Loading data into pandas dataframe

In [1]:

```
# Added by Duc for reproductible
from numpy.random import seed
seed(32342)
import tensorflow as tf
tf.random.set_seed(32342)
```

In [2]:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

df = pd.read_csv("online_shoppers_intention_numbers.csv")

print(df.columns)

#splitting the Class variable and the features
X = df.drop(columns=['Revenue'])
Y = df['Revenue']

#Making different datasets based on the top 10 features for testing purposes
datasets = {}

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 32342)
```

```
Index(['Administrative', 'Administrative_Duration', 'Informational',
       'Informational_Duration', 'ProductRelated', 'ProductRelated_Duration',
       'BounceRates', 'ExitRates', 'PageValues', 'SpecialDay', 'Month',
       'OperatingSystems', 'Browser', 'Region', 'TrafficType', 'VisitorType',
       'Weekend', 'Revenue'],
      dtype='object')
```

In [3]:

```
# added by Duc, data nomalization
#from sklearn.preprocessing import MinMaxScaler
#scaler = MinMaxScaler()
#scaler.fit(X_train)
#X_train_scaled = scaler.transform(X_train)
#X_test_scaled = scaler.transform(X_test)


#Commented out because datasets transformation in the follwing block of code cannot be done when
```

In [4]:

```
print("Original data")
print(X_train[0:9][:])
```

```
Original data
       Administrative  Administrative_Duration  Informational  \
2637               14               230.106944              0
9415                5               158.700000              0
11579              11               449.750000              0
6437                1                22.200000              2
2198                0                 0.000000              0
11826               0                 0.000000              4
5663                0                 0.000000              0
9553                1                 3.000000              0
2084                1                 5.000000              0

       Informational_Duration  ProductRelated  ProductRelated_Duration  \
2637                     0.00              52              2059.109203
9415                     0.00              51              1364.266667
11579                    0.00              52              1786.109649
6437                    44.40              37               400.800000
2198                     0.00               2               162.000000
11826                  117.25              96              5286.208333
5663                     0.00              45              2347.333333
9553                     0.00              13               543.000000
2084                     0.00              13               545.571429

       BounceRates  ExitRates  PageValues  SpecialDay  Month  \
2637      0.003747   0.008451    7.610431         0.0      5
9415      0.007692   0.011987    0.000000         0.0     11
11579     0.000000   0.017119   40.656712         0.0     11
6437      0.005128   0.002564    0.000000         0.0     10
2198      0.000000   0.050000    0.000000         0.0      5
11826     0.011000   0.024119    0.000000         0.0     11
5663      0.004545   0.015909    0.000000         0.0     11
9553      0.000000   0.008333    0.000000         0.0     12
2084      0.013333   0.020784   26.120154         0.0      3

       OperatingSystems  Browser  Region  TrafficType  VisitorType  Weekend
2637                  2        2       1           13            1        0
9415                  3        2       3            2            1        1
11579                 1        1       1            2            1        0
6437                  2       10       1            2            1        0
2198                  2        4       1            4            1        0
11826                 2        2       1            1            1        1
5663                  2        4       1            1            1        0
9553                  2        2       1            2            0        0
2084                  2        2       3            2            1        1
```

In [5]:

```
# print("Scaled data")
# print(X_train_scaled[0:9][:])
```

In [6]:

```
datasets.update({"17" : { "X_train": X_train, "X_test": X_test}})
datasets.update({"10" : { "X_train": datasets["17"]["X_train"].drop(columns =['Month', 'TrafficT
ype', 'Informational_Duration', 'OperatingSystems', 'Weekend','Region','Browser']),
                                                "X_test" : datasets["17"]["X_test"].drop(column
s =['Month', 'TrafficType', 'Informational_Duration', 'OperatingSystems', 'Weekend','Region','Br
owser'])}})
datasets.update({"9": { "X_train": datasets["10"]["X_train"].drop(columns =['SpecialDay']),
                        "X_test": datasets["10"]["X_test"].drop(columns = ['SpecialDay'])}})
datasets.update({"8": { "X_train": datasets["9"]["X_train"].drop(columns =['Administrative_Durat
ion']),
                        "X_test": datasets["9"]["X_test"].drop(columns = ['Administrative_Duratio
n'])}})
datasets.update({"7": { "X_train": datasets["8"]["X_train"].drop(columns =['Informational']),
                        "X_test": datasets["8"]["X_test"].drop(columns = ['Informational'])}})
datasets.update({"6": { "X_train": datasets["7"]["X_train"].drop(columns =['VisitorType']),
                        "X_test": datasets["7"]["X_test"].drop(columns = ['VisitorType'])}})
datasets.update({"5": { "X_train": datasets["6"]["X_train"].drop(columns =['Administrative']),
                        "X_test": datasets["6"]["X_test"].drop(columns = ['Administrative'])}})
datasets.update({"4": { "X_train": datasets["5"]["X_train"].drop(columns =['BounceRates']),
                        "X_test": datasets["5"]["X_test"].drop(columns = ['BounceRates'])}})
datasets.update({"3": { "X_train": datasets["4"]["X_train"].drop(columns =['ProductRelated_Durat
ion']),
                        "X_test": datasets["4"]["X_test"].drop(columns = ['ProductRelated_Duratio
n'])}})
datasets.update({"2": { "X_train": datasets["3"]["X_train"].drop(columns =['ProductRelated']),
                        "X_test": datasets["3"]["X_test"].drop(columns = ['ProductRelated'])}})
datasets.update({"1": { "X_train": datasets["2"]["X_train"].drop(columns =['ExitRates']),
                        "X_test": datasets["2"]["X_test"].drop(columns = ['ExitRates'])}})


#for dataset in datasets:
#    print(datasets[dataset])

print(Y.values)
print(datasets["5"]["X_train"])
```

```
[0 0 0 ... 0 0 0]
       ProductRelated  ProductRelated_Duration  BounceRates  ExitRates  \
2637               52              2059.109203     0.003747   0.008451
9415               51              1364.266667     0.007692   0.011987
11579              52              1786.109649     0.000000   0.017119
6437               37               400.800000     0.005128   0.002564
2198                2               162.000000     0.000000   0.050000
...               ...                      ...          ...        ...
8136               22              2281.583333     0.029697   0.043687
12078              45              1896.966667     0.011765   0.027451
7077               26               710.000000     0.003846   0.017521
4891               21               204.166667     0.028571   0.048413
4679               15               537.200000     0.011765   0.023529

       PageValues
2637     7.610431
9415     0.000000
11579   40.656712
6437     0.000000
2198     0.000000
...           ...
8136     0.000000
12078    7.521155
7077     0.000000
4891     0.000000
4679    27.664000

[9864 rows x 5 columns]
```

In [7]:

```
# added by Duc for debugging
print("Train data size", X_train.shape, y_train.shape)
print("Train test size", X_test.shape, y_test.shape)
```

```
Train data size (9864, 17) (9864,)
Train test size (2466, 17) (2466,)
```

In [8]:

```
# added by Duc
import collections
print("number of examples per class in train set:", collections.Counter(y_train))
print("number of examples per class in test set:", collections.Counter(y_test))
```

```
number of examples per class in train set: Counter({0: 8357, 1: 1507})
number of examples per class in test set: Counter({0: 2065, 1: 401})
```

In [9]:

```
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

def tsne_plot(x, y):

    sns.set(style ="whitegrid")

    tsne = TSNE(n_components = 2, random_state = 0)

    # Reducing the dimensionality of the data
    X_transformed = tsne.fit_transform(x)

    plt.figure(figsize =(12, 8))

    # Building the scatter plot
    plt.scatter(X_transformed[np.where(y == 0), 0],
                X_transformed[np.where(y == 0), 1],
                marker ='o', color ='y', linewidth ='1',
                alpha = 0.8, label ='Non-Buyer')
    plt.scatter(X_transformed[np.where(y == 1), 0],
                X_transformed[np.where(y == 1), 1],
                marker ='o', color ='k', linewidth ='1',
                alpha = 0.8, label ='Buyer')

    # Specifying the location of the legend
    plt.legend(loc ='best')

    # Plotting the reduced data
    plt.show()

#tsne_plot(X, Y)
```

In [10]:

```
from sklearn.preprocessing import MinMaxScaler



for dataset in datasets:


    scaler = MinMaxScaler()
    scaler.fit(datasets[dataset]["X_train"])
    X_train_scaled = scaler.transform(datasets[dataset]["X_train"])
    X_test_scaled = scaler.transform(datasets[dataset]["X_test"])
    datasets[dataset].update({"X_train": X_train_scaled,
                              "X_test" : X_test_scaled})


   # datasets[dataset] = scaler.fit_transform(datasets[dataset])

#scaled_datasets['10'] = X_scaled.drop(columns = ['Month', 'Traffictype', 'Informational_Duratio
n', 'OperatingSystems', 'Weekend','Region','Browser'])
#scaled_datasets['9'] = X10_scaled.drop(columns =['SpecialDay'])
#scaled_datasets['8'] = X9_scaled.drop(columns = ['Administrative_Duration'])
#scaled_datasets['7'] = X8_scaled.drop(columns = ['Informational'])
#scaled_datasets['6'] = X7_scaled.drop(columns = ['VisitorType'])
#scaled_datasets['5'] = X6_scaled.drop(columns = ['Administrative'])
#scaled_datasets['4'] = X5_scaled.drop(columns = ['BounceRates'])
#scaled_datasets['3'] = X4_scaled.drop(columns = ['ProductRelated_Duration'])
#scaled_datasets['2'] = X3_scaled.drop(columns = ['ProductRelated'])
#scaled_datasets['1'] = X2_scaled.drop(columns = ['ExitRates'])




#tsne_plot(X_scaled, Y)
```

In [11]:

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix

# Splitting dataset into train set and test set.
def makepredictions(datasets):

    for dataset in datasets:

     #   X_train, X_test, y_train, y_test = train_test_split(datasets[dataset], Y, test_size =
 0.2, random_state = 32342)


        svmclf = SVC()
        svmclf.fit(datasets[dataset]["X_train"], y_train)

        y_pred_svmclf = svmclf.predict(datasets[dataset]["X_test"])

        # Performance
        print('amount of features: ' + dataset)
        print(svmclf.get_params())
        print('Accuracy : '+str(accuracy_score(y_test, y_pred_svmclf)))
        print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_svmclf)))
        # incorrect order of output
#         tn, fn, fp, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()
        # the right order is as follows
        tn, fp, fn, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()

        print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'Fal
se negatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')

makepredictions(datasets)
```

```
amount of features: 17
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8807785888077859
Confusion Matrix:
[[2030   35]
 [ 259  142]]
True negatives: 2030
False positives: 35
False negatives: 259
True positives: 142

amount of features: 10
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8868613138686131
Confusion Matrix:
[[2018   47]
 [ 232  169]]
True negatives: 2018
False positives: 47
False negatives: 232
True positives: 169

amount of features: 9
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8872668288726683
Confusion Matrix:
[[2018   47]
 [ 231  170]]
True negatives: 2018
False positives: 47
False negatives: 231
True positives: 170

amount of features: 8
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8872668288726683
Confusion Matrix:
[[2017   48]
 [ 230  171]]
True negatives: 2017
False positives: 48
False negatives: 230
True positives: 171

amount of features: 7
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8876723438767234
Confusion Matrix:
[[2018   47]
 [ 230  171]]
True negatives: 2018
False positives: 47
False negatives: 230
True positives: 171

amount of features: 6
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8888888888888888
Confusion Matrix:
[[2013   52]
 [ 222  179]]
True negatives: 2013
False positives: 52
False negatives: 222
True positives: 179

amount of features: 5
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8868613138686131
```

```
Confusion Matrix:
[[1996   69]
 [ 210  191]]
True negatives: 1996
False positives: 69
False negatives: 210
True positives: 191

amount of features: 4
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8876723438767234
Confusion Matrix:
[[1990   75]
 [ 202  199]]
True negatives: 1990
False positives: 75
False negatives: 202
True positives: 199

amount of features: 3
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8876723438767234
Confusion Matrix:
[[1989   76]
 [ 201  200]]
True negatives: 1989
False positives: 76
False negatives: 201
True positives: 200

amount of features: 2
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8848337388483374
Confusion Matrix:
[[1973   92]
 [ 192  209]]
True negatives: 1973
False positives: 92
False negatives: 192
True positives: 209

amount of features: 1
{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'deg
ree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 't
ol': 0.001, 'verbose': False}
Accuracy : 0.8896999188969992
Confusion Matrix:
[[1961  104]
 [ 168  233]]
True negatives: 1961
False positives: 104
False negatives: 168
True positives: 233
```

Hyperparameter tuning for SVC

In [12]:

```
# added by Duc, hyper parameter tuning for SVC
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
# hyperparameter grid set for finetuning
# param_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel': ['rbf', 'poly', 'si
gmoid']}
param_grid = {'C': [0.1,1, 10, 100], 'kernel': ['rbf', 'sigmoid']}

# Splitting dataset into train set and test set.
def makepredictions(datasets):

    for dataset in datasets:
```

```
    #   X_train, X_test, y_train, y_test = train_test_split(datasets[dataset], Y, test_size =
 0.2, random_state = 32342)

########################
        svmclf = SVC()
        svmclf_grid = GridSearchCV(svmclf,param_grid,refit=True,verbose=1)
        svmclf_grid.fit(datasets[dataset]["X_train"], y_train)
        y_pred_svmclf_grid = svmclf_grid.predict(datasets[dataset]["X_test"])

        # Performance
        print('################# amount of features: ' + dataset + ' ############################
######')
        print('#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ###############
##')
        print("The best parameters are %s with a score of %0.2f" % (svmclf_grid.best_params_, sv
mclf_grid.best_score_))
        print('Accuracy with best hyperparameters: '+str(accuracy_score(y_test, y_pred_svmclf_gr
id)))
        print('Confusion Matrix with best hyperparameters: \n' + str(confusion_matrix(y_test,y_p
red_svmclf_grid)))
        print(classification_report(y_test,y_pred_svmclf_grid))
        # incorrect order of output
#         tn, fn, fp, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()
        # the right order is as follows
        tn, fp, fn, tp = confusion_matrix(y_test, y_pred_svmclf_grid).ravel()

        print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'Fal
se negatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
########################
        wsvmclf = SVC(class_weight='balanced')
        wsvmclf_grid = GridSearchCV(wsvmclf,param_grid,refit=True,verbose=1)
        wsvmclf_grid.fit(datasets[dataset]["X_train"], y_train)
        y_pred_wsvmclf_grid = wsvmclf_grid.predict(datasets[dataset]["X_test"])

        # Performance
#         print('amount of features: ' + dataset)
        print('#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ###############
###')
        print("The best parameters are %s with a score of %0.2f" % (wsvmclf_grid.best_params_, w
svmclf_grid.best_score_))
        print('Accuracy with best hyperparameters: '+str(accuracy_score(y_test, y_pred_wsvmclf_g
rid)))
        print('Confusion Matrix with best hyperparameters: \n' + str(confusion_matrix(y_test,y_p
red_wsvmclf_grid)))
        print(classification_report(y_test,y_pred_wsvmclf_grid))
        # incorrect order of output
#         tn, fn, fp, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()
        # the right order is as follows
        tn, fp, fn, tp = confusion_matrix(y_test, y_pred_wsvmclf_grid).ravel()

        print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'Fal
se negatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')

########################
        svmclf = SVC()
        svmclf.fit(datasets[dataset]["X_train"], y_train)

        y_pred_svmclf = svmclf.predict(datasets[dataset]["X_test"])

        # Performance
        print('#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)####
###############')
#         print(svmclf.get_params())
        print('Accuracy : '+str(accuracy_score(y_test, y_pred_svmclf)))
```

```
        print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_svmclf)))
        print(classification_report(y_test,y_pred_svmclf))
        # incorrect order of output
#         tn, fn, fp, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()
        # the right order is as follows
        tn, fp, fn, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()

        print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'Fal
se negatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')

makepredictions(datasets)
```

