6.1 PERFORMANCE METRICS:

```
#printing the train accuracy and test accuracy respectively
svm(x_train,x_test,y_train,y_test)
```

```
0.7628054264315052
0.7555555555555555
***Support Vector Machine***
Confusion_Matrix
[[719 314]
 [192 845]]
Classification Report
              precision    recall  f1-score   support

           0       0.79      0.70      0.74      1033
           1       0.73      0.81      0.77      1037

    accuracy                           0.76      2070
   macro avg       0.76      0.76      0.75      2070
weighted avg       0.76      0.76      0.75      2070
```


TUNE THE MODEL

```
0.8859627929451558
0.7454106280193237
***Random Forest after Hyperparameter tuning***
Confusion_Matrix
[[553 480]
 [ 47 990]]
Classification Report
              precision    recall  f1-score   support

           0       0.92      0.54      0.68      1033
           1       0.67      0.95      0.79      1037

    accuracy                           0.75      2070
   macro avg       0.80      0.75      0.73      2070
weighted avg       0.80      0.75      0.73      2070

Predicting on random input
output is:  [0]
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   4 out of   4 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   6 out of   6 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   7 out of   7 | elapsed:    0.0s remaining:    0.0s
```