.56905767
  0.44659481 -1.67067997 -1.60605292 -0.22738896  0.52582961 -2.50183178
  2.18653678 -2.43362863  0.79705488 -2.59528511 -2.98541635  1.82222146
 -1.43011946  1.67947797 -1.93040304 -2.37819418 -1.96187186 -1.36358482
  2.35197549 -4.51485231 -1.38561699  0.43196456  1.27803018 -2.67601299
  1.30105864  1.64910797 -4.06414647  1.21858262 -1.76938233 -2.43691585
 -1.75456791 -2.72448139 -3.9729416   1.72193258 -2.19981905  1.37716249
 -3.63943043  1.57945332 -0.59658929 -2.02911081  1.58879159 -2.46424861
 -2.7242094   0.72925902 -2.248491   -2.7203835 ]
----------Classifier's Decision Function Positive and Negative Count----------
702
298
```

We can see that the results are same from both Custom and SVC's decision function

In [5]:

```
y_positive = N_positive + 1 /N_positive + 2
y_negative = 1 / N_negative + 2

print(y_positive)
print(y_negative)
```

```
704.0014245014245
2.0033557046979866
```

In [6]:

```
class EpochLoss:
  def __init__(self, epoch, loss, weight, intercept):
    self.epoch = epoch
    self.loss = loss
    self.weight = weight
    self.intercept = intercept



class SGDLogisticRegression:


    def log_loss(self, w, b, X, Y):
        N = len(X)
        sum_log = 0
        for i in range(N):

            y = 0
            if Y[i] == 1:
                y = y_positive
            else:
                y = y_negative

            sum_log += y * np.log10(self.sigmoid(X[i], w, b)) + (1 - y) * np.log10(1 - self.sigmoid(
X[i], w, b))

        return -1 * sum_log/N


    def sigmoid(self, x, w, b):

        z = np.dot(x, w.T) + b
        sig = 1 / (1 + np.exp(-z))

        return sig
```

```python
    def fit(self, X, Y, eta0, alpha, num_iteration):

        N = len(X)
        lst_train = []


        #Initial w and b
        w = np.zeros_like(X[0])
        b = 0

        for epoch in range(1, num_iteration):


            error = self.log_loss(w, b, X, Y)

            #Updating weights and intercept
            w = (1 - (alpha * eta0)/N) * w + alpha * error
            b = (b + alpha * error)


            #Checking if previous epoch for train data is having the same value then breaking the epoch loop
            round_upto = 3
            if len(lst_train) > 0:

              found = False
              for loss in lst_train:
                if loss.loss == round(error, round_upto):
                  found = True
                  break

              if found == False:
                lst_train.append(EpochLoss(epoch, round(error, round_upto), w, b))
              else:
                break

            else:
                lst_train.append(EpochLoss(epoch, round(error, round_upto), w, b))



            print('Epoch= %d, Bias= %.3f, Avg. Loss= %.3f' % (epoch, b, error))

        self.coef = w
        self.intercept = b
        self.lstEpochLoss_train = lst_train
        return self

    def pred(self, X, w, b):

      N = len(X)
      predict = []
      for i in range(N):
        if self.sigmoid(X[i], np.array(w), b) >= 0.5:
          predict.append(1)
        else:
          predict.append(0)

      return np.array(predict)
```

In [7]:

```
eta0  = 0.001 #lambda
alpha = 0.001 #learning rate
num_iterations = 50

sgd_logistic_reg = SGDLogisticRegression()
model = sgd_logistic_reg.fit(f_cv, y_cv, eta0, alpha, num_iterations)

print('\n--------Weight--------')
print(model.coef)
print('\n--------Bias--------')
print(model.intercept)
```

```
Epoch= 1, Bias= 0.000, Avg. Loss= 0.301
Epoch= 2, Bias= 0.001, Avg. Loss= 0.251
Epoch= 3, Bias= 0.001, Avg. Loss= 0.209
Epoch= 4, Bias= 0.001, Avg. Loss= 0.174
Epoch= 5, Bias= 0.001, Avg. Loss= 0.145
Epoch= 6, Bias= 0.001, Avg. Loss= 0.121
Epoch= 7, Bias= 0.001, Avg. Loss= 0.101
Epoch= 8, Bias= 0.001, Avg. Loss= 0.084
Epoch= 9, Bias= 0.001, Avg. Loss= 0.070
Epoch= 10, Bias= 0.002, Avg. Loss= 0.059
Epoch= 11, Bias= 0.002, Avg. Loss= 0.049
Epoch= 12, Bias= 0.002, Avg. Loss= 0.041
Epoch= 13, Bias= 0.002, Avg. Loss= 0.034
Epoch= 14, Bias= 0.002, Avg. Loss= 0.028
Epoch= 15, Bias= 0.002, Avg. Loss= 0.024
Epoch= 16, Bias= 0.002, Avg. Loss= 0.020
Epoch= 17, Bias= 0.002, Avg. Loss= 0.016
Epoch= 18, Bias= 0.002, Avg. Loss= 0.014
Epoch= 19, Bias= 0.002, Avg. Loss= 0.011
Epoch= 20, Bias= 0.002, Avg. Loss= 0.010
Epoch= 21, Bias= 0.002, Avg. Loss= 0.008
Epoch= 22, Bias= 0.002, Avg. Loss= 0.007
```

```
Epoch= 22, Bias= 0.002, Avg. Loss= 0.007
Epoch= 23, Bias= 0.002, Avg. Loss= 0.006
Epoch= 24, Bias= 0.002, Avg. Loss= 0.005
Epoch= 25, Bias= 0.002, Avg. Loss= 0.004
Epoch= 26, Bias= 0.002, Avg. Loss= 0.003

--------Weight--------
0.0017974459628142048

--------Bias--------
0.0017974460008937745
```

In [8]:
```
f_test = decision_function(X_test, clf.support_vectors_, clf.dual_coef_[0], clf.intercept_, gamma)
```

In [9]:
```
y_pred = sgd_logistic_reg.pred(f_test, model.coef, model.intercept)
print(y_pred)
```

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1]
```