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   47.3s finished


################# amount of features: 17 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.89
Accuracy with best hyperparameters: 0.8888888888888888
Confusion Matrix with best hyperparameters:
[[2003   62]
 [ 212  189]]
              precision    recall  f1-score   support

           0       0.90      0.97      0.94      2065
           1       0.75      0.47      0.58       401

    accuracy                           0.89      2466
   macro avg       0.83      0.72      0.76      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 2003
False positives: 62
False negatives: 212
True positives: 189

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.5min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 1, 'kernel': 'rbf'} with a score of 0.86
Accuracy with best hyperparameters: 0.8661800486618005
Confusion Matrix with best hyperparameters:
[[1840  225]
 [ 105  296]]
              precision    recall  f1-score   support

           0       0.95      0.89      0.92      2065
           1       0.57      0.74      0.64       401

    accuracy                           0.87      2466
   macro avg       0.76      0.81      0.78      2466
weighted avg       0.88      0.87      0.87      2466

True negatives: 1840
False positives: 225
False negatives: 105
True positives: 296

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8807785888077859
Confusion Matrix:
[[2030   35]
 [ 259  142]]
              precision    recall  f1-score   support

           0       0.89      0.98      0.93      2065
           1       0.80      0.35      0.49       401

    accuracy                           0.88      2466
   macro avg       0.84      0.67      0.71      2466
weighted avg       0.87      0.88      0.86      2466

True negatives: 2030
False positives: 35
False negatives: 259
True positives: 142

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   37.8s finished
```

```
################ amount of features: 10 ################################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 10, 'kernel': 'rbf'} with a score of 0.89
Accuracy with best hyperparameters: 0.8888888888888888
Confusion Matrix with best hyperparameters:
[[2009   56]
 [ 218  183]]
              precision    recall  f1-score   support

           0       0.90      0.97      0.94      2065
           1       0.77      0.46      0.57       401

    accuracy                           0.89      2466
   macro avg       0.83      0.71      0.75      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 2009
False positives: 56
False negatives: 218
True positives: 183

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.1min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 10, 'kernel': 'rbf'} with a score of 0.87
Accuracy with best hyperparameters: 0.8665855636658556
Confusion Matrix with best hyperparameters:
[[1833  232]
 [  97  304]]
              precision    recall  f1-score   support

           0       0.95      0.89      0.92      2065
           1       0.57      0.76      0.65       401

    accuracy                           0.87      2466
   macro avg       0.76      0.82      0.78      2466
weighted avg       0.89      0.87      0.87      2466

True negatives: 1833
False positives: 232
False negatives: 97
True positives: 304

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8868613138686131
Confusion Matrix:
[[2018   47]
 [ 232  169]]
              precision    recall  f1-score   support

           0       0.90      0.98      0.94      2065
           1       0.78      0.42      0.55       401

    accuracy                           0.89      2466
   macro avg       0.84      0.70      0.74      2466
weighted avg       0.88      0.89      0.87      2466

True negatives: 2018
False positives: 47
False negatives: 232
True positives: 169

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   36.5s finished
```

```
################ amount of features: 9 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.90
Accuracy with best hyperparameters: 0.8909164639091647
Confusion Matrix with best hyperparameters:
[[1999   66]
 [ 203  198]]
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      2065
           1       0.75      0.49      0.60       401

    accuracy                           0.89      2466
   macro avg       0.83      0.73      0.77      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 1999
False positives: 66
False negatives: 203
True positives: 198

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.1min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 1, 'kernel': 'rbf'} with a score of 0.87
Accuracy with best hyperparameters: 0.8669910786699108
Confusion Matrix with best hyperparameters:
[[1840  225]
 [ 103  298]]
              precision    recall  f1-score   support

           0       0.95      0.89      0.92      2065
           1       0.57      0.74      0.65       401

    accuracy                           0.87      2466
   macro avg       0.76      0.82      0.78      2466
weighted avg       0.89      0.87      0.87      2466

True negatives: 1840
False positives: 225
False negatives: 103
True positives: 298

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8872668288726683
Confusion Matrix:
[[2018   47]
 [ 231  170]]
              precision    recall  f1-score   support

           0       0.90      0.98      0.94      2065
           1       0.78      0.42      0.55       401

    accuracy                           0.89      2466
   macro avg       0.84      0.70      0.74      2466
weighted avg       0.88      0.89      0.87      2466

True negatives: 2018
False positives: 47
False negatives: 231
True positives: 170

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   36.6s finished
```

```
################ amount of features: 8 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 10, 'kernel': 'rbf'} with a score of 0.90
Accuracy with best hyperparameters: 0.8901054339010543
Confusion Matrix with best hyperparameters:
[[2010   55]
 [ 216  185]]
              precision    recall  f1-score   support

           0       0.90      0.97      0.94      2065
           1       0.77      0.46      0.58       401

    accuracy                           0.89      2466
   macro avg       0.84      0.72      0.76      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 2010
False positives: 55
False negatives: 216
True positives: 185

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.1min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 1, 'kernel': 'rbf'} with a score of 0.87
Accuracy with best hyperparameters: 0.8669910786699108
Confusion Matrix with best hyperparameters:
[[1839  226]
 [ 102  299]]
              precision    recall  f1-score   support

           0       0.95      0.89      0.92      2065
           1       0.57      0.75      0.65       401

    accuracy                           0.87      2466
   macro avg       0.76      0.82      0.78      2466
weighted avg       0.89      0.87      0.87      2466

True negatives: 1839
False positives: 226
False negatives: 102
True positives: 299

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8872668288726683
Confusion Matrix:
[[2017   48]
 [ 230  171]]
              precision    recall  f1-score   support

           0       0.90      0.98      0.94      2065
           1       0.78      0.43      0.55       401

    accuracy                           0.89      2466
   macro avg       0.84      0.70      0.74      2466
weighted avg       0.88      0.89      0.87      2466

True negatives: 2017
False positives: 48
False negatives: 230
True positives: 171

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   35.8s finished
```

```
################ amount of features: 7 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.90
Accuracy with best hyperparameters: 0.8909164639091647
Confusion Matrix with best hyperparameters:
[[2007   58]
 [ 211  190]]
              precision    recall  f1-score   support

           0       0.90      0.97      0.94      2065
           1       0.77      0.47      0.59       401

    accuracy                           0.89      2466
   macro avg       0.84      0.72      0.76      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 2007
False positives: 58
False negatives: 211
True positives: 190

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.1min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 1, 'kernel': 'rbf'} with a score of 0.87
Accuracy with best hyperparameters: 0.8682076236820763
Confusion Matrix with best hyperparameters:
[[1844  221]
 [ 104  297]]
              precision    recall  f1-score   support

           0       0.95      0.89      0.92      2065
           1       0.57      0.74      0.65       401

    accuracy                           0.87      2466
   macro avg       0.76      0.82      0.78      2466
weighted avg       0.89      0.87      0.87      2466

True negatives: 1844
False positives: 221
False negatives: 104
True positives: 297

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8876723438767234
Confusion Matrix:
[[2018   47]
 [ 230  171]]
              precision    recall  f1-score   support

           0       0.90      0.98      0.94      2065
           1       0.78      0.43      0.55       401

    accuracy                           0.89      2466
   macro avg       0.84      0.70      0.74      2466
weighted avg       0.88      0.89      0.87      2466

True negatives: 2018
False positives: 47
False negatives: 230
True positives: 171

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   35.5s finished
```

```
################ amount of features: 6 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.90
Accuracy with best hyperparameters: 0.8941605839416058
Confusion Matrix with best hyperparameters:
[[2000   65]
 [ 196  205]]
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      2065
           1       0.76      0.51      0.61       401

    accuracy                           0.89      2466
   macro avg       0.84      0.74      0.77      2466
weighted avg       0.89      0.89      0.89      2466

True negatives: 2000
False positives: 65
False negatives: 196
True positives: 205

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.0min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 10, 'kernel': 'rbf'} with a score of 0.87
Accuracy with best hyperparameters: 0.8673965936739659
Confusion Matrix with best hyperparameters:
[[1833  232]
 [  95  306]]
              precision    recall  f1-score   support

           0       0.95      0.89      0.92      2065
           1       0.57      0.76      0.65       401

    accuracy                           0.87      2466
   macro avg       0.76      0.83      0.78      2466
weighted avg       0.89      0.87      0.87      2466

True negatives: 1833
False positives: 232
False negatives: 95
True positives: 306

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8888888888888888
Confusion Matrix:
[[2013   52]
 [ 222  179]]
              precision    recall  f1-score   support

           0       0.90      0.97      0.94      2065
           1       0.77      0.45      0.57       401

    accuracy                           0.89      2466
   macro avg       0.84      0.71      0.75      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 2013
False positives: 52
False negatives: 222
True positives: 179

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   31.5s finished
```

```
################ amount of features: 5 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.90
Accuracy with best hyperparameters: 0.8929440389294404
Confusion Matrix with best hyperparameters:
[[1996   69]
 [ 195  206]]
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      2065
           1       0.75      0.51      0.61       401

    accuracy                           0.89      2466
   macro avg       0.83      0.74      0.77      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 1996
False positives: 69
False negatives: 195
True positives: 206

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   58.8s finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 1, 'kernel': 'rbf'} with a score of 0.87
Accuracy with best hyperparameters: 0.8726682887266829
Confusion Matrix with best hyperparameters:
[[1852  213]
 [ 101  300]]
              precision    recall  f1-score   support

           0       0.95      0.90      0.92      2065
           1       0.58      0.75      0.66       401

    accuracy                           0.87      2466
   macro avg       0.77      0.82      0.79      2466
weighted avg       0.89      0.87      0.88      2466

True negatives: 1852
False positives: 213
False negatives: 101
True positives: 300

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8868613138686131
Confusion Matrix:
[[1996   69]
 [ 210  191]]
              precision    recall  f1-score   support

           0       0.90      0.97      0.93      2065
           1       0.73      0.48      0.58       401

    accuracy                           0.89      2466
   macro avg       0.82      0.72      0.76      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 1996
False positives: 69
False negatives: 210
True positives: 191

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   32.3s finished
```

```
################ amount of features: 4 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.90
Accuracy with best hyperparameters: 0.8933495539334956
Confusion Matrix with best hyperparameters:
[[1996   69]
 [ 194  207]]
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      2065
           1       0.75      0.52      0.61       401

    accuracy                           0.89      2466
   macro avg       0.83      0.74      0.77      2466
weighted avg       0.89      0.89      0.89      2466

True negatives: 1996
False positives: 69
False negatives: 194
True positives: 207

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   58.5s finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.87
Accuracy with best hyperparameters: 0.8698296836982968
Confusion Matrix with best hyperparameters:
[[1850  215]
 [ 106  295]]
              precision    recall  f1-score   support

           0       0.95      0.90      0.92      2065
           1       0.58      0.74      0.65       401

    accuracy                           0.87      2466
   macro avg       0.76      0.82      0.78      2466
weighted avg       0.89      0.87      0.88      2466

True negatives: 1850
False positives: 215
False negatives: 106
True positives: 295

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8876723438767234
Confusion Matrix:
[[1990   75]
 [ 202  199]]
              precision    recall  f1-score   support

           0       0.91      0.96      0.93      2065
           1       0.73      0.50      0.59       401

    accuracy                           0.89      2466
   macro avg       0.82      0.73      0.76      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 1990
False positives: 75
False negatives: 202
True positives: 199

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   35.5s finished
```

```
################ amount of features: 3 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.90
Accuracy with best hyperparameters: 0.894566098945661
Confusion Matrix with best hyperparameters:
[[1996   69]
 [ 191  210]]
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      2065
           1       0.75      0.52      0.62       401

    accuracy                           0.89      2466
   macro avg       0.83      0.75      0.78      2466
weighted avg       0.89      0.89      0.89      2466

True negatives: 1996
False positives: 69
False negatives: 191
True positives: 210

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.0min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.87
Accuracy with best hyperparameters: 0.8718572587185726
Confusion Matrix with best hyperparameters:
[[1855  210]
 [ 106  295]]
              precision    recall  f1-score   support

           0       0.95      0.90      0.92      2065
           1       0.58      0.74      0.65       401

    accuracy                           0.87      2466
   macro avg       0.77      0.82      0.79      2466
weighted avg       0.89      0.87      0.88      2466

True negatives: 1855
False positives: 210
False negatives: 106
True positives: 295

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8876723438767234
Confusion Matrix:
[[1989   76]
 [ 201  200]]
              precision    recall  f1-score   support

           0       0.91      0.96      0.93      2065
           1       0.72      0.50      0.59       401

    accuracy                           0.89      2466
   macro avg       0.82      0.73      0.76      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 1989
False positives: 76
False negatives: 201
True positives: 200

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   24.6s finished
```

```
################ amount of features: 2 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.89
Accuracy with best hyperparameters: 0.889294403892944
Confusion Matrix with best hyperparameters:
[[1975   90]
 [ 183  218]]
              precision    recall  f1-score   support

           0       0.92      0.96      0.94      2065
           1       0.71      0.54      0.61       401

    accuracy                           0.89      2466
   macro avg       0.81      0.75      0.78      2466
weighted avg       0.88      0.89      0.88      2466

True negatives: 1975
False positives: 90
False negatives: 183
True positives: 218

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.2min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.89
Accuracy with best hyperparameters: 0.8856447688564477
Confusion Matrix with best hyperparameters:
[[1914  151]
 [ 131  270]]
              precision    recall  f1-score   support

           0       0.94      0.93      0.93      2065
           1       0.64      0.67      0.66       401

    accuracy                           0.89      2466
   macro avg       0.79      0.80      0.79      2466
weighted avg       0.89      0.89      0.89      2466

True negatives: 1914
False positives: 151
False negatives: 131
True positives: 270

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8848337388483374
Confusion Matrix:
[[1973   92]
 [ 192  209]]
              precision    recall  f1-score   support

           0       0.91      0.96      0.93      2065
           1       0.69      0.52      0.60       401

    accuracy                           0.88      2466
   macro avg       0.80      0.74      0.76      2466
weighted avg       0.88      0.88      0.88      2466

True negatives: 1973
False positives: 92
False negatives: 192
True positives: 209

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   43.0s finished
```

```
################ amount of features: 1 ##############################
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC ##################
The best parameters are {'C': 1, 'kernel': 'rbf'} with a score of 0.89
Accuracy with best hyperparameters: 0.8896999188969992
Confusion Matrix with best hyperparameters:
[[1961  104]
 [ 168  233]]
              precision    recall  f1-score   support

           0       0.92      0.95      0.94      2065
           1       0.69      0.58      0.63       401

    accuracy                           0.89      2466
   macro avg       0.81      0.77      0.78      2466
weighted avg       0.88      0.89      0.89      2466

True negatives: 1961
False positives: 104
False negatives: 168
True positives: 233

Fitting 5 folds for each of 8 candidates, totalling 40 fits



[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   38.1s finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.88
Accuracy with best hyperparameters: 0.8775344687753447
Confusion Matrix with best hyperparameters:
[[1858  207]
 [  95  306]]
              precision    recall  f1-score   support

           0       0.95      0.90      0.92      2065
           1       0.60      0.76      0.67       401

    accuracy                           0.88      2466
   macro avg       0.77      0.83      0.80      2466
weighted avg       0.89      0.88      0.88      2466

