In [1]:

```
#Importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
#Read the dataset
data=pd.read_csv("thyroid1.csv")
```

In [3]:

```
data.head()
```

Out[3]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	TT4 measured	TT4	T4U measured	T4U	FTI measured	FTI	TBG measured	TBG	referral source	target
0	32	F	f	f	f	f	f	f	f	f	...	f	32	f	1.00	f	150	t	36	other	S
1	75	F	t	f	t	t	f	t	f	f	...	t	157	t	0.89	t	176	f	78	SVI	A
2	63	F	t	t	f	t	t	f	f	t	...	t	48	t	1.02	t	47	f	43	other	F
3	41	M	f	t	f	f	f	f	f	f	...	t	39	t	1.00	t	39	f	22	other	R
4	41	F	f	t	f	f	f	f	f	f	...	t	33	t	1.07	t	31	f	34	other	F

5 rows × 30 columns

In [4]:

```
data.shape
```

Out[4]:

```
(50, 30)
```

In [5]:

```
#Checking the null values
data.isnull().any()
```

Out[5]:

```
age                     False
sex                     False
on thyroxine            False
query on thyroxine      False
on antithyroid meds     False
sick                    False
pregnant                False
thyroid surgery         False
I131 treatment          False
query hypothyroid       False
query hyperthyroid      False
lithium                 False
goitre                  False
tumor                   False
hypopituitary           False
psych                   False
TSH measured            False
TSH                     False
T3 measured             False
T3                      False
TT4 measured            False
TT4                     False
T4U measured            False
T4U                     False
FTI measured            False
FTI                     False
TBG measured            False
TBG                     False
referral source         False
target                  False
dtype: bool
```

In [6]:

```
#Removing attributes from the dataset
data.drop(['TSH measured','T3 measured','TT4 measured','T4U measured','FTI measured','TBG measured','referral source'],axis=1,inplace=True)
```

In [7]:

```
data.head()
```

Out[7]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	tumor	hypopituitary	psych	TSH	T3	TT4	T4U	FTI	TBG	target
0	32	F	f	f	f	f	f	f	f	f	...	f	f	f	56.00	0.1	32	1.00	150	36	S
1	75	F	t	f	t	t	f	t	f	f	...	f	f	f	0.05	1.6	157	0.89	176	78	A
2	63	F	t	t	f	t	t	f	f	t	...	t	f	t	68.00	2.0	48	1.02	47	43	F
3	41	M	f	t	f	f	f	f	f	f	...	f	t	f	0.05	1.6	39	1.00	39	22	R
4	41	F	f	t	f	f	f	f	f	f	...	f	t	f	1.40	2.1	33	1.07	31	34	F

5 rows × 23 columns

In [8]:

```
#re-mapping the 'terget' values to the diagnostic Group
diagnoses={'A':'hyperthyroid conditions',
           'B':'hyperthyroid conditions',
           'C':'hyperthyroid conditions',
           'D':'hyperthyroid conditions',
           'E':'hyperthyroid conditions',
           'F':'hyperthyroid conditions',
           'G':'hyperthyroid conditions',
           'H':'hyperthyroid conditions',
           'I':'binding protien',
           'J':'binding protien',
           'K':'general health',
           'L':'replacement therapy',
           'M':'replacement therapy',
           'N':'replacement therapy',
           'O':'antithyroid treatment',
           'P':'antithyroid treatment',
           'Q':'antithyroid treatment',
           'R':'miscellaneous',
           'S':'miscellaneous',
           'T':'miscellaneous'}
data['target']=data['target'].map(diagnoses)
```

In [9]:

```
#Dropping Null Values
data.dropna(subset=['target'],inplace=True)
```

In [10]:

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

Out[10]:

```
hyperthyroid conditions    19
miscellaneous              10
replacement therapy         8
antithyroid treatment       5
general health              4
binding protien             4
Name: target, dtype: int64
```

In [11]:

```
#Checking whether the age above 100
data[data.age>100]
```

Out[11]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	tumor	hypopituitary	psych	TSH	T3	TT4	T4U	FTI	TBG	target

