```
print(classification_report(y_test_bal,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.87      0.16      0.28       122
           1       0.82      0.94      0.87       122
           2       0.93      0.98      0.96       122
           3       0.77      0.84      0.80       122
           4       0.49      0.89      0.63       122
           5       0.88      0.68      0.77       122
           6       0.59      0.53      0.56       122

    accuracy                           0.72       854
   macro avg       0.76      0.72      0.70       854
weighted avg       0.76      0.72      0.70       854
```

```
train_score = accuracy_score(y_bal,rfr1.predict(x_bal))
```

```
train_score
```

```
1.0
```

```
y_pred=xgb.predict(x_test_bal)
```

```
print(classification_report(y_test_bal,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.80      0.30      0.44       122
           1       0.82      0.94      0.88       122
           2       0.96      1.00      0.98       122
           3       0.77      0.84      0.81       122
           4       0.51      0.81      0.62       122
           5       0.84      0.70      0.76       122
           6       0.59      0.54      0.56       122

    accuracy                           0.73       854
```

```
y_pred = sv.predict(x_test_bal)
```

```
print(classification_report(y_test_bal,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.70      0.85      0.77       122
           1       0.76      0.81      0.79       122
           2       0.88      0.93      0.90       122
           3       0.71      0.65      0.68       122
           4       0.71      0.63      0.67       122
           5       0.76      0.54      0.63       122
           6       0.49      0.57      0.52       122

    accuracy                           0.71       854
   macro avg       0.72      0.71      0.71       854
weighted avg       0.72      0.71      0.71       854
```

```
train_score=accuracy_score(y_bal,sv.predict(x_bal))
train_score
```

```
0.7154989384288747
```

```
y_pred = model.predict(x_test_bal)
```

```
27/27 [==============================] - 0s 3ms/step
```

```
print(classification_report(y_test_bal,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       122
           1       0.14      1.00      0.25       122
           2       0.00      0.00      0.00       122
           3       0.00      0.00      0.00       122
           4       0.00      0.00      0.00       122
           5       0.00      0.00      0.00       122
           6       0.00      0.00      0.00       122

    accuracy                           0.14       854
   macro avg       0.02      0.14      0.04       854
weighted avg       0.02      0.14      0.04       854
```

```
C:\Users\Mahidhar reddy\anaconda3\lib\site-packages\sklearn\metrics\_
and F-score are ill-defined and being set to 0.0 in labels with no pr
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Mahidhar reddy\anaconda3\lib\site-packages\sklearn\metrics\_
and F-score are ill-defined and being set to 0.0 in labels with no pr
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Mahidhar reddy\anaconda3\lib\site-packages\sklearn\metrics\_
and F-score are ill-defined and being set to 0.0 in labels with no pr
```

```
params = {


        'C': [0.1, 1, 10, 100, 1000],
        'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
            'kernel': ['rbf','sqrt']


}
```

```
random_svc = RandomizedSearchCV(sv,params, scoring='accuracy',cv=5,n_jobs=-1)
```

```
random_svc.fit(x_bal,y_bal)
```

```
random_svc.best_params_
```

```
{'kernel': 'rbf', 'gamma': 1, 'C': 1}
```

```
sv1=SVC(kernel= 'rbf', gamma= 0.1, C= 100)
```

```
sv1.fit(x_bal,y_bal)
```

```
C:\Users\SmartBridge-PC\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111:
was passed when a 1d array was expected. Please change the shape of y to (n_samples,
  y = column_or_1d(y, warn=True)
```

```
SVC
SVC(C=100, gamma=0.1)
```

```
y_pred= sv1.predict(x_test_bal)
```

```
print(classification_report(y_test_bal,y_pred))
```

```
              precision    recall  f1-score   support

           -       - --      - --      - --       ---
```

```
print(classification_report(y_test_bal,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.74      0.75      0.75       122
           1       0.77      0.86      0.81       122
           2       0.95      0.91      0.93       122
           3       0.70      0.66      0.68       122
           4       0.66      0.73      0.70       122
           5       0.72      0.72      0.72       122
           6       0.57      0.48      0.52       122

    accuracy                           0.73       854
   macro avg       0.73      0.73      0.73       854
weighted avg       0.73      0.73      0.73       854
```

```
train_score= accuracy_score(y_bal,sv1.predict(x_bal))
train_score
```

```
0.8125568698817106
```

ving the model as
yroid1_model.pkl

```
# saving the model
import pickle
pickle.dump(sv1,open('thyroid_1_model.pkl','wb'))
```

```
features = np.array([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]])
print(label_encoder.inverse_transform(xgb1.predict(features)))
```

```
['hypothyroid conditions']
```

ere, we are saving
bel_encoding also as
bel_encoder.pkl

```
pickle.dump(label_encoder,open('label_encoder.pkl','wb'))
```

```
data['target'].unique()
```

```
pickle.dump(label_encoder,open('label_encoder.pkl','wb'))
```

```
data['target'].unique()
```

```
array(['miscellaneous', 'hypothyroid conditions', 'binding protein',
       'replacement therapy', 'general health', 'hyperthyroid conditions',
       'antithyroid treatment'], dtype=object)
```

```
y['target'].unique()
```

```
array([5, 4, 1, 6, 2, 3, 0])
```