True negatives: 1858
False positives: 207
False negatives: 95
True positives: 306

#########PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf)##################
Accuracy : 0.8896999188969992
Confusion Matrix:
[[1961  104]
 [ 168  233]]
              precision    recall  f1-score   support

           0       0.92      0.95      0.94      2065
           1       0.69      0.58      0.63       401

    accuracy                           0.89      2466
   macro avg       0.81      0.77      0.78      2466
weighted avg       0.88      0.89      0.89      2466

True negatives: 1961
False positives: 104
False negatives: 168
True positives: 233
```

Deep Autoencoder

In [13]:

```
from keras.layers import Input, Dense
from keras.models import Model, Sequential
from keras import regularizers
from keras.layers import Input, Dense, Conv2D, MaxPooling2D, UpSampling2D
from keras import backend as K

#scaler = MinMaxScaler()
```

```
#X_scaled = scaler.fit_transform(X)
#X_nonbuyer_scaled = X_scaled[Y == 0]
#X_buyer_scaled = X_scaled[Y == 1]
i = 1
for i in range(1,11):

    input_layer = Input(shape =(X.shape[1], ))

    #Encoder network Deep
    encoded_deep = Dense(100, activation ='tanh')(input_layer)
    encoded_deep = Dense(50, activation ='tanh')(encoded_deep)
    encoded_deep = Dense(25, activation ='tanh')(encoded_deep)
    encoded_deep = Dense(12, activation ='tanh')(encoded_deep)
    encoded_deep = Dense(i, activation ='relu')(encoded_deep)

    #Decoder network deep
    decoded_deep = Dense(12, activation ='tanh')(encoded_deep)
    decoded_deep = Dense(25, activation ='tanh')(decoded_deep)
    decoded_deep = Dense(50, activation ='tanh')(decoded_deep)
    decoded_deep = Dense(100, activation ='tanh')(decoded_deep)

    output_layer_deep = Dense(X.shape[1], activation ='relu')(decoded_deep)



        # Defining the parameters of the Auto-encoder network
    encoder = Model(input_layer, encoded_deep)
    autoencoder = Model(input_layer, output_layer_deep)
    autoencoder.compile(optimizer ="adadelta", loss ="mse")
    autoencoder.summary()

        # Training the Auto-encoder network
    autoencoder.fit(datasets["17"]["X_train"],datasets["17"]["X_train"],
                        batch_size = 3000, epochs = 50,
                        shuffle = True, validation_split = 0.2, verbose=0)


    hidden_representation = Sequential()
    hidden_representation.add(autoencoder.layers[0])
    hidden_representation.add(autoencoder.layers[1])
    hidden_representation.add(autoencoder.layers[2])
    hidden_representation.add(autoencoder.layers[3])
    hidden_representation.add(autoencoder.layers[4])
    hidden_representation.add(autoencoder.layers[5])

        # Separating the points encoded by the Auto-encoder as normal and fraud
        #nonbuyer_hidden_rep = hidden_representation.predict(X_nonbuyer_scaled)
        #buyer_hidden_rep = hidden_representation.predict(X_buyer_scaled)

        # Combining the encoded points into a single table
        #encoded_X = np.append(nonbuyer_hidden_rep, buyer_hidden_rep, axis = 0)
        #y_nonbuyer = np.zeros(nonbuyer_hidden_rep.shape[0])
        #y_buyer = np.ones(buyer_hidden_rep.shape[0])
        #encoded_y = np.append(y_nonbuyer, y_buyer)
    encoded_X_train = hidden_representation.predict(datasets["17"]["X_train"])
    encoded_X_test = hidden_representation.predict(datasets["17"]["X_test"])
    print("encoded_X_train", encoded_X_train.shape)
    print("encoded_X_test", encoded_X_test.shape)

#     encoded_X_train = encoder.predict(datasets["17"]["X_train"])
#     encoded_X_test = encoder.predict(datasets["17"]["X_test"])


    classifier = SVC()
```

```
    classifier.fit(encoded_X_train, y_train)

        # Storing the predictions of the linear model
    y_pred_classifier = classifier.predict(encoded_X_test)


        # Plotting the encoded points
        #tsne_plot(encoded_X, encoded_y)


        # Performance
    print('amount of features: '+ str(i))
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')

    i = i + 1
```

```
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               1800
_________________________________________________________________
dense_2 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_3 (Dense)              (None, 25)                1275
_________________________________________________________________
dense_4 (Dense)              (None, 12)                312
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 13
_________________________________________________________________
dense_6 (Dense)              (None, 12)                24
_________________________________________________________________
dense_7 (Dense)              (None, 25)                325
_________________________________________________________________
dense_8 (Dense)              (None, 50)                1300
_________________________________________________________________
dense_9 (Dense)              (None, 100)               5100
_________________________________________________________________
dense_10 (Dense)             (None, 17)                1717
=================================================================
Total params: 16,916
Trainable params: 16,916
Non-trainable params: 0
_________________________________________________________________


Using TensorFlow backend.
```

```
encoded_X_train (9864, 1)
encoded_X_test (2466, 1)
amount of features: 1
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_11 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_12 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_13 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_14 (Dense)             (None, 12)                312
_________________________________________________________________
dense_15 (Dense)             (None, 2)                 26
_________________________________________________________________
dense_16 (Dense)             (None, 12)                36
_________________________________________________________________
dense_17 (Dense)             (None, 25)                325
_________________________________________________________________
dense_18 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_19 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_20 (Dense)             (None, 17)                1717
=================================================================
Total params: 16,941
Trainable params: 16,941
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 2)
encoded_X_test (2466, 2)
amount of features: 2
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_21 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_22 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_23 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_24 (Dense)             (None, 12)                312
_________________________________________________________________
dense_25 (Dense)             (None, 3)                 39
_________________________________________________________________
dense_26 (Dense)             (None, 12)                48
_________________________________________________________________
dense_27 (Dense)             (None, 25)                325
_________________________________________________________________
dense_28 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_29 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_30 (Dense)             (None, 17)                1717
=================================================================
Total params: 16,966
Trainable params: 16,966
Non-trainable params: 0
```

```
_________________________________________________________________
encoded_X_train (9864, 3)
encoded_X_test (2466, 3)
amount of features: 3
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_31 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_32 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_33 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_34 (Dense)             (None, 12)                312
_________________________________________________________________
dense_35 (Dense)             (None, 4)                 52
_________________________________________________________________
dense_36 (Dense)             (None, 12)                60
_________________________________________________________________
dense_37 (Dense)             (None, 25)                325
_________________________________________________________________
dense_38 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_39 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_40 (Dense)             (None, 17)                1717
=================================================================
Total params: 16,991
Trainable params: 16,991
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 4)
encoded_X_test (2466, 4)
amount of features: 4
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_5 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_41 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_42 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_43 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_44 (Dense)             (None, 12)                312
_________________________________________________________________
dense_45 (Dense)             (None, 5)                 65
_________________________________________________________________
dense_46 (Dense)             (None, 12)                72
_________________________________________________________________
dense_47 (Dense)             (None, 25)                325
_________________________________________________________________
dense_48 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_49 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_50 (Dense)             (None, 17)                1717
=================================================================
Total params: 17,016
Trainable params: 17,016
Non-trainable params: 0
```

```
_________________________________________________________________
encoded_X_train (9864, 5)
encoded_X_test (2466, 5)
amount of features: 5
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_12"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_51 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_52 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_53 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_54 (Dense)             (None, 12)                312
_________________________________________________________________
dense_55 (Dense)             (None, 6)                 78
_________________________________________________________________
dense_56 (Dense)             (None, 12)                84
_________________________________________________________________
dense_57 (Dense)             (None, 25)                325
_________________________________________________________________
dense_58 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_59 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_60 (Dense)             (None, 17)                1717
=================================================================
Total params: 17,041
Trainable params: 17,041
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 6)
encoded_X_test (2466, 6)
amount of features: 6
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_14"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_7 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_61 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_62 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_63 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_64 (Dense)             (None, 12)                312
_________________________________________________________________
dense_65 (Dense)             (None, 7)                 91
_________________________________________________________________
dense_66 (Dense)             (None, 12)                96
_________________________________________________________________
dense_67 (Dense)             (None, 25)                325
_________________________________________________________________
dense_68 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_69 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_70 (Dense)             (None, 17)                1717
=================================================================
Total params: 17,066
Trainable params: 17,066
Non-trainable params: 0
```

```
_________________________________________________________________
encoded_X_train (9864, 7)
encoded_X_test (2466, 7)
amount of features: 7
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_8 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_71 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_72 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_73 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_74 (Dense)             (None, 12)                312
_________________________________________________________________
dense_75 (Dense)             (None, 8)                 104
_________________________________________________________________
dense_76 (Dense)             (None, 12)                108
_________________________________________________________________
dense_77 (Dense)             (None, 25)                325
_________________________________________________________________
dense_78 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_79 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_80 (Dense)             (None, 17)                1717
=================================================================
Total params: 17,091
Trainable params: 17,091
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 8)
encoded_X_test (2466, 8)
amount of features: 8
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_18"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_9 (InputLayer)         (None, 17)                0
_________________________________________________________________
dense_81 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_82 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_83 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_84 (Dense)             (None, 12)                312
_________________________________________________________________
dense_85 (Dense)             (None, 9)                 117
_________________________________________________________________
dense_86 (Dense)             (None, 12)                120
_________________________________________________________________
dense_87 (Dense)             (None, 25)                325
_________________________________________________________________
dense_88 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_89 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_90 (Dense)             (None, 17)                1717
=================================================================
Total params: 17,116
Trainable params: 17,116
Non-trainable params: 0
```

```
_________________________________________________________________
encoded_X_train (9864, 9)
encoded_X_test (2466, 9)
amount of features: 9
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_20"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_10 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_91 (Dense)             (None, 100)               1800
_________________________________________________________________
dense_92 (Dense)             (None, 50)                5050
_________________________________________________________________
dense_93 (Dense)             (None, 25)                1275
_________________________________________________________________
dense_94 (Dense)             (None, 12)                312
_________________________________________________________________
dense_95 (Dense)             (None, 10)                130
_________________________________________________________________
dense_96 (Dense)             (None, 12)                132
_________________________________________________________________
dense_97 (Dense)             (None, 25)                325
_________________________________________________________________
dense_98 (Dense)             (None, 50)                1300
_________________________________________________________________
dense_99 (Dense)             (None, 100)               5100
_________________________________________________________________
dense_100 (Dense)            (None, 17)                1717
=================================================================
Total params: 17,141
Trainable params: 17,141
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 10)
encoded_X_test (2466, 10)
amount of features: 10
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0
```

Deep Autoencoder with the best hyperparameters of SVC

In [14]:

```
# added by Duc, hyper parameter tuning for SVC
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
# hyperparameter grid set for finetuning
# param_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel': ['rbf', 'poly', 'si
gmoid']}
param_grid = {'C': [0.1,1, 10, 100], 'kernel': ['rbf', 'sigmoid']}
from keras.layers import Input, Dense
from keras.models import Model, Sequential
from keras import regularizers
from keras.layers import Input, Dense, Conv2D, MaxPooling2D, UpSampling2D
from keras import backend as K

#scaler = MinMaxScaler()

#X_scaled = scaler.fit_transform(X)
#X_nonbuyer_scaled = X_scaled[Y == 0]
```

```
#X_buyer_scaled = X_scaled[Y == 1]
i = 1
for i in range(1,11):

    input_layer = Input(shape =(X.shape[1], ))

    #Encoder network Deep
    encoded_deep = Dense(100, activation ='tanh')(input_layer)
    encoded_deep = Dense(50, activation ='tanh')(encoded_deep)
    encoded_deep = Dense(25, activation ='tanh')(encoded_deep)
    encoded_deep = Dense(12, activation ='tanh')(encoded_deep)
    encoded_deep = Dense(i, activation ='relu')(encoded_deep)

    #Decoder network deep
    decoded_deep = Dense(12, activation ='tanh')(encoded_deep)
    decoded_deep = Dense(25, activation ='tanh')(decoded_deep)
    decoded_deep = Dense(50, activation ='tanh')(decoded_deep)
    decoded_deep = Dense(100, activation ='tanh')(decoded_deep)

    output_layer_deep = Dense(X.shape[1], activation ='relu')(decoded_deep)



        # Defining the parameters of the Auto-encoder network
    autoencoder = Model(input_layer, output_layer_deep)
    autoencoder.compile(optimizer ="adadelta", loss ="mse")

        # Training the Auto-encoder network
    autoencoder.fit(datasets["17"]["X_train"],datasets["17"]["X_train"],
                    batch_size = 3000, epochs = 50,
                    shuffle = True, validation_split = 0.2, verbose=0)


    hidden_representation = Sequential()
    hidden_representation.add(autoencoder.layers[0])
    hidden_representation.add(autoencoder.layers[1])
    hidden_representation.add(autoencoder.layers[2])
    hidden_representation.add(autoencoder.layers[3])
    hidden_representation.add(autoencoder.layers[4])
    hidden_representation.add(autoencoder.layers[5])

        # Separating the points encoded by the Auto-encoder as normal and fraud
        #nonbuyer_hidden_rep = hidden_representation.predict(X_nonbuyer_scaled)
        #buyer_hidden_rep = hidden_representation.predict(X_buyer_scaled)

        # Combining the encoded points into a single table
        #encoded_X = np.append(nonbuyer_hidden_rep, buyer_hidden_rep, axis = 0)
        #y_nonbuyer = np.zeros(nonbuyer_hidden_rep.shape[0])
        #y_buyer = np.ones(buyer_hidden_rep.shape[0])
        #encoded_y = np.append(y_nonbuyer, y_buyer)
    encoded_X_train = hidden_representation.predict(datasets["17"]["X_train"])
    encoded_X_test = hidden_representation.predict(datasets["17"]["X_test"])
    print("encoded_X_train", encoded_X_train.shape)
    print("encoded_X_test", encoded_X_test.shape)
################################## section below is modified by Duc
    # Performance
    classifier = SVC()
    classifier.fit(encoded_X_train, y_train)

    y_pred_classifier = classifier.predict(encoded_X_test)

    print('######################### amount of features: '+ str(i) + ' ##########################
#######')
    print('######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-
RESCALED ##################')
```

```
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    print(classification_report(y_test,y_pred_classifier))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#####################################
    # Performance
    # rescale encoded features
    scaler = MinMaxScaler()
    scaler.fit(encoded_X_train)
    encoded_X_train = scaler.transform(encoded_X_train)
    encoded_X_test = scaler.transform(encoded_X_test)

    print('######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESC
ALED ##################')
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    print(classification_report(y_test,y_pred_classifier))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#####################################
    # Performance
    classifier = SVC()
    classifier_grid = GridSearchCV(classifier,param_grid,refit=True,verbose=1)
    classifier_grid.fit(encoded_X_train, y_train)

    y_pred_classifier_grid = classifier_grid.predict(encoded_X_test)

    print('######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ########
##########')
    print("The best parameters are %s with a score of %0.2f" % (classifier_grid.best_params_, cl
assifier_grid.best_score_))
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier_grid)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier_grid)))
    print(classification_report(y_test,y_pred_classifier_grid))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier_grid).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#####################################
    # Performance
    wsvmclf = SVC(class_weight='balanced')
    wsvmclf_grid = GridSearchCV(wsvmclf,param_grid,refit=True,verbose=1)
    wsvmclf_grid.fit(encoded_X_train, y_train)

    y_pred_wsvmclf_grid = wsvmclf_grid.predict(encoded_X_test)

    print('##########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ########
##########')
    print("The best parameters are %s with a score of %0.2f" % (wsvmclf_grid.best_params_, wsvmc
lf_grid.best_score_))
    print('Accuracy with best hyperparameters: '+str(accuracy_score(y_test, y_pred_wsvmclf_grid
)))
    print('Confusion Matrix with best hyperparameters: \n' + str(confusion_matrix(y_test,y_pred_
wsvmclf_grid)))
    print(classification_report(y_test,y_pred_wsvmclf_grid))
    # incorrect order of output
#         tn, fn, fp, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()
    # the right order is as follows
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_wsvmclf_grid).ravel()
```