0 rows × 23 columns

In [12]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 23 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   age                  50 non-null     int64  
 1   sex                  50 non-null     object 
 2   on thyroxine         50 non-null     object 
 3   query on thyroxine   50 non-null     object 
 4   on antithyroid meds  50 non-null     object 
 5   sick                 50 non-null     object 
 6   pregnant             50 non-null     object 
 7   thyroid surgery      50 non-null     object 
 8   I131 treatment       50 non-null     object 
 9   query hypothyroid    50 non-null     object 
 10  query hyperthyroid   50 non-null     object 
 11  lithium              50 non-null     object 
 12  goitre               50 non-null     object 
 13  tumor                50 non-null     object 
 14  hypopituitary        50 non-null     object 
 15  psych                50 non-null     object 
 16  TSH                  50 non-null     float64
 17  T3                   50 non-null     float64
 18  TT4                  50 non-null     int64  
 19  T4U                  50 non-null     float64
 20  FTI                  50 non-null     int64  
 21  TBG                  50 non-null     int64  
 22  target               50 non-null     object 
dtypes: float64(3), int64(4), object(16)
memory usage: 9.4+ KB
```

In [13]:

```
data['sex']=data['sex'].replace({'F':1,'M':0})
data['on thyroxine']=data['on thyroxine'].replace({'f':1,'t':0})
data['query on thyroxine']=data['query on thyroxine'].replace({'f':1,'t':0})
data['on antithyroid meds']=data['on antithyroid meds'].replace({'f':1,'t':0})
data['sick']=data['sick'].replace({'f':1,'t':0})
data['pregnant']=data['pregnant'].replace({'f':1,'t':0})
data['thyroid surgery']=data['thyroid surgery'].replace({'f':1,'t':0})
data['I131 treatment']=data['I131 treatment'].replace({'f':1,'t':0})
data['query hypothyroid']=data['query hypothyroid'].replace({'f':1,'t':0})
data['query hyperthyroid']=data['query hyperthyroid'].replace({'f':1,'t':0})
data['lithium']=data['lithium'].replace({'f':1,'t':0})
data['goitre']=data['goitre'].replace({'f':1,'t':0})
data['tumor']=data['tumor'].replace({'f':1,'t':0})
data['hypopituitary']=data['hypopituitary'].replace({'f':1,'t':0})
data['psych']=data['psych'].replace({'f':1,'t':0})
```

In [14]:

```
data.head()
```

Out[14]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	tumor	hypopituitary	psych	TSH	T3	TT4	T4U	FTI	TBG	target
0	32	1	1	1	1	1	1	1	1	1	...	1	1	1	56.00	0.1	32	1.00	150	36	miscellaneous
1	75	1	0	1	0	0	1	0	1	1	...	1	1	1	0.05	1.6	157	0.89	176	78	hyperthyroid conditions
2	63	1	0	0	1	0	0	1	1	0	...	0	1	0	68.00	2.0	48	1.02	47	43	hyperthyroid conditions
3	41	0	1	0	1	1	1	1	1	1	...	1	0	1	0.05	1.6	39	1.00	39	22	miscellaneous
4	41	1	1	0	1	1	1	1	1	1	...	1	0	1	1.40	2.1	33	1.07	31	34	hyperthyroid conditions

5 rows × 23 columns

In [15]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 23 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
```

```
 0   age                 50 non-null     int64
 1   sex                 50 non-null     int64
 2   on thyroxine        50 non-null     int64
 3   query on thyroxine  50 non-null     int64
 4   on antithyroid meds 50 non-null     int64
 5   sick                50 non-null     int64
 6   pregnant            50 non-null     int64
 7   thyroid surgery     50 non-null     int64
 8   I131 treatment      50 non-null     int64
 9   query hypothyroid   50 non-null     int64
 10  query hyperthyroid  50 non-null     int64
 11  lithium             50 non-null     int64
 12  goitre              50 non-null     int64
 13  tumor               50 non-null     int64
 14  hypopituitary       50 non-null     int64
 15  psych               50 non-null     int64
 16  TSH                 50 non-null     float64
 17  T3                  50 non-null     float64
 18  TT4                 50 non-null     int64
 19  T4U                 50 non-null     float64
 20  FTI                 50 non-null     int64
 21  TBG                 50 non-null     int64
 22  target              50 non-null     object
dtypes: float64(3), int64(19), object(1)
memory usage: 9.4+ KB
```

In [16]:

```
#splitting the data values as x and y
x=data.iloc[:,0:-1]
y=data.iloc[:,-1]
```

