8_B

Importing libraries

In [52]:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import SGDClassifier
from sklearn.linear_model import LogisticRegression
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler, Normalizer
import matplotlib.pyplot as plt
from sklearn.svm import SVC
import warnings
warnings.filterwarnings("ignore")
```

Reading the dataset

In [53]:

```
data = pd.read_csv('task_b.csv')
data=data.iloc[:,1:]
```

In [54]:

```
data.std() # CALCULATING THE STANDARD DEVIATION
```

Out[54]:

```
f1      488.195035
f2    10403.417325
f3        2.926662
y         0.501255
dtype: float64
```

Task 1

In [55]:

```
X=data[['f1','f2','f3']].values
Y=data['y'].values
print(X.shape)
print(Y.shape)
```

```
(200, 3)
(200,)
```

In [56]:

```
clf = SGDClassifier(loss = 'log',random_state = 42)
```

In [57]:

```
clf.fit(X,Y)
```

Out[57]:

```
SGDClassifier(alpha=0.0001, average=False, class_weight=None, epsilon=
0.1,
       eta0=0.0, fit_intercept=True, l1_ratio=0.15,
       learning_rate='optimal', loss='log', max_iter=None, n_iter=None,
       n_jobs=1, penalty='l2', power_t=0.5, random_state=42, shuffle=Tr
ue,
       tol=None, verbose=0, warm_start=False)
```

In [58]:

```
coef_dict = {}
for coef, feat in zip(clf.coef_[0,:],['f1','f2','f3']): # PRINTING THE
 WEIGHT COEFFICENT LOGISTIC REGRESSION
    coef_dict[feat] = coef
```

In [59]:

```
coef_dict
```

Out[59]:

```
{'f1': 13617.93477406387, 'f2': -15636.943966575614, 'f3': 5704.9422734
29228}
```

In [60]:

```
clf = SGDClassifier(loss = 'hinge',random_state = 42)
```

In [61]:

```
clf.fit(X,Y)
```

Out[61]:

```
SGDClassifier(alpha=0.0001, average=False, class_weight=None, epsilon=
0.1,
          eta0=0.0, fit_intercept=True, l1_ratio=0.15,
          learning_rate='optimal', loss='hinge', max_iter=None, n_iter=Non
e,
          n_jobs=1, penalty='l2', power_t=0.5, random_state=42, shuffle=Tr
ue,
          tol=None, verbose=0, warm_start=False)
```

In [62]:

```
coef_dict = {}
for coef, feat in zip(clf.coef_[0,:],['f1','f2','f3']): # # PRINTING TH
E WEIGHT COEFFICENT SVM
    coef_dict[feat] = coef
```

In [63]:

```
coef_dict
```

Out[63]:

```
{'f1': -5752.778874720262, 'f2': -37467.51703073541, 'f3': 5284.7826362
11242}
```

Observations

- ACCORDING TO ABOVE RESULTS IMPORTANT FEATURES ARE f2>f1>f3 IT IS PROPORTIONAL TO VARIENCE OF FEATURES(var(F2)>>var(F1)>>Var(F3))
- BOTH IN LR AND SVM FEATURES WHICH HAVE HIGH VARIENCE IS THE MOST IMPORTANT FEATURE
- SINCE f2 HAS HIGH NEGATIVE VALUE DOES NOT MEAN IT HAVE IMPACT ON CLASSYFYING NEGATIVE CLASS BECAUSE OUR DATA NOT STANDARDIZED

Task 2

In [64]:

```
df = StandardScaler().fit_transform(data[['f1','f2','f3']]) # STANDARDI
ZING THE DATA
```

In [65]:

```
clf = SGDClassifier(loss = 'log',random_state = 42)
```

In [66]:
```
clf.fit(df,Y)
```

Out[66]:
```
SGDClassifier(alpha=0.0001, average=False, class_weight=None, epsilon=
0.1,
       eta0=0.0, fit_intercept=True, l1_ratio=0.15,
       learning_rate='optimal', loss='log', max_iter=None, n_iter=None,
       n_jobs=1, penalty='l2', power_t=0.5, random_state=42, shuffle=Tr
ue,
       tol=None, verbose=0, warm_start=False)
```

In [67]:
```
coef_dict = {}
for coef, feat in zip(clf.coef_[0,:],['f1','f2','f3']):
    coef_dict[feat] = coef
```

In [68]:
```
coef_dict
```

Out[68]:
```
{'f1': -7.27392033428641, 'f2': 3.1539523793187896, 'f3': 38.8758546179
5804}
```

In [69]:
```
clf = SGDClassifier(loss = 'hinge',random_state = 42)
```

In [70]:
```
clf.fit(df,Y)
```

Out[70]:
```
SGDClassifier(alpha=0.0001, average=False, class_weight=None, epsilon=
0.1,
       eta0=0.0, fit_intercept=True, l1_ratio=0.15,
       learning_rate='optimal', loss='hinge', max_iter=None, n_iter=Non
e,
       n_jobs=1, penalty='l2', power_t=0.5, random_state=42, shuffle=Tr
ue,
       tol=None, verbose=0, warm_start=False)
```

In [71]:
```
coef_dict = {}
for coef, feat in zip(clf.coef_[0,:],['f1','f2','f3']):
    coef_dict[feat] = coef
```

In [72]:
```
coef_dict
```

```
Out[72]: {'f1': -8.608946430950358, 'f2': 2.8645127265880586, 'f3': 41.290143425
         43963}
```

Observations

- AFTER STANDARDIZATION MOST IMPORTANT FEATURE IS f3
- f3>f1>f2 IS THE NEW FEATURE IMPORTANCE"
- AFTER STANDARDIZATION EFFECT OF VARIENCE IS REMOVED.