```
    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')


    i = i + 1
```

```
encoded_X_train (9864, 1)
encoded_X_test (2466, 1)
######################## amount of features: 1 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED #################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED #################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   29.5s finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits
```

```
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.1min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 10, 'kernel': 'rbf'} with a score of 0.57
Accuracy with best hyperparameters: 0.5632603406326034
Confusion Matrix with best hyperparameters:
[[1115  950]
 [ 127  274]]
              precision    recall  f1-score   support

           0       0.90      0.54      0.67      2065
           1       0.22      0.68      0.34       401

    accuracy                           0.56      2466
   macro avg       0.56      0.61      0.51      2466
weighted avg       0.79      0.56      0.62      2466

True negatives: 1115
False positives: 950
False negatives: 127
True positives: 274

encoded_X_train (9864, 2)
encoded_X_test (2466, 2)
######################## amount of features: 2 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits



/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.2min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.6min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.55
Accuracy with best hyperparameters: 0.5437956204379562
Confusion Matrix with best hyperparameters:
[[1062 1003]
 [ 122  279]]
              precision    recall  f1-score   support

           0       0.90      0.51      0.65      2065
           1       0.22      0.70      0.33       401

    accuracy                           0.54      2466
   macro avg       0.56      0.61      0.49      2466
weighted avg       0.79      0.54      0.60      2466

True negatives: 1062
False positives: 1003
False negatives: 122
True positives: 279

encoded_X_train (9864, 3)
encoded_X_test (2466, 3)
####################### amount of features: 3 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.56
Accuracy with best hyperparameters: 0.5523114355231143
Confusion Matrix with best hyperparameters:
[[1034 1031]
 [  73  328]]
              precision    recall  f1-score   support

           0       0.93      0.50      0.65      2065
           1       0.24      0.82      0.37       401

    accuracy                           0.55      2466
   macro avg       0.59      0.66      0.51      2466
weighted avg       0.82      0.55      0.61      2466

True negatives: 1034
False positives: 1031
False negatives: 73
True positives: 328

encoded_X_train (9864, 4)
encoded_X_test (2466, 4)
####################### amount of features: 4 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 1, 'kernel': 'rbf'} with a score of 0.51
Accuracy with best hyperparameters: 0.5198702351987023
Confusion Matrix with best hyperparameters:
[[ 954 1111]
 [  73  328]]
              precision    recall  f1-score   support

           0       0.93      0.46      0.62      2065
           1       0.23      0.82      0.36       401

    accuracy                           0.52      2466
   macro avg       0.58      0.64      0.49      2466
weighted avg       0.81      0.52      0.57      2466

True negatives: 954
False positives: 1111
False negatives: 73
True positives: 328

encoded_X_train (9864, 5)
encoded_X_test (2466, 5)
####################### amount of features: 5 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.9min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8390105433901054
Confusion Matrix:
[[2063    2]
 [ 395    6]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.75      0.01      0.03       401

    accuracy                           0.84      2466
   macro avg       0.79      0.51      0.47      2466
weighted avg       0.82      0.84      0.77      2466

True negatives: 2063
False positives: 2
False negatives: 395
True positives: 6

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.65
Accuracy with best hyperparameters: 0.6382806163828062
Confusion Matrix with best hyperparameters:
[[1286  779]
 [ 113  288]]
              precision    recall  f1-score   support

           0       0.92      0.62      0.74      2065
           1       0.27      0.72      0.39       401

    accuracy                           0.64      2466
   macro avg       0.59      0.67      0.57      2466
weighted avg       0.81      0.64      0.69      2466

True negatives: 1286
False positives: 779
False negatives: 113
True positives: 288

encoded_X_train (9864, 6)
encoded_X_test (2466, 6)
####################### amount of features: 6 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.1min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8418491484184915
Confusion Matrix:
[[2063    2]
 [ 388   13]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.87      0.03      0.06       401

    accuracy                           0.84      2466
   macro avg       0.85      0.52      0.49      2466
weighted avg       0.85      0.84      0.78      2466

True negatives: 2063
False positives: 2
False negatives: 388
True positives: 13

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.66
Accuracy with best hyperparameters: 0.6666666666666666
Confusion Matrix with best hyperparameters:
[[1352  713]
 [ 109  292]]
              precision    recall  f1-score   support

           0       0.93      0.65      0.77      2065
           1       0.29      0.73      0.42       401

    accuracy                           0.67      2466
   macro avg       0.61      0.69      0.59      2466
weighted avg       0.82      0.67      0.71      2466

True negatives: 1352
False positives: 713
False negatives: 109
True positives: 292

encoded_X_train (9864, 7)
encoded_X_test (2466, 7)
####################### amount of features: 7 ##############################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   57.1s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8454987834549879
Confusion Matrix:
[[2061    4]
 [ 377   24]]
              precision    recall  f1-score   support

           0       0.85      1.00      0.92      2065
           1       0.86      0.06      0.11       401

    accuracy                           0.85      2466
   macro avg       0.85      0.53      0.51      2466
weighted avg       0.85      0.85      0.78      2466

True negatives: 2061
False positives: 4
False negatives: 377
True positives: 24

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.66
Accuracy with best hyperparameters: 0.670316301703163
Confusion Matrix with best hyperparameters:
[[1362  703]
 [ 110  291]]
              precision    recall  f1-score   support

           0       0.93      0.66      0.77      2065
           1       0.29      0.73      0.42       401

    accuracy                           0.67      2466
   macro avg       0.61      0.69      0.59      2466
weighted avg       0.82      0.67      0.71      2466

True negatives: 1362
False positives: 703
False negatives: 110
True positives: 291

encoded_X_train (9864, 8)
encoded_X_test (2466, 8)
####################### amount of features: 8 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  2.0min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.62
Accuracy with best hyperparameters: 0.6224655312246553
Confusion Matrix with best hyperparameters:
[[1223  842]
 [  89  312]]
              precision    recall  f1-score   support

           0       0.93      0.59      0.72      2065
           1       0.27      0.78      0.40       401

    accuracy                           0.62      2466
   macro avg       0.60      0.69      0.56      2466
weighted avg       0.82      0.62      0.67      2466

True negatives: 1223
False positives: 842
False negatives: 89
True positives: 312

encoded_X_train (9864, 9)
encoded_X_test (2466, 9)
####################### amount of features: 9 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   48.7s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.86
Accuracy : 0.8523925385239254
Confusion Matrix:
[[2049   16]
 [ 348   53]]
              precision    recall  f1-score   support

           0       0.85      0.99      0.92      2065
           1       0.77      0.13      0.23       401

    accuracy                           0.85      2466
   macro avg       0.81      0.56      0.57      2466
weighted avg       0.84      0.85      0.81      2466

True negatives: 2049
False positives: 16
False negatives: 348
True positives: 53

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  2.0min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.70
Accuracy with best hyperparameters: 0.7043795620437956
Confusion Matrix with best hyperparameters:
[[1438  627]
 [ 102  299]]
              precision    recall  f1-score   support

           0       0.93      0.70      0.80      2065
           1       0.32      0.75      0.45       401

    accuracy                           0.70      2466
   macro avg       0.63      0.72      0.62      2466
weighted avg       0.83      0.70      0.74      2466

True negatives: 1438
False positives: 627
False negatives: 102
True positives: 299

encoded_X_train (9864, 10)
encoded_X_test (2466, 10)
####################### amount of features: 10 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.0min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8369829683698297
Confusion Matrix:
[[2063    2]
 [ 400    1]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.33      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.59      0.50      0.46      2466
weighted avg       0.76      0.84      0.76      2466

True negatives: 2063
False positives: 2
False negatives: 400
True positives: 1

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.5min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.63
Accuracy with best hyperparameters: 0.6545012165450121
Confusion Matrix with best hyperparameters:
[[1310  755]
 [  97  304]]
              precision    recall  f1-score   support

           0       0.93      0.63      0.75      2065
           1       0.29      0.76      0.42       401

    accuracy                           0.65      2466
   macro avg       0.61      0.70      0.59      2466
weighted avg       0.83      0.65      0.70      2466

True negatives: 1310
False positives: 755
False negatives: 97
True positives: 304
```

Sparse Autoencoder

In [15]:

```
i = 1
for i in range(1,11):


    input_layer_sparse = Input(shape =(X.shape[1], ))



    #Encoder network Sparsity constraint
    encoded_sparse = Dense(100, activation ='tanh',
    activity_regularizer = regularizers.l1(10e-5))(input_layer_sparse)
    encoded_sparse = Dense(50, activation ='tanh',
    activity_regularizer = regularizers.l1(10e-5))(encoded_sparse)
    encoded_sparse = Dense(25, activation ='tanh',
    activity_regularizer = regularizers.l1(10e-5))(encoded_sparse)
    encoded_sparse = Dense(12, activation ='tanh',
    activity_regularizer = regularizers.l1(10e-5))(encoded_sparse)
    encoded_sparse = Dense(i, activation ='relu')(encoded_sparse)

    #Decoder network Sparsity constraint
    decoded_sparse = Dense(12, activation ='tanh')(encoded_sparse)
    decoded_sparse = Dense(25, activation ='tanh')(decoded_sparse)
```

```
    decoded_sparse = Dense(50, activation ='tanh')(decoded_sparse)
    decoded_sparse = Dense(100, activation ='tanh')(decoded_sparse)

    output_layer_sparse = Dense(X.shape[1], activation ='relu')(decoded_sparse)



    # Defining the parameters of the Auto-encoder network
    autoencoder = Model(input_layer_sparse, output_layer_sparse)
    autoencoder.compile(optimizer ="adadelta", loss ="mse")
    autoencoder.summary()

    # Training the Auto-encoder network
    autoencoder.fit(datasets["17"]["X_train"], datasets["17"]["X_train"],
                        batch_size = 3000, epochs = 50,
                        shuffle = True, validation_split = 0.2, verbose=0)


    hidden_representation = Sequential()
    hidden_representation.add(autoencoder.layers[0])
    hidden_representation.add(autoencoder.layers[1])
    hidden_representation.add(autoencoder.layers[2])
    hidden_representation.add(autoencoder.layers[3])
    hidden_representation.add(autoencoder.layers[4])
    hidden_representation.add(autoencoder.layers[5])

        # Separating the points encoded by the Auto-encoder as normal and fraud
        #nonbuyer_hidden_rep = hidden_representation.predict(X_nonbuyer_scaled)
        #buyer_hidden_rep = hidden_representation.predict(X_buyer_scaled)

        # Combining the encoded points into a single table
        #encoded_X = np.append(nonbuyer_hidden_rep, buyer_hidden_rep, axis = 0)
        #y_nonbuyer = np.zeros(nonbuyer_hidden_rep.shape[0])
        #y_buyer = np.ones(buyer_hidden_rep.shape[0])
        #encoded_y = np.append(y_nonbuyer, y_buyer)
    encoded_X_train = hidden_representation.predict(datasets["17"]["X_train"])
    encoded_X_test = hidden_representation.predict(datasets["17"]["X_test"])
    print("encoded_X_train", encoded_X_train.shape)
    print("encoded_X_test", encoded_X_test.shape)


    classifier = SVC()
    classifier.fit(encoded_X_train, y_train)

        # Storing the predictions of the linear model
    y_pred_classifier = classifier.predict(encoded_X_test)


        # Plotting the encoded points
        #tsne_plot(encoded_X, encoded_y)


        # Performance
    print('amount of features: '+ str(i))
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
    i = i + 1
```

```
Model: "model_31"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_21 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_201 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_202 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_203 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_204 (Dense)            (None, 12)                312
_________________________________________________________________
dense_205 (Dense)            (None, 1)                 13
_________________________________________________________________
dense_206 (Dense)            (None, 12)                24
_________________________________________________________________
dense_207 (Dense)            (None, 25)                325
_________________________________________________________________
dense_208 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_209 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_210 (Dense)            (None, 17)                1717
=================================================================
Total params: 16,916
Trainable params: 16,916
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 1)
encoded_X_test (2466, 1)
amount of features: 1
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_32"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_22 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_211 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_212 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_213 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_214 (Dense)            (None, 12)                312
_________________________________________________________________
dense_215 (Dense)            (None, 2)                 26
_________________________________________________________________
dense_216 (Dense)            (None, 12)                36
_________________________________________________________________
dense_217 (Dense)            (None, 25)                325
_________________________________________________________________
dense_218 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_219 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_220 (Dense)            (None, 17)                1717
=================================================================
Total params: 16,941
Trainable params: 16,941
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 2)
encoded_X_test (2466, 2)
amount of features: 2
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0
```

```
Model: "model_33"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_23 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_221 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_222 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_223 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_224 (Dense)            (None, 12)                312
_________________________________________________________________
dense_225 (Dense)            (None, 3)                 39
_________________________________________________________________
dense_226 (Dense)            (None, 12)                48
_________________________________________________________________
dense_227 (Dense)            (None, 25)                325
_________________________________________________________________
dense_228 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_229 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_230 (Dense)            (None, 17)                1717
=================================================================
Total params: 16,966
Trainable params: 16,966
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 3)
encoded_X_test (2466, 3)
amount of features: 3
Accuracy : 0.83779399837794
Confusion Matrix:
[[2063    2]
 [ 398    3]]
True negatives: 2063
False positives: 2
False negatives: 398
True positives: 3

Model: "model_34"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_24 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_231 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_232 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_233 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_234 (Dense)            (None, 12)                312
_________________________________________________________________
dense_235 (Dense)            (None, 4)                 52
_________________________________________________________________
dense_236 (Dense)            (None, 12)                60
_________________________________________________________________
dense_237 (Dense)            (None, 25)                325
_________________________________________________________________
dense_238 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_239 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_240 (Dense)            (None, 17)                1717
=================================================================
Total params: 16,991
Trainable params: 16,991
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 4)
encoded_X_test (2466, 4)
amount of features: 4
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0
```

```
Model: "model_35"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_25 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_241 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_242 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_243 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_244 (Dense)            (None, 12)                312
_________________________________________________________________
dense_245 (Dense)            (None, 5)                 65
_________________________________________________________________
dense_246 (Dense)            (None, 12)                72
_________________________________________________________________
dense_247 (Dense)            (None, 25)                325
_________________________________________________________________
dense_248 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_249 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_250 (Dense)            (None, 17)                1717
=================================================================
Total params: 17,016
Trainable params: 17,016
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 5)
encoded_X_test (2466, 5)
amount of features: 5
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_36"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_26 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_251 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_252 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_253 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_254 (Dense)            (None, 12)                312
_________________________________________________________________
dense_255 (Dense)            (None, 6)                 78
_________________________________________________________________
dense_256 (Dense)            (None, 12)                84
_________________________________________________________________
dense_257 (Dense)            (None, 25)                325
_________________________________________________________________
dense_258 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_259 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_260 (Dense)            (None, 17)                1717
=================================================================
Total params: 17,041
Trainable params: 17,041
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 6)
encoded_X_test (2466, 6)
amount of features: 6
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0
```