In [17]:

```
x.head()
```

Out[17]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	goitre	tumor	hypopituitary	psych	TSH	T3	TT4	T4U	FTI	TBG
0	32	1	1	1	1	1	1	1	1	1	...	1	1	1	1	56.00	0.1	32	1.00	150	36
1	75	1	0	1	0	0	1	0	1	1	...	1	1	1	1	0.05	1.6	157	0.89	176	78
2	63	1	0	0	1	0	0	1	1	0	...	1	0	1	0	68.00	2.0	48	1.02	47	43
3	41	0	1	0	1	1	1	1	1	1	...	0	1	0	1	0.05	1.6	39	1.00	39	22
4	41	1	1	0	1	1	1	1	1	1	...	1	1	0	1	1.40	2.1	33	1.07	31	34

5 rows × 22 columns

In [18]:

```
y.head()
```

Out[18]:

```
0               miscellaneous
1    hyperthyroid conditions
2    hyperthyroid conditions
3               miscellaneous
4    hyperthyroid conditions
Name: target, dtype: object
```

In [19]:

```
x['sex'].unique()
```

Out[19]:

```
array([1, 0], dtype=int64)
```

In [20]:

```
x['sex'].value_counts()
```

Out[20]:

```
1    41
0     9
Name: sex, dtype: int64
```

In [21]:

```
#Converting the datatype
x['age']=x['age'].astype(float)
x['TSH']=x['TSH'].astype(float)
x['T3']=x['T3'].astype(float)
x['TT4']=x['TT4'].astype(float)
x['T4U']=x['T4U'].astype(float)
x['FTI']=x['FTI'].astype(float)
x['TBG']=x['TBG'].astype(float)
x['sex']=x['sex'].astype(int)
```

In [22]:

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 22 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   age                 50 non-null     float64
 1   sex                 50 non-null     int32
 2   on thyroxine        50 non-null     int64
 3   query on thyroxine  50 non-null     int64
 4   on antithyroid meds 50 non-null     int64
 5   sick                50 non-null     int64
 6   pregnant            50 non-null     int64
 7   thyroid surgery     50 non-null     int64
 8   I131 treatment      50 non-null     int64
 9   query hypothyroid   50 non-null     int64
 10  query hyperthyroid  50 non-null     int64
 11  lithium             50 non-null     int64
 12  goitre              50 non-null     int64
 13  tumor               50 non-null     int64
 14  hypopituitary       50 non-null     int64
 15  psych               50 non-null     int64
 16  TSH                 50 non-null     float64
 17  T3                  50 non-null     float64
 18  TT4                 50 non-null     float64
 19  T4U                 50 non-null     float64
 20  FTI                 50 non-null     float64
 21  TBG                 50 non-null     float64
dtypes: float64(7), int32(1), int64(14)
memory usage: 8.8 KB
```

In [23]:

```
x.head()
```

Out[23]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	goitre	tumor	hypopituitary	psych	TSH	T3	TT4	T4U	FTI	TBG
0	32.0	1	1	1	1	1	1	1	1	1	...	1	1	1	1	56.00	0.1	32.0	1.00	150.0	36.0

1	75.0	1	0	1	0	0	1	0	1	1	...	1	1	1	1	0.05	1.6	157.0	0.89	176.0	78.0
2	63.0	1	0	0	1	0	0	1	1	0	...	1	0	1	0	68.00	2.0	48.0	1.02	47.0	43.0
3	41.0	0	1	0	1	1	1	1	1	1	...	0	1	0	1	0.05	1.6	39.0	1.00	39.0	22.0
4	41.0	1	1	0	1	1	1	1	1	1	...	1	1	0	1	1.40	2.1	33.0	1.07	31.0	34.0

5 rows × 22 columns

In [24]:

```
#Encoding the categorical data
#Encoding the independent(output) variable
from sklearn.preprocessing import OrdinalEncoder,LabelEncoder
#categorical data
ordinal_encoder=OrdinalEncoder(dtype='int64')
x.iloc[:,1:16]=ordinal_encoder.fit_transform(x.iloc[:,1:16])
#ordinal_encoder,fit_transform(x[['sex']])
```

In [25]:

```
x.head()
```

Out[25]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	goitre	tumor	hypopituitary	psych	TSH	T3	TT4	T4U	FTI	TBG
0	32.0	1	1	1	1	1	1	1	1	1	...	1	1	1	1	56.00	0.1	32.0	1.00	150.0	36.0
1	75.0	1	0	1	0	0	1	0	1	1	...	1	1	1	1	0.05	1.6	157.0	0.89	176.0	78.0
2	63.0	1	0	0	1	0	0	1	1	0	...	1	0	1	0	68.00	2.0	48.0	1.02	47.0	43.0
3	41.0	0	1	0	1	1	1	1	1	1	...	0	1	0	1	0.05	1.6	39.0	1.00	39.0	22.0
4	41.0	1	1	0	1	1	1	1	1	1	...	1	1	0	1	1.40	2.1	33.0	1.07	31.0	34.0

5 rows × 22 columns

In [26]:

```
label_encoder=LabelEncoder()
y_dt=label_encoder.fit_transform(y)
y=pd.DataFrame(y_dt,columns=['target'])
```

In [27]:

```
y.head()
```

Out[27]:

	target
0	4
1	3
2	3
3	4
4	3

In [28]:

```
#Splitting data into train and test
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
```

In [29]:

```
X_train.head()
```

Out[29]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	goitre	tumor	hypopituitary	psych	TSH	T3	TT4	T4U	FTI	TBG
33	65.0	1	1	1	1	0	1	1	1	1	...	0	1	1	0	1.5	2.2	44.0	0.98	105.0	34.0
35	65.0	1	1	1	0	1	1	0	0	0	...	1	1	0	1	176.0	2.3	45.0	1.38	99.0	132.0
26	45.0	1	1	1	1	1	0	1	0	0	...	1	1	1	1	4.1	2.6	213.0	1.22	106.0	31.0
34	45.0	1	1	0	1	1	0	1	0	1	...	1	1	1	1	0.8	1.5	130.0	0.99	103.0	36.0
18	55.0	0	1	1	1	1	1	1	0	1	...	1	0	1	0	0.9	1.6	105.0	1.06	99.0	27.0

5 rows × 22 columns

In [30]:

```
pip install imblearn
```

```
Requirement already satisfied: imblearn in c:\users\elcot\appdata\roaming\python\python38\site-packages (0.0)
Requirement already satisfied: imbalanced-learn in c:\users\elcot\appdata\roaming\python\python38\site-packages (from imblearn) (0.10.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from imbalanced-learn->imblearn) (2.1.0)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\elcot\appdata\roaming\python\python38\site-packages (from imbalanced-learn->imblearn) (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in c:\programdata\anaconda3\lib\site-packages (from imbalanced-learn->imblearn) (1.20.1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\elcot\appdata\roaming\python\python38\site-packages (from imbalanced-learn->imblearn) (1.2.0)
Requirement already satisfied: scipy>=1.3.2 in c:\programdata\anaconda3\lib\site-packages (from imbalanced-learn->imblearn) (1.6.2)
Note: you may need to restart the kernel to use updated packages.
```

In [31]:

```
#Handling Imbalanced Data
from imblearn.over_sampling import SMOTE
smote=SMOTE(random_state=0,k_neighbors=1)
```

In [32]:

```
y_train.value_counts()
```

Out[32]:

```
target
3         13
4          8
5          7
0          5
1          4
2          3
dtype: int64
```

In [33]:

```
X_bal,y_bal=smote.fit_resample(X_train,y_train)
```

In [34]:

```
#Applying StandardScaler
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_bal=sc.fit_transform(X_bal)
```

In [35]:

```
X_bal
```

Out[35]:

```
array([[ 0.89432169,  0.60697698,  0.79056942, ..., -0.34826475,
         0.07443767, -0.64835438],
       [ 0.89432169,  0.60697698,  0.79056942, ...,  1.09066114,
        -0.04597256,  2.07746051],
       [-0.36197595,  0.60697698,  0.79056942, ...,  0.51509078,
         0.09450604, -0.73179769],
       ...,
       [-0.21920522, -1.64750894, -1.26491106, ..., -0.51533762,
        -0.29972462, -0.9798262 ],
       [-0.38560423, -1.64750894, -1.26491106, ..., -0.67127363,
        -0.38671678, -1.02671443],
       [-0.23639523, -1.64750894, -1.26491106, ..., -0.53144674,
        -0.30871143, -0.98467004]])
```

In [36]:

```python
columns=['age','sex','on thyroxine','query on thyroxine','on antithyroid meds','sick','pregnant','thyroid surgery','I131 treatment','query hypothyroid','query
```

In [37]:

```python
X_bal=pd.DataFrame(X_bal,columns=columns)
```

In [38]:

```python
X_bal
```

Out[38]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	goitre	tumor	hypopituitary	psych	TSH	T3	TT4	
0	0.894322	0.606977	0.790569	0.902194	0.748331	-1.456022	0.769309	0.547723	0.834058	0.405190	...	-1.595448	0.707107	0.606977	-1.374369	-0.170024	0.084903	-1.141962	-0.3
1	0.894322	0.606977	0.790569	0.902194	-1.336306	0.686803	0.769309	-1.825742	-1.198958	-2.467977	...	0.626783	0.707107	-1.647509	0.727607	8.218877	0.172730	-1.122568	1.0
2	-0.361976	0.606977	0.790569	0.902194	0.748331	0.686803	-1.299867	0.547723	-1.198958	-2.467977	...	0.626783	0.707107	0.606977	0.727607	-0.045032	0.436210	2.135608	0.5
3	-0.361976	0.606977	0.790569	-1.108409	0.748331	0.686803	-1.299867	0.547723	-1.198958	0.405190	...	0.626783	0.707107	0.606977	0.727607	-0.203676	-0.529885	0.525914	-0.3
4	0.266173	-1.647509	0.790569	0.902194	0.748331	0.686803	0.769309	0.547723	-1.198958	0.405190	...	0.626783	-1.414214	0.606977	-1.374369	-0.198869	-0.442058	0.041066	-0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
73	-0.779478	0.606977	-1.264911	-1.108409	0.748331	-1.456022	0.769309	0.547723	0.834058	-2.467977	...	0.626783	-1.414214	-1.647509	0.727607	-0.201355	-0.970029	-0.881147	-0.6
74	0.000579	-1.647509	-1.264911	-1.108409	0.748331	-1.456022	0.769309	0.547723	-1.198958	0.405190	...	0.626783	-1.414214	0.606977	-1.374369	-0.191477	-0.543336	-0.212392	-0.3
75	-0.219205	-1.647509	-1.264911	-1.108409	0.748331	-1.456022	0.769309	0.547723	-1.198958	0.405190	...	0.626783	-1.414214	0.606977	-1.374369	-0.185361	-0.627144	-0.422135	-0.5
76	-0.385604	-1.647509	-1.264911	-1.108409	0.748331	-1.456022	0.769309	0.547723	-1.198958	0.405190	...	0.626783	-1.414214	0.606977	-1.374369	-0.180730	-0.690596	-0.580931	-0.6
77	-0.236395	-1.647509	-1.264911	-1.108409	0.748331	-1.456022	0.769309	0.547723	-1.198958	0.405190	...	0.626783	-1.414214	0.606977	-1.374369	-0.184882	-0.633699	-0.438539	-0.5

78 rows × 22 columns

In [40]:

```python
X_bal.drop=(['age','sex','on thyroxine','query on thyroxine','on antithyroid meds','sick','pregnant','thyroid surgery','I131 treatment','query hypothyroid','q
```

In [41]:

```python
X_bal.head()
```

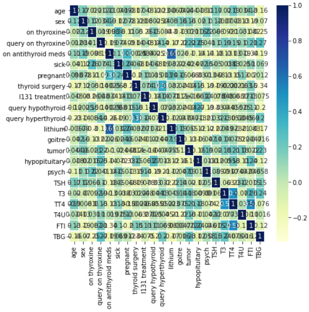
Out[41]:

	age	sex	on thyroxine	query on thyroxine	on antithyroid meds	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	...	goitre	tumor	hypopituitary	psych	TSH	T3	TT4	
0	0.894322	0.606977	0.790569	0.902194	0.748331	-1.456022	0.769309	0.547723	0.834058	0.405190	...	-1.595448	0.707107	0.606977	-1.374369	-0.170024	0.084903	-1.141962	-0.34
1	0.894322	0.606977	0.790569	0.902194	-1.336306	0.686803	0.769309	-1.825742	-1.198958	-2.467977	...	0.626783	0.707107	-1.647509	0.727607	8.218877	0.172730	-1.122568	1.09
2	-0.361976	0.606977	0.790569	0.902194	0.748331	0.686803	-1.299867	0.547723	-1.198958	-2.467977	...	0.626783	0.707107	0.606977	0.727607	-0.045032	0.436210	2.135608	0.51
3	-0.361976	0.606977	0.790569	-1.108409	0.748331	0.686803	-1.299867	0.547723	-1.198958	0.405190	...	0.626783	0.707107	0.606977	0.727607	-0.203676	-0.529885	0.525914	-0.31
4	0.266173	-1.647509	0.790569	0.902194	0.748331	0.686803	0.769309	0.547723	-1.198958	0.405190	...	0.626783	-1.414214	0.606977	-1.374369	-0.198869	-0.442058	0.041066	-0.06

5 rows × 22 columns

In [42]:

```python
import seaborn as sns
plt.figure(figsize=(7,7))
sns.heatmap(x.corr(),cmap="YlGnBu",annot=True,linewidth=0.1)
plt.show()
```



In [43]:

```python
#Model Building
```

In [44]:

```python
#RandomForestClassifier Model
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
rfc=RandomForestClassifier().fit(X_bal,y_bal.values.ravel())
y_pred=rfc.predict(X_test)
```

In [45]:

```
rfc.fit(X_bal,y_bal.values.ravel())
```

Out[45]:

```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [46]:

```
y_pred=rfc.predict(X_test)
```

In [47]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.60      1.00      0.75         6
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.60        10
   macro avg       0.15      0.25      0.19        10
weighted avg       0.36      0.60      0.45        10
```

In [55]:

```
from sklearn.metrics import accuracy_score,classification_report
train_score=accuracy_score(y_bal,rfc.predict(X_bal))
train_score
```

Out[55]:

```
1.0
```

In [56]:

```
#XGBClassifier model
from xgboost import XGBClassifier
xgb1=XGBClassifier()
xgb1.fit(X_bal,y_bal)
```

Out[56]:

```
▾                              XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              objective='multi:softprob', predictor=None, ...)
```

In [57]:

```
y_pred=xgb1.predict(X_test)
```

In [58]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.60      1.00      0.75         6
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.60        10
   macro avg       0.15      0.25      0.19        10
weighted avg       0.36      0.60      0.45        10
```

In [59]:

```
accuracy_score(y_test,y_pred)
```

Out[59]:

```
0.6
```

In [60]:

```
#SVC model
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score,classification_report
svc=SVC()
```

In [61]:

```
svc.fit(X_bal,y_bal)
```

Out[61]:

```
▾ SVC
SVC()
```

In [62]:

```
y_pred=svc.predict(X_test)
```

In [63]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.60      1.00      0.75         6
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.60        10
   macro avg       0.15      0.25      0.19        10
weighted avg       0.36      0.60      0.45        10
```

In [64]:

```
train_score=accuracy_score(y_bal,svc.predict(X_bal))
train_score
```

Out[64]:

```
0.9615384615384616
```

In [66]:

```
#Compare the model
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.60      1.00      0.75         6
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.60        10
   macro avg       0.15      0.25      0.19        10
weighted avg       0.36      0.60      0.45        10
```

In [67]:

```
train_score=accuracy_score(y_bal,rfc.predict(X_bal))
```

In [68]:

```
train_score
```

Out[68]:

```
1.0
```

In [69]:

```
y_pred=xgb1.predict(X_test)
```

In [70]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.60      1.00      0.75         6
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.60        10
   macro avg       0.15      0.25      0.19        10
weighted avg       0.36      0.60      0.45        10
```

In [71]:

```
train_score=accuracy_score(y_bal,xgb1.predict(X_bal))
train_score
```

Out[71]:

```
1.0
```

In [72]:

```
y_pred=svc.predict(X_test)
```

In [73]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.60      1.00      0.75         6
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.60        10
   macro avg       0.15      0.25      0.19        10
weighted avg       0.36      0.60      0.45        10
```

In [74]:

```
train_score=accuracy_score(y_test,y_pred)
```

In [75]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.60      1.00      0.75         6
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.60        10
   macro avg       0.15      0.25      0.19        10
weighted avg       0.36      0.60      0.45        10
```

In [76]:

```
train_score=accuracy_score(y_bal,svc.predict(X_bal))
train_score
```

Out[76]:

```
0.9615384615384616
```

In [77]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         1
           3       0.60      1.00      0.75         6
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.60        10
   macro avg       0.15      0.25      0.19        10
weighted avg       0.36      0.60      0.45        10
```

In [78]:

```
import pickle
pickle.dump(svc,open('thyroid_1_model.pkl','wb'))
```

In [79]:

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

In [80]:

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

Out[80]:

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

In [81]:

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

Out[81]:

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

In [ ]:

```

```