**4.1 Logistic Regression**

**4.1.1 Model Building**

- The dataset is split into train and test sets using stratified fraction to balance out the splits evenly.

- Preprocessing is done by segregating the numeric and non-numeric data and then standardising.

- The first model was prepared using SVC with kernel as Radial basis function. The parameters, C and gamma were tuned to train the model. C finds a hyperplane that correctly separates as many instances as possible and gamma is the 'spread' of the kernel. When gamma is low, the 'curve' of the decision boundary is very low and thus the decision region is very broad. When gamma is high, the 'curve' of the decision boundary is high, which creates islands of decision-boundaries around data points.

- The second model is prepared using LinearSVM having kernel as ‘linear’. The parameters, loss and penalty were tuned to train the model. Loss specifies the loss function. I have used ‘squared\_hinge’ as the square of the hinge loss. Penalty is used to penalise the parameter. I have used L2- ridge loss.

- Third model is trained using Nu-SVM with kernel as polynomial . The parameters,nu and degree are tuned to train the model. Nu parameter sets the number of support vectors. Nu is being set to 0.399 with third degree polynomial kernel.

**4.1.2 Model Evaluation**

1. **SVC with Radial Basis Function**

Parameters: SVC(C=8, gamma=0.035625 , kernel='rbf' )

Confusion matrix

array([[ 87, 140],

[ 73, 643]], dtype=int64)

Accuracy

0.77412513255567339

Sensitivity/Specificity:

0.89/0.38

1. **Linear SVC**

Parameters: LinearSVC( C=1.0, class\_weight=None, dual=True, fit\_intercept=True,

intercept\_scaling=1, loss='squared\_hinge', max\_iter=10000, penalty='l2', random\_state=0, tol=0.0001, verbose=0)

Confusion Matrix

array([[ 87, 140],

[ 45, 671]], dtype=int64)

Accuracy

0.80381760339342523

Sensitivity/ Specificity:

0.93/0.38

1. **Nu SVC**

Parameters: NuSVC(cache\_size=200, class\_weight =None, coef0 =0.0, decision\_function\_shape='ovr', degree=2, gamma=0.1, kernel='poly', max\_iter=-1, nu=0.399, probability=False, random\_state=None, shrinking=True, tol=0.001, verbose=False)

Confusion Matrix:

array([[ 72, 155],

[ 50, 666]], dtype=int64)

Accuracy:

0.78260869565217395

Sensitivity/ Specificity:

0.93/0.68