```
Model: "model_37"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_27 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_261 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_262 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_263 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_264 (Dense)            (None, 12)                312
_________________________________________________________________
dense_265 (Dense)            (None, 7)                 91
_________________________________________________________________
dense_266 (Dense)            (None, 12)                96
_________________________________________________________________
dense_267 (Dense)            (None, 25)                325
_________________________________________________________________
dense_268 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_269 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_270 (Dense)            (None, 17)                1717
=================================================================
Total params: 17,066
Trainable params: 17,066
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 7)
encoded_X_test (2466, 7)
amount of features: 7
Accuracy : 0.8390105433901054
Confusion Matrix:
[[2065    0]
 [ 397    4]]
True negatives: 2065
False positives: 0
False negatives: 397
True positives: 4

Model: "model_38"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_28 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_271 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_272 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_273 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_274 (Dense)            (None, 12)                312
_________________________________________________________________
dense_275 (Dense)            (None, 8)                 104
_________________________________________________________________
dense_276 (Dense)            (None, 12)                108
_________________________________________________________________
dense_277 (Dense)            (None, 25)                325
_________________________________________________________________
dense_278 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_279 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_280 (Dense)            (None, 17)                1717
=================================================================
Total params: 17,091
Trainable params: 17,091
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 8)
encoded_X_test (2466, 8)
amount of features: 8
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0
```

```
Model: "model_39"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_29 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_281 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_282 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_283 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_284 (Dense)            (None, 12)                312
_________________________________________________________________
dense_285 (Dense)            (None, 9)                 117
_________________________________________________________________
dense_286 (Dense)            (None, 12)                120
_________________________________________________________________
dense_287 (Dense)            (None, 25)                325
_________________________________________________________________
dense_288 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_289 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_290 (Dense)            (None, 17)                1717
=================================================================
Total params: 17,116
Trainable params: 17,116
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 9)
encoded_X_test (2466, 9)
amount of features: 9
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_40"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_30 (InputLayer)        (None, 17)                0
_________________________________________________________________
dense_291 (Dense)            (None, 100)               1800
_________________________________________________________________
dense_292 (Dense)            (None, 50)                5050
_________________________________________________________________
dense_293 (Dense)            (None, 25)                1275
_________________________________________________________________
dense_294 (Dense)            (None, 12)                312
_________________________________________________________________
dense_295 (Dense)            (None, 10)                130
_________________________________________________________________
dense_296 (Dense)            (None, 12)                132
_________________________________________________________________
dense_297 (Dense)            (None, 25)                325
_________________________________________________________________
dense_298 (Dense)            (None, 50)                1300
_________________________________________________________________
dense_299 (Dense)            (None, 100)               5100
_________________________________________________________________
dense_300 (Dense)            (None, 17)                1717
=================================================================
Total params: 17,141
Trainable params: 17,141
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 10)
encoded_X_test (2466, 10)
amount of features: 10
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
```

```
True positives: 0
```

Sparse Autoencoder with the best hyperparameters of SVC

In [16]:

```
# added by Duc, hyper parameter tuning for SVC
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
# hyperparameter grid set for finetuning
# param_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel': ['rbf', 'poly', 'si
gmoid']}
param_grid = {'C': [0.1,1, 10, 100], 'kernel': ['rbf', 'sigmoid']}

i = 1
for i in range(1,11):


    input_layer_sparse = Input(shape =(X.shape[1], ))



    #Encoder network Sparsity constraint
    encoded_sparse = Dense(100, activation ='tanh',
    activity_regularizer = regularizers.l1(10e-5))(input_layer_sparse)
    encoded_sparse = Dense(50, activation ='tanh',
    activity_regularizer = regularizers.l1(10e-5))(encoded_sparse)
    encoded_sparse = Dense(25, activation ='tanh',
    activity_regularizer = regularizers.l1(10e-5))(encoded_sparse)
    encoded_sparse = Dense(12, activation ='tanh',
    activity_regularizer = regularizers.l1(10e-5))(encoded_sparse)
    encoded_sparse = Dense(i, activation ='relu')(encoded_sparse)

    #Decoder network Sparsity constraint
    decoded_sparse = Dense(12, activation ='tanh')(encoded_sparse)
    decoded_sparse = Dense(25, activation ='tanh')(decoded_sparse)
    decoded_sparse = Dense(50, activation ='tanh')(decoded_sparse)
    decoded_sparse = Dense(100, activation ='tanh')(decoded_sparse)

    output_layer_sparse = Dense(X.shape[1], activation ='relu')(decoded_sparse)



    # Defining the parameters of the Auto-encoder network
    autoencoder = Model(input_layer_sparse, output_layer_sparse)
    autoencoder.compile(optimizer ="adadelta", loss ="mse")

    # Training the Auto-encoder network
    autoencoder.fit(datasets["17"]["X_train"], datasets["17"]["X_train"],
                    batch_size = 3000, epochs = 50,
                    shuffle = True, validation_split = 0.2, verbose=0)


    hidden_representation = Sequential()
    hidden_representation.add(autoencoder.layers[0])
    hidden_representation.add(autoencoder.layers[1])
    hidden_representation.add(autoencoder.layers[2])
    hidden_representation.add(autoencoder.layers[3])
    hidden_representation.add(autoencoder.layers[4])
    hidden_representation.add(autoencoder.layers[5])

        # Separating the points encoded by the Auto-encoder as normal and fraud
        #nonbuyer_hidden_rep = hidden_representation.predict(X_nonbuyer_scaled)
```

```
        #buyer_hidden_rep = hidden_representation.predict(X_buyer_scaled)

        # Combining the encoded points into a single table
        #encoded_X = np.append(nonbuyer_hidden_rep, buyer_hidden_rep, axis = 0)
        #y_nonbuyer = np.zeros(nonbuyer_hidden_rep.shape[0])
        #y_buyer = np.ones(buyer_hidden_rep.shape[0])
        #encoded_y = np.append(y_nonbuyer, y_buyer)
    encoded_X_train = hidden_representation.predict(datasets["17"]["X_train"])
    encoded_X_test = hidden_representation.predict(datasets["17"]["X_test"])
    print("encoded_X_train", encoded_X_train.shape)
    print("encoded_X_test", encoded_X_test.shape)


    ################################### section below is modified by Duc
    # Performance
    classifier = SVC()
    classifier.fit(encoded_X_train, y_train)

    y_pred_classifier = classifier.predict(encoded_X_test)

    print('######################### amount of features: '+ str(i) + ' ##########################
#######')
    print('######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-
RESCALED ##################')
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    print(classification_report(y_test,y_pred_classifier))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#######################################
    # Performance
    # rescale encoded features
    scaler = MinMaxScaler()
    scaler.fit(encoded_X_train)
    encoded_X_train = scaler.transform(encoded_X_train)
    encoded_X_test = scaler.transform(encoded_X_test)

    print('######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESC
ALED ##################')
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    print(classification_report(y_test,y_pred_classifier))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
###################################
    # Performance
    classifier = SVC()
    classifier_grid = GridSearchCV(classifier,param_grid,refit=True,verbose=1)
    classifier_grid.fit(encoded_X_train, y_train)

    y_pred_classifier_grid = classifier_grid.predict(encoded_X_test)

    print('######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ########
##########')
    print("The best parameters are %s with a score of %0.2f" % (classifier_grid.best_params_, cl
assifier_grid.best_score_))
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier_grid)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier_grid)))
    print(classification_report(y_test,y_pred_classifier_grid))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier_grid).ravel()
```

```
    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
####################################
    # Performance
    wsvmclf = SVC(class_weight='balanced')
    wsvmclf_grid = GridSearchCV(wsvmclf,param_grid,refit=True,verbose=1)
    wsvmclf_grid.fit(encoded_X_train, y_train)

    y_pred_wsvmclf_grid = wsvmclf_grid.predict(encoded_X_test)

    print('##########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ########
##########')
    print("The best parameters are %s with a score of %0.2f" % (wsvmclf_grid.best_params_, wsvmc
lf_grid.best_score_))
    print('Accuracy with best hyperparameters: '+str(accuracy_score(y_test, y_pred_wsvmclf_grid
)))
    print('Confusion Matrix with best hyperparameters: \n' + str(confusion_matrix(y_test,y_pred_
wsvmclf_grid)))
    print(classification_report(y_test,y_pred_wsvmclf_grid))
    # incorrect order of output
#         tn, fn, fp, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()
    # the right order is as follows
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_wsvmclf_grid).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')

    i = i + 1
```

```
encoded_X_train (9864, 1)
encoded_X_test (2466, 1)
######################## amount of features: 1 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED #################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED #################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   41.1s finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits
```

```
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.7min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'sigmoid'} with a score of 0.82
Accuracy with best hyperparameters: 0.8029197080291971
Confusion Matrix with best hyperparameters:
[[1976   89]
 [ 397    4]]
              precision    recall  f1-score   support

           0       0.83      0.96      0.89      2065
           1       0.04      0.01      0.02       401

    accuracy                           0.80      2466
   macro avg       0.44      0.48      0.45      2466
weighted avg       0.70      0.80      0.75      2466

True negatives: 1976
False positives: 89
False negatives: 397
True positives: 4

encoded_X_train (9864, 2)
encoded_X_test (2466, 2)
######################### amount of features: 2 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   35.8s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.5min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.68
Accuracy with best hyperparameters: 0.6780210867802109
Confusion Matrix with best hyperparameters:
[[1498  567]
 [ 227  174]]
              precision    recall  f1-score   support

           0       0.87      0.73      0.79      2065
           1       0.23      0.43      0.30       401

    accuracy                           0.68      2466
   macro avg       0.55      0.58      0.55      2466
weighted avg       0.77      0.68      0.71      2466

True negatives: 1498
False positives: 567
False negatives: 227
True positives: 174

encoded_X_train (9864, 3)
encoded_X_test (2466, 3)
####################### amount of features: 3 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   30.6s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'sigmoid'} with a score of 0.79
Accuracy with best hyperparameters: 0.7935928629359287
Confusion Matrix with best hyperparameters:
[[1882  183]
 [ 326   75]]
              precision    recall  f1-score   support

           0       0.85      0.91      0.88      2065
           1       0.29      0.19      0.23       401

    accuracy                           0.79      2466
   macro avg       0.57      0.55      0.55      2466
weighted avg       0.76      0.79      0.77      2466

True negatives: 1882
False positives: 183
False negatives: 326
True positives: 75

encoded_X_train (9864, 4)
encoded_X_test (2466, 4)
####################### amount of features: 4 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   43.5s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8386050283860503
Confusion Matrix:
[[2065    0]
 [ 398    3]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       1.00      0.01      0.01       401

    accuracy                           0.84      2466
   macro avg       0.92      0.50      0.46      2466
weighted avg       0.86      0.84      0.77      2466

True negatives: 2065
False positives: 0
False negatives: 398
True positives: 3

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.5min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.79
Accuracy with best hyperparameters: 0.7777777777777778
Confusion Matrix with best hyperparameters:
[[1745  320]
 [ 228  173]]
              precision    recall  f1-score   support

           0       0.88      0.85      0.86      2065
           1       0.35      0.43      0.39       401

    accuracy                           0.78      2466
   macro avg       0.62      0.64      0.63      2466
weighted avg       0.80      0.78      0.79      2466

True negatives: 1745
False positives: 320
False negatives: 228
True positives: 173

encoded_X_train (9864, 5)
encoded_X_test (2466, 5)
####################### amount of features: 5 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   50.2s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8402270884022709
Confusion Matrix:
[[2060    5]
 [ 389   12]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.71      0.03      0.06       401

    accuracy                           0.84      2466
   macro avg       0.77      0.51      0.49      2466
weighted avg       0.82      0.84      0.77      2466

True negatives: 2060
False positives: 5
False negatives: 389
True positives: 12

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.5min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.77
Accuracy with best hyperparameters: 0.7631792376317924
Confusion Matrix with best hyperparameters:
[[1679  386]
 [ 198  203]]
              precision    recall  f1-score   support

           0       0.89      0.81      0.85      2065
           1       0.34      0.51      0.41       401

    accuracy                           0.76      2466
   macro avg       0.62      0.66      0.63      2466
weighted avg       0.81      0.76      0.78      2466

True negatives: 1679
False positives: 386
False negatives: 198
True positives: 203

encoded_X_train (9864, 6)
encoded_X_test (2466, 6)
####################### amount of features: 6 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  3.7min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8357664233576643
Confusion Matrix:
[[2061    4]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2061
False positives: 4
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 1, 'kernel': 'rbf'} with a score of 0.54
Accuracy with best hyperparameters: 0.5340632603406326
Confusion Matrix with best hyperparameters:
[[1075  990]
 [ 159  242]]
              precision    recall  f1-score   support

           0       0.87      0.52      0.65      2065
           1       0.20      0.60      0.30       401

    accuracy                           0.53      2466
   macro avg       0.53      0.56      0.47      2466
weighted avg       0.76      0.53      0.59      2466

True negatives: 1075
False positives: 990
False negatives: 159
True positives: 242

encoded_X_train (9864, 7)
encoded_X_test (2466, 7)
####################### amount of features: 7 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   50.4s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.5min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.77
Accuracy with best hyperparameters: 0.767639902676399
Confusion Matrix with best hyperparameters:
[[1766  299]
 [ 274  127]]
              precision    recall  f1-score   support

           0       0.87      0.86      0.86      2065
           1       0.30      0.32      0.31       401

    accuracy                           0.77      2466
   macro avg       0.58      0.59      0.58      2466
weighted avg       0.77      0.77      0.77      2466

True negatives: 1766
False positives: 299
False negatives: 274
True positives: 127

encoded_X_train (9864, 8)
encoded_X_test (2466, 8)
####################### amount of features: 8 ##############################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.2min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.6min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'sigmoid'} with a score of 0.76
Accuracy with best hyperparameters: 0.7522303325223033
Confusion Matrix with best hyperparameters:
[[1698  367]
 [ 244  157]]
              precision    recall  f1-score   support

           0       0.87      0.82      0.85      2065
           1       0.30      0.39      0.34       401

    accuracy                           0.75      2466
   macro avg       0.59      0.61      0.59      2466
weighted avg       0.78      0.75      0.76      2466

True negatives: 1698
False positives: 367
False negatives: 244
True positives: 157

encoded_X_train (9864, 9)
encoded_X_test (2466, 9)
####################### amount of features: 9 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  2.4min finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 10, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.83779399837794
Confusion Matrix:
[[2065    0]
 [ 400    1]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       1.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.92      0.50      0.46      2466
weighted avg       0.86      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 400
True positives: 1

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.7min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'sigmoid'} with a score of 0.71
Accuracy with best hyperparameters: 0.16342254663422548
Confusion Matrix with best hyperparameters:
[[   2 2063]
 [   0  401]]
              precision    recall  f1-score   support

           0       1.00      0.00      0.00      2065
           1       0.16      1.00      0.28       401

    accuracy                           0.16      2466
   macro avg       0.58      0.50      0.14      2466
weighted avg       0.86      0.16      0.05      2466

True negatives: 2
False positives: 2063
False negatives: 0
True positives: 401

encoded_X_train (9864, 10)
encoded_X_test (2466, 10)
######################## amount of features: 10 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8394160583941606
Confusion Matrix:
[[2065    0]
 [ 396    5]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       1.00      0.01      0.02       401

    accuracy                           0.84      2466
   macro avg       0.92      0.51      0.47      2466
weighted avg       0.87      0.84      0.77      2466

True negatives: 2065
False positives: 0
False negatives: 396
True positives: 5

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8394160583941606
Confusion Matrix:
[[2065    0]
 [ 396    5]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       1.00      0.01      0.02       401

    accuracy                           0.84      2466
   macro avg       0.92      0.51      0.47      2466
weighted avg       0.87      0.84      0.77      2466

True negatives: 2065
False positives: 0
False negatives: 396
True positives: 5

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   37.2s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.86
Accuracy : 0.8552311435523114
Confusion Matrix:
[[2061    4]
 [ 353   48]]
              precision    recall  f1-score   support

           0       0.85      1.00      0.92      2065
           1       0.92      0.12      0.21       401

    accuracy                           0.86      2466
   macro avg       0.89      0.56      0.57      2466
weighted avg       0.87      0.86      0.81      2466

True negatives: 2061
False positives: 4
False negatives: 353
True positives: 48

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.5min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.82
Accuracy with best hyperparameters: 0.8203568532035685
Confusion Matrix with best hyperparameters:
[[1819  246]
 [ 197  204]]
              precision    recall  f1-score   support

           0       0.90      0.88      0.89      2065
           1       0.45      0.51      0.48       401

    accuracy                           0.82      2466
   macro avg       0.68      0.69      0.69      2466
weighted avg       0.83      0.82      0.82      2466

True negatives: 1819
False positives: 246
False negatives: 197
True positives: 204
```

Convolutional Autoencoder

In [17]:

```
#Encoder network Convolutional
i = 1
for i in range(1, 11):


    #print(X.shape[0])
    X_train_conv = X_train.drop(columns =['OperatingSystems'])
    X_test_conv = X_test.drop(columns = ['OperatingSystems'])
    scaler = MinMaxScaler()
    scaler.fit(X_train_conv)
    X_train_scaled_conv = scaler.transform(X_train_conv)
    X_test_scaled_conv = scaler.transform(X_test_conv)
    #print(X_train_scaled_conv.shape[1])
    #print(X_test_scaled_conv.shape[1])
    #print("Train data size", X_train_scaled_conv.shape, y_train.shape)
    #print("Train test size", X_test_scaled_conv.shape, y_test.shape)
    X_train_scaled_conv = np.reshape(X_train_scaled_conv, (len(X_train_scaled_conv), 4, 4, 1))
# adapt this if using `channels_first` image data format
    X_test_scaled_conv = np.reshape(X_test_scaled_conv, (len(X_test_scaled_conv), 4, 4, 1))  # a
dapt this if using `channels_first` image data format

    input_layer = Input(shape =(4,4,1))

    x = Conv2D(16, (2, 2), activation='relu', padding='same')(input_layer)
```

```
    x = MaxPooling2D((2, 2), padding='same')(x)
    x = Conv2D(i, (2, 2), activation='relu', padding='same')(x)
    encoded_conv = MaxPooling2D((2, 2), padding='same')(x)

    # at this point the representation is (4, 4, 8) i.e. 128-dimensional

    x = Conv2D(i, (2, 2), activation='relu', padding='same')(encoded_conv)
    x = UpSampling2D((2, 2))(x)
    x = Conv2D(16, (2, 2), activation='relu', padding='same')(x)
    x = UpSampling2D((2, 2))(x)
    decoded_conv = Conv2D(1, (2, 2), activation='sigmoid', padding='same')(x)


    # Defining the parameters of the Auto-encoder network
    autoencoder = Model(input_layer, decoded_conv)
    autoencoder.compile(optimizer ="adadelta", loss ='mse')
    autoencoder.summary()
        # Training the Auto-encoder network
    autoencoder.fit(X_train_scaled_conv, X_train_scaled_conv,
                        batch_size = 3000, epochs = 50,
                        shuffle = True, validation_split = 0.2, verbose=0)

    encoder = Model(input_layer, encoded_conv)

    hidden_representation = Sequential()
    hidden_representation.add(autoencoder.layers[0])
    hidden_representation.add(autoencoder.layers[1])
    hidden_representation.add(autoencoder.layers[2])
    hidden_representation.add(autoencoder.layers[3])
    hidden_representation.add(autoencoder.layers[4])

        # Separating the points encoded by the Auto-encoder as normal and fraud
        #nonbuyer_hidden_rep = hidden_representation.predict(X_nonbuyer_scaled)
        #buyer_hidden_rep = hidden_representation.predict(X_buyer_scaled)

        # Combining the encoded points into a single table
        #encoded_X = np.append(nonbuyer_hidden_rep, buyer_hidden_rep, axis = 0)
        #y_nonbuyer = np.zeros(nonbuyer_hidden_rep.shape[0])
        #y_buyer = np.ones(buyer_hidden_rep.shape[0])
        #encoded_y = np.append(y_nonbuyer, y_buyer)
    encoded_X_train = hidden_representation.predict(X_train_scaled_conv)
    encoded_X_test = hidden_representation.predict(X_test_scaled_conv)
#     print("encoded_X_train", encoded_X_train.shape)
#     print("encoded_X_test", encoded_X_test.shape)

#     encoded_X_train = encoder.predict(X_train_scaled_conv)
#     encoded_X_test = encoder.predict(X_test_scaled_conv)


    classifier = SVC()
#     print("Train data size", encoded_X_train.shape, y_train.shape)
#     print("Train test size", encoded_X_test.shape, y_test.shape)
#     print(len(encoded_X_train))
    encoded_X_train = np.reshape(encoded_X_train, (len(encoded_X_train), i))
    encoded_X_test = np.reshape(encoded_X_test, (len(encoded_X_test), i))
    print("encoded_X_train", encoded_X_train.shape)
    print("encoded_X_test", encoded_X_test.shape)
    classifier.fit(encoded_X_train, y_train)

        # Storing the predictions of the linear model
    y_pred_classifier = classifier.predict(encoded_X_test)


        # Plotting the encoded points
        #tsne_plot(encoded_X, encoded_y)
```

```
        # Performance
    print('amount of features: ' + str(i))
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()
    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')

    i = i + 1
```

```
Model: "model_51"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_41 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 2, 2, 1)           65
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 1, 1, 1)           0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 1, 1)           5
_________________________________________________________________
up_sampling2d_1 (UpSampling2 (None, 2, 2, 1)           0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 2, 2, 16)          80
_________________________________________________________________
up_sampling2d_2 (UpSampling2 (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 4, 4, 1)           65
=================================================================
Total params: 295
Trainable params: 295
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 1)
encoded_X_test (2466, 1)
amount of features: 1
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_53"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_42 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 2, 2, 2)           130
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 1, 1, 2)           0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 1, 1, 2)           18
_________________________________________________________________
up_sampling2d_3 (UpSampling2 (None, 2, 2, 2)           0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 2, 2, 16)          144
_________________________________________________________________
up_sampling2d_4 (UpSampling2 (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 437
Trainable params: 437
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 2)
encoded_X_test (2466, 2)
amount of features: 2
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_55"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

```
=================================================================
input_43 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 2, 2, 3)           195
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 1, 1, 3)           0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 1, 1, 3)           39
_________________________________________________________________
up_sampling2d_5 (UpSampling2 (None, 2, 2, 3)           0
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 2, 2, 16)          208
_________________________________________________________________
up_sampling2d_6 (UpSampling2 (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 587
Trainable params: 587
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 3)
encoded_X_test (2466, 3)
amount of features: 3
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_57"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_44 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 2, 2, 4)           260
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 1, 1, 4)           0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 1, 1, 4)           68
_________________________________________________________________
up_sampling2d_7 (UpSampling2 (None, 2, 2, 4)           0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 2, 2, 16)          272
_________________________________________________________________
up_sampling2d_8 (UpSampling2 (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 745
Trainable params: 745
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 4)
encoded_X_test (2466, 4)
amount of features: 4
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_59"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_45 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 4, 4, 16)          80
```

```
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_22 (Conv2D)           (None, 2, 2, 5)           325
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 1, 1, 5)           0
_________________________________________________________________
conv2d_23 (Conv2D)           (None, 1, 1, 5)           105
_________________________________________________________________
up_sampling2d_9 (UpSampling2 (None, 2, 2, 5)           0
_________________________________________________________________
conv2d_24 (Conv2D)           (None, 2, 2, 16)          336
_________________________________________________________________
up_sampling2d_10 (UpSampling (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_25 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 911
Trainable params: 911
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 5)
encoded_X_test (2466, 5)
amount of features: 5
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_61"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_46 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_26 (Conv2D)           (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_27 (Conv2D)           (None, 2, 2, 6)           390
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 1, 1, 6)           0
_________________________________________________________________
conv2d_28 (Conv2D)           (None, 1, 1, 6)           150
_________________________________________________________________
up_sampling2d_11 (UpSampling (None, 2, 2, 6)           0
_________________________________________________________________
conv2d_29 (Conv2D)           (None, 2, 2, 16)          400
_________________________________________________________________
up_sampling2d_12 (UpSampling (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_30 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 1,085
Trainable params: 1,085
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 6)
encoded_X_test (2466, 6)
amount of features: 6
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_63"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_47 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_31 (Conv2D)           (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_32 (Conv2D)           (None, 2, 2, 7)           455
```

```
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 1, 1, 7)           0
_________________________________________________________________
conv2d_33 (Conv2D)           (None, 1, 1, 7)           203
_________________________________________________________________
up_sampling2d_13 (UpSampling (None, 2, 2, 7)           0
_________________________________________________________________
conv2d_34 (Conv2D)           (None, 2, 2, 16)          464
_________________________________________________________________
up_sampling2d_14 (UpSampling (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 1,267
Trainable params: 1,267
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 7)
encoded_X_test (2466, 7)
amount of features: 7
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_65"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_48 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_36 (Conv2D)           (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_37 (Conv2D)           (None, 2, 2, 8)           520
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 1, 1, 8)           0
_________________________________________________________________
conv2d_38 (Conv2D)           (None, 1, 1, 8)           264
_________________________________________________________________
up_sampling2d_15 (UpSampling (None, 2, 2, 8)           0
_________________________________________________________________
conv2d_39 (Conv2D)           (None, 2, 2, 16)          528
_________________________________________________________________
up_sampling2d_16 (UpSampling (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_40 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 1,457
Trainable params: 1,457
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 8)
encoded_X_test (2466, 8)
amount of features: 8
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_67"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_49 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_41 (Conv2D)           (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_17 (MaxPooling (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_42 (Conv2D)           (None, 2, 2, 9)           585
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 1, 1, 9)           0
_________________________________________________________________
conv2d_43 (Conv2D)           (None, 1, 1, 9)           333
```

```
_________________________________________________________________
up_sampling2d_17 (UpSampling (None, 2, 2, 9)           0
_________________________________________________________________
conv2d_44 (Conv2D)           (None, 2, 2, 16)          592
_________________________________________________________________
up_sampling2d_18 (UpSampling (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_45 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 1,655
Trainable params: 1,655
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 9)
encoded_X_test (2466, 9)
amount of features: 9
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Model: "model_69"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_50 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_46 (Conv2D)           (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_47 (Conv2D)           (None, 2, 2, 10)          650
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 1, 1, 10)          0
_________________________________________________________________
conv2d_48 (Conv2D)           (None, 1, 1, 10)          410
_________________________________________________________________
up_sampling2d_19 (UpSampling (None, 2, 2, 10)          0
_________________________________________________________________
conv2d_49 (Conv2D)           (None, 2, 2, 16)          656
_________________________________________________________________
up_sampling2d_20 (UpSampling (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_50 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 1,861
Trainable params: 1,861
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 10)
encoded_X_test (2466, 10)
amount of features: 10
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0
```

Convolutional autoencoder with the best hyperparemeters of SVC

In [18]:

```
# added by Duc, hyper parameter tuning for SVC
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
# hyperparameter grid set for finetuning
# param_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel': ['rbf', 'poly', 'si
gmoid']}
param_grid = {'C': [0.1,1, 10, 100], 'kernel': ['rbf', 'sigmoid']}
#Encoder network Convolutional
i = 1
```

```
for i in range(1, 2):


    #print(X.shape[0])
    X_train_conv = X_train.drop(columns =['OperatingSystems'])
    X_test_conv = X_test.drop(columns = ['OperatingSystems'])
    scaler = MinMaxScaler()
    scaler.fit(X_train_conv)
    X_train_scaled_conv = scaler.transform(X_train_conv)
    X_test_scaled_conv = scaler.transform(X_test_conv)
    #print(X_train_scaled_conv.shape[1])
    #print(X_test_scaled_conv.shape[1])
    #print("Train data size", X_train_scaled_conv.shape, y_train.shape)
    #print("Train test size", X_test_scaled_conv.shape, y_test.shape)
    X_train_scaled_conv = np.reshape(X_train_scaled_conv, (len(X_train_scaled_conv), 4, 4, 1))
# adapt this if using `channels_first` image data format
    X_test_scaled_conv = np.reshape(X_test_scaled_conv, (len(X_test_scaled_conv), 4, 4, 1))  # a
dapt this if using `channels_first` image data format

    input_layer = Input(shape =(4,4,1))

    x = Conv2D(16, (2, 2), activation='relu', padding='same')(input_layer)
    x = MaxPooling2D((2, 2), padding='same')(x)
    x = Conv2D(i, (2, 2), activation='relu', padding='same')(x)
    encoded_conv = MaxPooling2D((2, 2), padding='same')(x)

    # at this point the representation is (4, 4, 8) i.e. 128-dimensional

    x = Conv2D(i, (2, 2), activation='relu', padding='same')(encoded_conv)
    x = UpSampling2D((2, 2))(x)
    x = Conv2D(16, (2, 2), activation='relu', padding='same')(x)
    x = UpSampling2D((2, 2))(x)
    decoded_conv = Conv2D(1, (2, 2), activation='sigmoid', padding='same')(x)


    # Defining the parameters of the Auto-encoder network
    encoder = Model(input_layer, encoded_conv)
    autoencoder = Model(input_layer, decoded_conv)
    autoencoder.compile(optimizer ="adadelta", loss ="mse")
    autoencoder.summary()
        # Training the Auto-encoder network
    autoencoder.fit(X_train_scaled_conv, X_train_scaled_conv,
                        batch_size = 3000, epochs = 50,
                        shuffle = True, validation_split = 0.2, verbose=0)


    hidden_representation = Sequential()
    hidden_representation.add(autoencoder.layers[0])
    hidden_representation.add(autoencoder.layers[1])
    hidden_representation.add(autoencoder.layers[2])
    hidden_representation.add(autoencoder.layers[3])
    hidden_representation.add(autoencoder.layers[4])

        # Separating the points encoded by the Auto-encoder as normal and fraud
        #nonbuyer_hidden_rep = hidden_representation.predict(X_nonbuyer_scaled)
        #buyer_hidden_rep = hidden_representation.predict(X_buyer_scaled)

        # Combining the encoded points into a single table
        #encoded_X = np.append(nonbuyer_hidden_rep, buyer_hidden_rep, axis = 0)
        #y_nonbuyer = np.zeros(nonbuyer_hidden_rep.shape[0])
        #y_buyer = np.ones(buyer_hidden_rep.shape[0])
        #encoded_y = np.append(y_nonbuyer, y_buyer)
    encoded_X_train = hidden_representation.predict(X_train_scaled_conv)
    encoded_X_test = hidden_representation.predict(X_test_scaled_conv)
```

```
#     encoded_X_train = encoder.predict(X_train_scaled_conv)
#     encoded_X_test = encoder.predict(X_test_scaled_conv)

    encoded_X_train = np.reshape(encoded_X_train, (len(encoded_X_train), i))
    encoded_X_test = np.reshape(encoded_X_test, (len(encoded_X_test), i))
    print("encoded_X_train", encoded_X_train.shape)
    print("encoded_X_test", encoded_X_test.shape)

################################## section below is modified by Duc
    # Performance
    classifier = SVC()
    classifier.fit(encoded_X_train, y_train)

    y_pred_classifier = classifier.predict(encoded_X_test)

    print('######################## amount of features: '+ str(i) + ' ##########################
#######')
    print('######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-
RESCALED ##################')
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    print(classification_report(y_test,y_pred_classifier))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#######################################
    # Performance
    # rescale encoded features
    scaler = MinMaxScaler()
    scaler.fit(encoded_X_train)
    encoded_X_train = scaler.transform(encoded_X_train)
    encoded_X_test = scaler.transform(encoded_X_test)

    print('######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESC
ALED ##################')
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    print(classification_report(y_test,y_pred_classifier))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#####################################
    # Performance
    classifier = SVC()
    classifier_grid = GridSearchCV(classifier,param_grid,refit=True,verbose=1)
    classifier_grid.fit(encoded_X_train, y_train)

    y_pred_classifier_grid = classifier_grid.predict(encoded_X_test)

    print('######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ########
##########')
    print("The best parameters are %s with a score of %0.2f" % (classifier_grid.best_params_, cl
assifier_grid.best_score_))
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier_grid)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier_grid)))
    print(classification_report(y_test,y_pred_classifier_grid))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier_grid).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#####################################
    # Performance
    wsvmclf = SVC(class_weight='balanced')
```

```
    wsvmclf_grid = GridSearchCV(wsvmclf,param_grid,refit=True,verbose=1)
    wsvmclf_grid.fit(encoded_X_train, y_train)

    y_pred_wsvmclf_grid = wsvmclf_grid.predict(encoded_X_test)

    print('##########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ########
##########')
    print("The best parameters are %s with a score of %0.2f" % (wsvmclf_grid.best_params_, wsvmc
lf_grid.best_score_))
    print('Accuracy with best hyperparameters: '+str(accuracy_score(y_test, y_pred_wsvmclf_grid
)))
    print('Confusion Matrix with best hyperparameters: \n' + str(confusion_matrix(y_test,y_pred_
wsvmclf_grid)))
    print(classification_report(y_test,y_pred_wsvmclf_grid))
    # incorrect order of output
#         tn, fn, fp, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()
    # the right order is as follows
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_wsvmclf_grid).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')

    i = i + 1
```

```
Model: "model_72"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_51 (InputLayer)        (None, 4, 4, 1)           0
_________________________________________________________________
conv2d_51 (Conv2D)           (None, 4, 4, 16)          80
_________________________________________________________________
max_pooling2d_21 (MaxPooling (None, 2, 2, 16)          0
_________________________________________________________________
conv2d_52 (Conv2D)           (None, 2, 2, 1)           65
_________________________________________________________________
max_pooling2d_22 (MaxPooling (None, 1, 1, 1)           0
_________________________________________________________________
conv2d_53 (Conv2D)           (None, 1, 1, 1)           5
_________________________________________________________________
up_sampling2d_21 (UpSampling (None, 2, 2, 1)           0
_________________________________________________________________
conv2d_54 (Conv2D)           (None, 2, 2, 16)          80
_________________________________________________________________
up_sampling2d_22 (UpSampling (None, 4, 4, 16)          0
_________________________________________________________________
conv2d_55 (Conv2D)           (None, 4, 4, 1)           65
=================================================================
Total params: 295
Trainable params: 295
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 1)
encoded_X_test (2466, 1)
####################### amount of features: 1 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   44.8s finished
```

```
######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.6min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'sigmoid'} with a score of 0.55
Accuracy with best hyperparameters: 0.5506893755068938
Confusion Matrix with best hyperparameters:
[[1138  927]
 [ 181  220]]
              precision    recall  f1-score   support

           0       0.86      0.55      0.67      2065
           1       0.19      0.55      0.28       401

    accuracy                           0.55      2466
   macro avg       0.53      0.55      0.48      2466
weighted avg       0.75      0.55      0.61      2466

True negatives: 1138
False positives: 927
False negatives: 181
True positives: 220
```

1D Convolutional Autoencoder with SVC tuning

In [19]:

```
# added by Duc, hyper parameter tuning for SVC
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
# hyperparameter grid set for finetuning
# param_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001],'kernel': ['rbf', 'poly', 'si
gmoid']}
param_grid = {'C': [0.1,1, 10, 100], 'kernel': ['rbf', 'sigmoid']}
#Encoder network Convolutional
from keras.layers import Input, Dense, Conv1D, MaxPooling1D, UpSampling1D

X_train_conv = X_train.drop(columns =['OperatingSystems'])
X_test_conv = X_test.drop(columns = ['OperatingSystems'])
scaler = MinMaxScaler()
scaler.fit(X_train_conv)
X_train_scaled_conv = scaler.transform(X_train_conv)
X_test_scaled_conv = scaler.transform(X_test_conv)
X_train_scaled_conv = np.reshape(X_train_scaled_conv, (len(X_train_scaled_conv), X_train_scaled_
conv.shape[1], 1))  # adapt this if using `channels_first` image data format
X_test_scaled_conv = np.reshape(X_test_scaled_conv, (len(X_test_scaled_conv), X_train_scaled_con
v.shape[1], 1))  # adapt this if using `channels_first` image data format
```

```
for i in range(1, 11):
    input_layer = Input(shape=(X_train_scaled_conv.shape[1], 1))

    x = Conv1D(100, 3, activation='relu', padding='same')(input_layer)
    x = MaxPooling1D(2, padding='same')(x)
    x = Conv1D(50, 3, activation='relu', padding='same')(x)
    x = MaxPooling1D(2, padding='same')(x)
    x = Conv1D(25, 3, activation='relu', padding='same')(x)
    x = MaxPooling1D(2, padding='same')(x)
    x = Conv1D(i, 3, activation='relu', padding='same')(x)
    encoded_conv = MaxPooling1D(2, padding='same')(x)

    # at this point the representation is (4, 4, 8) i.e. 128-dimensional

    x = Conv1D(i, 3, activation='relu', padding='same')(encoded_conv)
    x = UpSampling1D(2)(x)
    x = Conv1D(25, 3, activation='relu', padding='same')(x)
    x = UpSampling1D(2)(x)
    x = Conv1D(50, 3, activation='relu', padding='same')(x)
    x = UpSampling1D(2)(x)
    x = Conv1D(100, 3, activation='relu', padding='same')(x)
    x = UpSampling1D(2)(x)
    decoded_conv = Conv1D(1, 3, activation='sigmoid', padding='same')(x)


    # Defining the parameters of the Auto-encoder network
    encoder = Model(input_layer, encoded_conv)
    autoencoder = Model(input_layer, decoded_conv)
    autoencoder.compile(optimizer ="adadelta", loss ="mse")
    autoencoder.summary()
        # Training the Auto-encoder network
    autoencoder.fit(X_train_scaled_conv, X_train_scaled_conv,
                        batch_size = 3000, epochs = 50,
                        shuffle = True, validation_split = 0.2, verbose=0)


    hidden_representation = Sequential()
    hidden_representation.add(autoencoder.layers[0])
    hidden_representation.add(autoencoder.layers[1])
    hidden_representation.add(autoencoder.layers[2])
    hidden_representation.add(autoencoder.layers[3])
    hidden_representation.add(autoencoder.layers[4])
    hidden_representation.add(autoencoder.layers[5])
    hidden_representation.add(autoencoder.layers[6])
    hidden_representation.add(autoencoder.layers[7])
    hidden_representation.add(autoencoder.layers[8])

        # Separating the points encoded by the Auto-encoder as normal and fraud
        #nonbuyer_hidden_rep = hidden_representation.predict(X_nonbuyer_scaled)
        #buyer_hidden_rep = hidden_representation.predict(X_buyer_scaled)

        # Combining the encoded points into a single table
        #encoded_X = np.append(nonbuyer_hidden_rep, buyer_hidden_rep, axis = 0)
        #y_nonbuyer = np.zeros(nonbuyer_hidden_rep.shape[0])
        #y_buyer = np.ones(buyer_hidden_rep.shape[0])
        #encoded_y = np.append(y_nonbuyer, y_buyer)
    encoded_X_train = hidden_representation.predict(X_train_scaled_conv)
    encoded_X_test = hidden_representation.predict(X_test_scaled_conv)

#     encoded_X_train = encoder.predict(X_train_scaled_conv)
#     encoded_X_test = encoder.predict(X_test_scaled_conv)

    encoded_X_train = np.reshape(encoded_X_train, (len(encoded_X_train), i))
    encoded_X_test = np.reshape(encoded_X_test, (len(encoded_X_test), i))
```

```
    print("encoded_X_train", encoded_X_train.shape)
    print("encoded_X_test", encoded_X_test.shape)

################################## section below is modified by Duc
    # Performance
    classifier = SVC()
    classifier.fit(encoded_X_train, y_train)

    y_pred_classifier = classifier.predict(encoded_X_test)

    print('######################### amount of features: '+ str(i) + ' ##########################
#######')
    print('######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-
RESCALED ##################')
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    print(classification_report(y_test,y_pred_classifier))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#####################################
    # Performance
    # rescale encoded features
    scaler = MinMaxScaler()
    scaler.fit(encoded_X_train)
    encoded_X_train = scaler.transform(encoded_X_train)
    encoded_X_test = scaler.transform(encoded_X_test)

    print('######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESC
ALED ##################')
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier)))
    print(classification_report(y_test,y_pred_classifier))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#####################################
    # Performance
    classifier = SVC()
    classifier_grid = GridSearchCV(classifier,param_grid,refit=True,verbose=1)
    classifier_grid.fit(encoded_X_train, y_train)

    y_pred_classifier_grid = classifier_grid.predict(encoded_X_test)

    print('######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ########
##########')
    print("The best parameters are %s with a score of %0.2f" % (classifier_grid.best_params_, cl
assifier_grid.best_score_))
    print('Accuracy : '+str(accuracy_score(y_test, y_pred_classifier_grid)))
    print('Confusion Matrix: \n' + str(confusion_matrix(y_test,y_pred_classifier_grid)))
    print(classification_report(y_test,y_pred_classifier_grid))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_classifier_grid).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')
#####################################
    # Performance
    wsvmclf = SVC(class_weight='balanced')
    wsvmclf_grid = GridSearchCV(wsvmclf,param_grid,refit=True,verbose=1)
    wsvmclf_grid.fit(encoded_X_train, y_train)

    y_pred_wsvmclf_grid = wsvmclf_grid.predict(encoded_X_test)
```

```
    print('#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ########
##########')
    print("The best parameters are %s with a score of %0.2f" % (wsvmclf_grid.best_params_, wsvmc
lf_grid.best_score_))
    print('Accuracy with best hyperparameters: '+str(accuracy_score(y_test, y_pred_wsvmclf_grid
)))
    print('Confusion Matrix with best hyperparameters: \n' + str(confusion_matrix(y_test,y_pred_
wsvmclf_grid)))
    print(classification_report(y_test,y_pred_wsvmclf_grid))
    # incorrect order of output
#         tn, fn, fp, tp = confusion_matrix(y_test, y_pred_svmclf).ravel()
    # the right order is as follows
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred_wsvmclf_grid).ravel()

    print('True negatives: ' + str(tn) + '\n' + 'False positives: ' + str(fp) +  '\n' + 'False n
egatives: ' + str(fn) + '\n'+ 'True positives: ' + str(tp) + '\n')

    i = i + 1
```

```
Model: "model_74"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_52 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 8, 100)            0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 4, 50)             0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 2, 25)             0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 2, 1)              76
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 1, 1)              0
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 1, 1)              4
_________________________________________________________________
up_sampling1d_1 (UpSampling1 (None, 2, 1)              0
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 2, 25)             100
_________________________________________________________________
up_sampling1d_2 (UpSampling1 (None, 4, 25)             0
_________________________________________________________________
conv1d_7 (Conv1D)            (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_3 (UpSampling1 (None, 8, 50)             0
_________________________________________________________________
conv1d_8 (Conv1D)            (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_4 (UpSampling1 (None, 16, 100)           0
_________________________________________________________________
conv1d_9 (Conv1D)            (None, 16, 1)             301
=================================================================
Total params: 38,606
Trainable params: 38,606
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 1)
encoded_X_test (2466, 1)
####################### amount of features: 1 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
```

```
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   51.0s finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.57
Accuracy with best hyperparameters: 0.5685320356853204
Confusion Matrix with best hyperparameters:
[[1234  831]
 [ 233  168]]
              precision    recall  f1-score   support

           0       0.84      0.60      0.70      2065
           1       0.17      0.42      0.24       401

    accuracy                           0.57      2466
   macro avg       0.50      0.51      0.47      2466
weighted avg       0.73      0.57      0.62      2466

True negatives: 1234
False positives: 831
False negatives: 233
True positives: 168

Model: "model_76"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_53 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 8, 100)            0
_________________________________________________________________
conv1d_11 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_6 (MaxPooling1 (None, 4, 50)             0
_________________________________________________________________
conv1d_12 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_7 (MaxPooling1 (None, 2, 25)             0
_________________________________________________________________
conv1d_13 (Conv1D)           (None, 2, 2)              152
_________________________________________________________________
max_pooling1d_8 (MaxPooling1 (None, 1, 2)              0
_________________________________________________________________
conv1d_14 (Conv1D)           (None, 1, 2)              14
_________________________________________________________________
up_sampling1d_5 (UpSampling1 (None, 2, 2)              0
_________________________________________________________________
conv1d_15 (Conv1D)           (None, 2, 25)             175
_________________________________________________________________
up_sampling1d_6 (UpSampling1 (None, 4, 25)             0
_________________________________________________________________
conv1d_16 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_7 (UpSampling1 (None, 8, 50)             0
_________________________________________________________________
conv1d_17 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_8 (UpSampling1 (None, 16, 100)           0
_________________________________________________________________
conv1d_18 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 38,767
Trainable params: 38,767
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 2)
encoded_X_test (2466, 2)
######################## amount of features: 2 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.9min finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'sigmoid'} with a score of 0.58
Accuracy with best hyperparameters: 0.5904298459042985
Confusion Matrix with best hyperparameters:
[[1219  846]
 [ 164  237]]
              precision    recall  f1-score   support

           0       0.88      0.59      0.71      2065
           1       0.22      0.59      0.32       401

    accuracy                           0.59      2466
   macro avg       0.55      0.59      0.51      2466
weighted avg       0.77      0.59      0.64      2466

True negatives: 1219
False positives: 846
False negatives: 164
True positives: 237

Model: "model_78"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_54 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_9 (MaxPooling1 (None, 8, 100)            0
_________________________________________________________________
conv1d_20 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_10 (MaxPooling (None, 4, 50)             0
_________________________________________________________________
conv1d_21 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_11 (MaxPooling (None, 2, 25)             0
_________________________________________________________________
conv1d_22 (Conv1D)           (None, 2, 3)              228
_________________________________________________________________
max_pooling1d_12 (MaxPooling (None, 1, 3)              0
_________________________________________________________________
conv1d_23 (Conv1D)           (None, 1, 3)              30
_________________________________________________________________
up_sampling1d_9 (UpSampling1 (None, 2, 3)              0
_________________________________________________________________
conv1d_24 (Conv1D)           (None, 2, 25)             250
_________________________________________________________________
up_sampling1d_10 (UpSampling (None, 4, 25)             0
_________________________________________________________________
conv1d_25 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_11 (UpSampling (None, 8, 50)             0
_________________________________________________________________
conv1d_26 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_12 (UpSampling (None, 16, 100)           0
_________________________________________________________________
conv1d_27 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 38,934
Trainable params: 38,934
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 3)
encoded_X_test (2466, 3)
######################## amount of features: 3 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.0min finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'sigmoid'} with a score of 0.52
Accuracy with best hyperparameters: 0.5251419302514193
Confusion Matrix with best hyperparameters:
[[1085  980]
 [ 191  210]]
              precision    recall  f1-score   support

           0       0.85      0.53      0.65      2065
           1       0.18      0.52      0.26       401

    accuracy                           0.53      2466
   macro avg       0.51      0.52      0.46      2466
weighted avg       0.74      0.53      0.59      2466

True negatives: 1085
False positives: 980
False negatives: 191
True positives: 210

Model: "model_80"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_55 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_28 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_13 (MaxPooling (None, 8, 100)            0
_________________________________________________________________
conv1d_29 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_14 (MaxPooling (None, 4, 50)             0
_________________________________________________________________
conv1d_30 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_15 (MaxPooling (None, 2, 25)             0
_________________________________________________________________
conv1d_31 (Conv1D)           (None, 2, 4)              304
_________________________________________________________________
max_pooling1d_16 (MaxPooling (None, 1, 4)              0
_________________________________________________________________
conv1d_32 (Conv1D)           (None, 1, 4)              52
_________________________________________________________________
up_sampling1d_13 (UpSampling (None, 2, 4)              0
_________________________________________________________________
conv1d_33 (Conv1D)           (None, 2, 25)             325
_________________________________________________________________
up_sampling1d_14 (UpSampling (None, 4, 25)             0
_________________________________________________________________
conv1d_34 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_15 (UpSampling (None, 8, 50)             0
_________________________________________________________________
conv1d_35 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_16 (UpSampling (None, 16, 100)           0
_________________________________________________________________
conv1d_36 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 39,107
Trainable params: 39,107
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 4)
encoded_X_test (2466, 4)
######################### amount of features: 4 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  2.1min finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8386050283860503
Confusion Matrix:
[[2064    1]
 [ 397    4]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.80      0.01      0.02       401

    accuracy                           0.84      2466
   macro avg       0.82      0.50      0.47      2466
weighted avg       0.83      0.84      0.77      2466

True negatives: 2064
False positives: 1
False negatives: 397
True positives: 4

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.66
Accuracy with best hyperparameters: 0.6995133819951338
Confusion Matrix with best hyperparameters:
[[1514  551]
 [ 190  211]]
              precision    recall  f1-score   support

           0       0.89      0.73      0.80      2065
           1       0.28      0.53      0.36       401

    accuracy                           0.70      2466
   macro avg       0.58      0.63      0.58      2466
weighted avg       0.79      0.70      0.73      2466

True negatives: 1514
False positives: 551
False negatives: 190
True positives: 211

Model: "model_82"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_56 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_37 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_17 (MaxPooling (None, 8, 100)            0
_________________________________________________________________
conv1d_38 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_18 (MaxPooling (None, 4, 50)             0
_________________________________________________________________
conv1d_39 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 2, 25)             0
_________________________________________________________________
conv1d_40 (Conv1D)           (None, 2, 5)              380
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 1, 5)              0
_________________________________________________________________
conv1d_41 (Conv1D)           (None, 1, 5)              80
_________________________________________________________________
up_sampling1d_17 (UpSampling (None, 2, 5)              0
_________________________________________________________________
conv1d_42 (Conv1D)           (None, 2, 25)             400
_________________________________________________________________
up_sampling1d_18 (UpSampling (None, 4, 25)             0
_________________________________________________________________
conv1d_43 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_19 (UpSampling (None, 8, 50)             0
_________________________________________________________________
conv1d_44 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_20 (UpSampling (None, 16, 100)           0
_________________________________________________________________
conv1d_45 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 39,286
Trainable params: 39,286
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 5)
encoded_X_test (2466, 5)
######################### amount of features: 5 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   51.4s finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.57
Accuracy with best hyperparameters: 0.5790754257907542
Confusion Matrix with best hyperparameters:
[[1174  891]
 [ 147  254]]
              precision    recall  f1-score   support

           0       0.89      0.57      0.69      2065
           1       0.22      0.63      0.33       401

    accuracy                           0.58      2466
   macro avg       0.56      0.60      0.51      2466
weighted avg       0.78      0.58      0.63      2466

True negatives: 1174
False positives: 891
False negatives: 147
True positives: 254

Model: "model_84"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_57 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_46 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_21 (MaxPooling (None, 8, 100)            0
_________________________________________________________________
conv1d_47 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_22 (MaxPooling (None, 4, 50)             0
_________________________________________________________________
conv1d_48 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_23 (MaxPooling (None, 2, 25)             0
_________________________________________________________________
conv1d_49 (Conv1D)           (None, 2, 6)              456
_________________________________________________________________
max_pooling1d_24 (MaxPooling (None, 1, 6)              0
_________________________________________________________________
conv1d_50 (Conv1D)           (None, 1, 6)              114
_________________________________________________________________
up_sampling1d_21 (UpSampling (None, 2, 6)              0
_________________________________________________________________
conv1d_51 (Conv1D)           (None, 2, 25)             475
_________________________________________________________________
up_sampling1d_22 (UpSampling (None, 4, 25)             0
_________________________________________________________________
conv1d_52 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_23 (UpSampling (None, 8, 50)             0
_________________________________________________________________
conv1d_53 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_24 (UpSampling (None, 16, 100)           0
_________________________________________________________________
conv1d_54 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 39,471
Trainable params: 39,471
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 6)
encoded_X_test (2466, 6)
######################## amount of features: 6 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  2.2min finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.61
Accuracy with best hyperparameters: 0.6269261962692619
Confusion Matrix with best hyperparameters:
[[1280  785]
 [ 135  266]]
              precision    recall  f1-score   support

           0       0.90      0.62      0.74      2065
           1       0.25      0.66      0.37       401

    accuracy                           0.63      2466
   macro avg       0.58      0.64      0.55      2466
weighted avg       0.80      0.63      0.68      2466

True negatives: 1280
False positives: 785
False negatives: 135
True positives: 266

Model: "model_86"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_58 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_55 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_25 (MaxPooling (None, 8, 100)            0
_________________________________________________________________
conv1d_56 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_26 (MaxPooling (None, 4, 50)             0
_________________________________________________________________
conv1d_57 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_27 (MaxPooling (None, 2, 25)             0
_________________________________________________________________
conv1d_58 (Conv1D)           (None, 2, 7)              532
_________________________________________________________________
max_pooling1d_28 (MaxPooling (None, 1, 7)              0
_________________________________________________________________
conv1d_59 (Conv1D)           (None, 1, 7)              154
_________________________________________________________________
up_sampling1d_25 (UpSampling (None, 2, 7)              0
_________________________________________________________________
conv1d_60 (Conv1D)           (None, 2, 25)             550
_________________________________________________________________
up_sampling1d_26 (UpSampling (None, 4, 25)             0
_________________________________________________________________
conv1d_61 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_27 (UpSampling (None, 8, 50)             0
_________________________________________________________________
conv1d_62 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_28 (UpSampling (None, 16, 100)           0
_________________________________________________________________
conv1d_63 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 39,662
Trainable params: 39,662
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 7)
encoded_X_test (2466, 7)
######################### amount of features: 7 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  2.6min finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 0.1, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.61
Accuracy with best hyperparameters: 0.615977291159773
Confusion Matrix with best hyperparameters:
[[1266  799]
 [ 148  253]]
              precision    recall  f1-score   support

           0       0.90      0.61      0.73      2065
           1       0.24      0.63      0.35       401

    accuracy                           0.62      2466
   macro avg       0.57      0.62      0.54      2466
weighted avg       0.79      0.62      0.67      2466

True negatives: 1266
False positives: 799
False negatives: 148
True positives: 253

Model: "model_88"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_59 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_64 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_29 (MaxPooling (None, 8, 100)            0
_________________________________________________________________
conv1d_65 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_30 (MaxPooling (None, 4, 50)             0
_________________________________________________________________
conv1d_66 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_31 (MaxPooling (None, 2, 25)             0
_________________________________________________________________
conv1d_67 (Conv1D)           (None, 2, 8)              608
_________________________________________________________________
max_pooling1d_32 (MaxPooling (None, 1, 8)              0
_________________________________________________________________
conv1d_68 (Conv1D)           (None, 1, 8)              200
_________________________________________________________________
up_sampling1d_29 (UpSampling (None, 2, 8)              0
_________________________________________________________________
conv1d_69 (Conv1D)           (None, 2, 25)             625
_________________________________________________________________
up_sampling1d_30 (UpSampling (None, 4, 25)             0
_________________________________________________________________
conv1d_70 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_31 (UpSampling (None, 8, 50)             0
_________________________________________________________________
conv1d_71 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_32 (UpSampling (None, 16, 100)           0
_________________________________________________________________
conv1d_72 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 39,859
Trainable params: 39,859
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 8)
encoded_X_test (2466, 8)
######################## amount of features: 8 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   49.2s finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8410381184103812
Confusion Matrix:
[[2065    0]
 [ 392    9]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       1.00      0.02      0.04       401

    accuracy                           0.84      2466
   macro avg       0.92      0.51      0.48      2466
weighted avg       0.87      0.84      0.77      2466

True negatives: 2065
False positives: 0
False negatives: 392
True positives: 9

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.66
Accuracy with best hyperparameters: 0.6796431467964315
Confusion Matrix with best hyperparameters:
[[1378  687]
 [ 103  298]]
              precision    recall  f1-score   support

           0       0.93      0.67      0.78      2065
           1       0.30      0.74      0.43       401

    accuracy                           0.68      2466
   macro avg       0.62      0.71      0.60      2466
weighted avg       0.83      0.68      0.72      2466

True negatives: 1378
False positives: 687
False negatives: 103
True positives: 298

Model: "model_90"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_60 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_73 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_33 (MaxPooling (None, 8, 100)            0
_________________________________________________________________
conv1d_74 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_34 (MaxPooling (None, 4, 50)             0
_________________________________________________________________
conv1d_75 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_35 (MaxPooling (None, 2, 25)             0
_________________________________________________________________
conv1d_76 (Conv1D)           (None, 2, 9)              684
_________________________________________________________________
max_pooling1d_36 (MaxPooling (None, 1, 9)              0
_________________________________________________________________
conv1d_77 (Conv1D)           (None, 1, 9)              252
_________________________________________________________________
up_sampling1d_33 (UpSampling (None, 2, 9)              0
_________________________________________________________________
conv1d_78 (Conv1D)           (None, 2, 25)             700
_________________________________________________________________
up_sampling1d_34 (UpSampling (None, 4, 25)             0
_________________________________________________________________
conv1d_79 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_35 (UpSampling (None, 8, 50)             0
_________________________________________________________________
conv1d_80 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_36 (UpSampling (None, 16, 100)           0
_________________________________________________________________
conv1d_81 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 40,062
Trainable params: 40,062
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 9)
encoded_X_test (2466, 9)
######################### amount of features: 9 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:   43.1s finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8527980535279805
Confusion Matrix:
[[2059    6]
 [ 357   44]]
              precision    recall  f1-score   support

           0       0.85      1.00      0.92      2065
           1       0.88      0.11      0.20       401

    accuracy                           0.85      2466
   macro avg       0.87      0.55      0.56      2466
weighted avg       0.86      0.85      0.80      2466

True negatives: 2059
False positives: 6
False negatives: 357
True positives: 44

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.3min finished
```

```
#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.74
Accuracy with best hyperparameters: 0.7433090024330901
Confusion Matrix with best hyperparameters:
[[1564  501]
 [ 132  269]]
              precision    recall  f1-score   support

           0       0.92      0.76      0.83      2065
           1       0.35      0.67      0.46       401

    accuracy                           0.74      2466
   macro avg       0.64      0.71      0.65      2466
weighted avg       0.83      0.74      0.77      2466

True negatives: 1564
False positives: 501
False negatives: 132
True positives: 269

Model: "model_92"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_61 (InputLayer)        (None, 16, 1)             0
_________________________________________________________________
conv1d_82 (Conv1D)           (None, 16, 100)           400
_________________________________________________________________
max_pooling1d_37 (MaxPooling (None, 8, 100)            0
_________________________________________________________________
conv1d_83 (Conv1D)           (None, 8, 50)             15050
_________________________________________________________________
max_pooling1d_38 (MaxPooling (None, 4, 50)             0
_________________________________________________________________
conv1d_84 (Conv1D)           (None, 4, 25)             3775
_________________________________________________________________
max_pooling1d_39 (MaxPooling (None, 2, 25)             0
_________________________________________________________________
conv1d_85 (Conv1D)           (None, 2, 10)             760
_________________________________________________________________
max_pooling1d_40 (MaxPooling (None, 1, 10)             0
_________________________________________________________________
conv1d_86 (Conv1D)           (None, 1, 10)             310
_________________________________________________________________
up_sampling1d_37 (UpSampling (None, 2, 10)             0
_________________________________________________________________
conv1d_87 (Conv1D)           (None, 2, 25)             775
_________________________________________________________________
up_sampling1d_38 (UpSampling (None, 4, 25)             0
_________________________________________________________________
conv1d_88 (Conv1D)           (None, 4, 50)             3800
_________________________________________________________________
up_sampling1d_39 (UpSampling (None, 8, 50)             0
_________________________________________________________________
conv1d_89 (Conv1D)           (None, 8, 100)            15100
_________________________________________________________________
up_sampling1d_40 (UpSampling (None, 16, 100)           0
_________________________________________________________________
conv1d_90 (Conv1D)           (None, 16, 1)             301
=================================================================
Total params: 40,271
Trainable params: 40,271
Non-trainable params: 0
_________________________________________________________________
encoded_X_train (9864, 10)
encoded_X_test (2466, 10)
######################### amount of features: 10 ################################
######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - NOT-RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
```

```
False negatives: 401
True positives: 0

######### PERFORMANCE WITH DEFAULT HYPERPARAMETERS FOR SVC (C=1, kernel = rbf) - RESCALED ##################
Accuracy : 0.8373884833738848
Confusion Matrix:
[[2065    0]
 [ 401    0]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.00      0.00      0.00       401

    accuracy                           0.84      2466
   macro avg       0.42      0.50      0.46      2466
weighted avg       0.70      0.84      0.76      2466

True negatives: 2065
False positives: 0
False negatives: 401
True positives: 0

Fitting 5 folds for each of 8 candidates, totalling 40 fits


/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ubuntu/anaconda3/envs/tf-gpu/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarnin
g: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.5min finished


######### PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR SVC - RESCALED ##################
The best parameters are {'C': 100, 'kernel': 'rbf'} with a score of 0.85
Accuracy : 0.8369829683698297
Confusion Matrix:
[[2062    3]
 [ 399    2]]
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2065
           1       0.40      0.00      0.01       401

    accuracy                           0.84      2466
   macro avg       0.62      0.50      0.46      2466
weighted avg       0.77      0.84      0.76      2466

True negatives: 2062
False positives: 3
False negatives: 399
True positives: 2

Fitting 5 folds for each of 8 candidates, totalling 40 fits


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  40 out of  40 | elapsed:  1.4min finished


#########PERFORMANCE WITH THE BEST HYPERPARAMETERS TUNED FOR WSVC - RESCALED ##################
The best parameters are {'C': 10, 'kernel': 'rbf'} with a score of 0.62
Accuracy with best hyperparameters: 0.6447688564476886
Confusion Matrix with best hyperparameters:
[[1306  759]
 [ 117  284]]
              precision    recall  f1-score   support

           0       0.92      0.63      0.75      2065
           1       0.27      0.71      0.39       401

    accuracy                           0.64      2466
   macro avg       0.60      0.67      0.57      2466
weighted avg       0.81      0.64      0.69      2466

True negatives: 1306
False positives: 759
False negatives: 117
True positives: 284
```

In [ ]: