# Exploratory Data Analysis (EDA)

In [3]:
```
#Importing the Required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plot
import seaborn as sns
from sklearn.utils import resample
from imblearn.over_sampling import SMOTENC,RandomOverSampler,KMeansSMOTE
from sklearn.impute import KNNImputer
from sklearn.preprocessing import LabelEncoder
sns.set()
```

In [4]:
```
df = pd.read_csv("InputFile.csv")
```

In [5]:
```
df
```

Out[5]:

| | age | sex | on_thyroxine | query_on_thyroxine | on_antithyroid_medication | sick | pregnant |
|---|---|---|---|---|---|---|---|
| **0** | 41 | F | f | f | f | f | f |
| **1** | 23 | F | f | f | f | f | f |
| **2** | 46 | M | f | f | f | f | f |
| **3** | 70 | F | t | f | f | f | f |
| **4** | 70 | F | f | f | f | f | f |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **3767** | 30 | F | f | f | f | f | f |
| **3768** | 68 | F | f | f | f | f | f |
| **3769** | 74 | F | f | f | f | f | f |
| **3770** | 72 | M | f | f | f | f | f |
| **3771** | 64 | F | f | f | f | f | f |

3772 rows × 30 columns

Problem Statement : To build a classification methodology to predict the type of Thyroid a person has ,based on the below features.

age - Age of the person

sex - Male or Female

on_thyroxine - true or false

on_antithyroid_medication - true or false

sick - true or false

pregnant - true or false

thyroid_surgery - true or false

I131_treatment - true or false

query_hypothyroid - true or false

query_hyperthyroid -true or false

lithium - true or false

goitre - true or false

tumor - true or false

hypopituitary- true or false

psych - true or false

TSH_measured - true or false

TSH - thyroid stimulating hormone floating value

T3_measured - true or false

T3 - triiodothyronine value

TT4_measured- true or false

TT4 - Thyroxine value

T4U_measured- true or false

T4U - numerical value

FTI_measured- true or false

FTI -Free Thyroxine Index

TBG_measured- true or false

TBG -Thyroid-Binding Globulin value

referral_source - different sources of referals

Class - different types of thyroid

In [6]:
```
df.describe()
```

Out[6]:

| | age | sex | on_thyroxine | query_on_thyroxine | on_antithyroid_medication | sick | preg |
|---|---|---|---|---|---|---|---|
| **count** | 3772 | 3772 | 3772 | 3772 | 3772 | 3772 | |
| **unique** | 94 | 3 | 2 | 2 | 2 | 2 | |
| **top** | 59 | F | f | f | f | f | |
| **freq** | 95 | 2480 | 3308 | 3722 | 3729 | 3625 | |

4 rows × 30 columns

In [8]:
```
for column in df.columns:
    count = df[column][df[column]=="?"].count()
    if count != 0:
        print(column, count)
```

```
age 1
sex 150
TSH 369
T3 769
TT4 231
T4U 387
FTI 385
TBG 3772
```

In [9]:
```
df = df.drop(["TBG"], axis = 1)
```

In [10]:
```
df[["T4U_measured", "T4U"]]
```

Out[10]:

| | T4U_measured | T4U |
|---|---|---|
| **0** | t | 1.14 |
| **1** | f | ? |
| **2** | t | 0.91 |
| **3** | f | ? |
| **4** | t | 0.87 |
| **...** | ... | ... |
| **3767** | f | ? |
| **3768** | t | 1.08 |
| **3769** | t | 1.07 |
| **3770** | t | 0.94 |
| **3771** | t | 1.07 |

3772 rows × 2 columns

In [11]:
```
df = df.drop(['TSH_measured','T3_measured','TT4_measured','T4U_measured','F
```

In [12]:
```
for column in df.columns:
    count = df[column][df[column] == "?"].count()
    if count!=0:
        df[column] = df[column].replace("?", np.nan)
```

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
age                          1
sex                        150
on_thyroxine                 0
query_on_thyroxine           0
on_antithyroid_medication    0
sick                         0
pregnant                     0
thyroid_surgery              0
I131_treatment               0
query_hypothyroid            0
query_hyperthyroid           0
lithium                      0
goitre                       0
tumor                        0
hypopituitary                0
psych                        0
TSH                        369
T3                         769
TT4                        231
T4U                        387
FTI                        385
referral_source              0
Class                        0
dtype: int64
```

In [15]:
```
df['sex'] = df['sex'].map({"F" : 0, "M" : 1})
```

In [16]:
```
for column in df.columns:
    if len(df[column].unique()) == 2:
        df[column] = df[column].map({"f":0, "t":1})
```

In [17]:
```
df.head()
```

Out[17]:

| | age | sex | on_thyroxine | query_on_thyroxine | on_antithyroid_medication | sick | pregnant | tl |
|---|---|---|---|---|---|---|---|---|
| **0** | 41 | 0.0 | 0 | 0 | 0 | 0 | 0 | |
| **1** | 23 | 0.0 | 0 | 0 | 0 | 0 | 0 | |
| **2** | 46 | 1.0 | 0 | 0 | 0 | 0 | 0 | |
| **3** | 70 | 0.0 | 1 | 0 | 0 | 0 | 0 | |
| **4** | 70 | 0.0 | 0 | 0 | 0 | 0 | 0 | |

5 rows × 23 columns

In [18]:
```
df = df.drop(columns=["referral_source"], axis = True)
```

In [19]:
```
df["Class"].unique()
```

Out[19]:
```
array(['negative', 'compensated_hypothyroid', 'primary_hypothyroid',
       'secondary_hypothyroid'], dtype=object)
```

In [20]:
```
lbn = LabelEncoder()
df["Class"] = lbn.fit_transform(df["Class"])
```

In [21]:
```
df.head()
```

Out[21]:

| | age | sex | on_thyroxine | query_on_thyroxine | on_antithyroid_medication | sick | pregnant | tl |
|---|---|---|---|---|---|---|---|---|
| **0** | 41 | 0.0 | 0 | 0 | 0 | 0 | 0 | |
| **1** | 23 | 0.0 | 0 | 0 | 0 | 0 | 0 | |
| **2** | 46 | 1.0 | 0 | 0 | 0 | 0 | 0 | |
| **3** | 70 | 0.0 | 1 | 0 | 0 | 0 | 0 | |
| **4** | 70 | 0.0 | 0 | 0 | 0 | 0 | 0 | |

5 rows × 22 columns

In [22]:
```
df.describe(include='all')
```

Out[22]:

| | age | sex | on_thyroxine | query_on_thyroxine | on_antithyroid_medication | |
|---|---|---|---|---|---|---|
| **count** | 3771 | 3622.000000 | 3772.000000 | 3772.000000 | 3772.000000 | 377 |
| **unique** | 93 | NaN | NaN | NaN | NaN | |
| **top** | 59 | NaN | NaN | NaN | NaN | |
| **freq** | 95 | NaN | NaN | NaN | NaN | |
| **mean** | NaN | 0.315295 | 0.123012 | 0.013256 | 0.011400 | |
| **std** | NaN | 0.464698 | 0.328494 | 0.114382 | 0.106174 | |
| **min** | NaN | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **25%** | NaN | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **50%** | NaN | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **75%** | NaN | 1.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **max** | NaN | 1.000000 | 1.000000 | 1.000000 | 1.000000 | |

11 rows × 22 columns

In [23]:
```
imputer = KNNImputer(n_neighbors=3, weights='uniform', missing_values=np.na
new_arr = imputer.fit_transform(df)
```

In [25]:
```
new_df = pd.DataFrame(new_arr)
```

In [26]:
```
new_df
```

Out[26]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 12 | 13 | 14 | 15 | 16 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 41.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 1.300000 | 2.5000 |
| **1** | 23.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 4.100000 | 2.0000 |
| **2** | 46.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.980000 | 1.6333 |
| **3** | 70.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.160000 | 1.9000 |
| **4** | 70.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.720000 | 1.2000 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **3767** | 30.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 1.0 | 0.0 | 0.0 | 1.566667 | 2.5333 |
| **3768** | 68.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 1.000000 | 2.1000 |
| **3769** | 74.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 5.100000 | 1.8000 |
| **3770** | 72.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.700000 | 2.0000 |
| **3771** | 64.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 1.000000 | 2.2000 |

3772 rows × 22 columns

In [27]:
```
new_df.isnull().sum()
```

Out[27]:
```
0     0
1     0
2     0
3     0
4     0
5     0
6     0
7     0
8     0
9     0
10    0
11    0
12    0
13    0
14    0
15    0
16    0
17    0
18    0
19    0
20    0
21    0
dtype: int64
```

In [29]:
```
new_df.describe()
```

Out[29]:

| | 0 | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| **count** | 3772.000000 | 3772.000000 | 3772.000000 | 3772.000000 | 3772.000000 | 3772.000000 | 3772. |
| **mean** | 51.737275 | 0.308855 | 0.123012 | 0.013256 | 0.011400 | 0.038971 | 0. |
| **std** | 20.082478 | 0.458819 | 0.328494 | 0.114382 | 0.106174 | 0.193552 | 0. |
| **min** | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0. |
| **25%** | 36.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0. |
| **50%** | 54.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0. |
| **75%** | 67.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0. |
| **max** | 455.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1. |

8 rows × 22 columns

In [32]:
```
columns = df.columns
```

In [34]:
```
new_df = pd.DataFrame(new_arr, columns=columns)
```

In [35]:
```
new_df
```

Out[35]:

| | age | sex | on_thyroxine | query_on_thyroxine | on_antithyroid_medication | sick | pregnan |
|---|---|---|---|---|---|---|---|
| **0** | 41.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **1** | 23.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **2** | 46.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **3** | 70.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **4** | 70.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **...** | ... | ... | ... | ... | ... | ... | .. |
| **3767** | 30.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **3768** | 68.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **3769** | 74.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **3770** | 72.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |
| **3771** | 64.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.( |

3772 rows × 22 columns

In [39]:
```
columns = ['age','TSH','T3','TT4','T4U','FTI']

plot.figure(figsize=(10,15),facecolor='white')
plotnumber = 1

for column in columns:
    ax = plot.subplot(3,2,plotnumber)
    sns.distplot(new_df[column])
    plot.xlabel(column,fontsize=10)
    plotnumber+=1
plot.show()
```

```
similar flexibility) or `histplot` (an axes-level function for histogr
ams).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http
s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(new_df[column])
C:\Users\Dinesh\AppData\Local\Temp\ipykernel_39476\1990233037.py:8: Us
erWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.
14.0.

Please adapt your code to use either `displot` (a figure-level functio
n with
similar flexibility) or `histplot` (an axes-level function for histogr
ams).

For a guide to updating your code to use the new functions, please see
```

In [40]:
```
plot.figure(figsize=(10,15),facecolor='white')
plotnumber = 1

for column in columns:
    new_df[column]+=1
    ax = plot.subplot(3,2,plotnumber)
    sns.distplot(np.log(new_df[column]))
    plot.xlabel(column,fontsize=10)
    plotnumber+=1
plot.show()
```

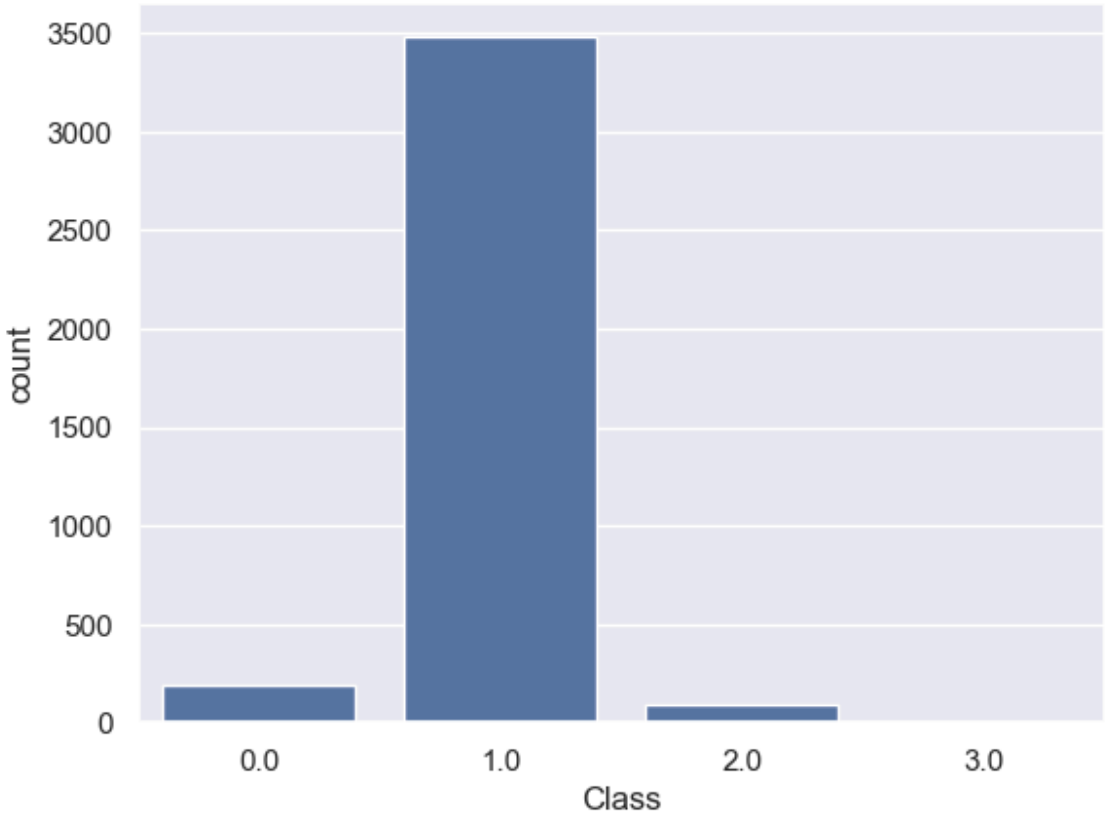
```
C:\Users\Dinesh\AppData\Local\Temp\ipykernel_39476\3553379851.py:7: Us
erWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.
14.0.

Please adapt your code to use either `displot` (a figure-level functio
n with
similar flexibility) or `histplot` (an axes-level function for histogr
ams).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http
s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(np.log(new_df[column]))
C:\Users\Dinesh\AppData\Local\Temp\ipykernel_39476\3553379851.py:7: Us
erWarning:
```

In [41]:
```
new_df = new_df.drop(columns="TSH")
```

In [43]:
```
new_df.shape
```

Out[43]:
```
(3772, 21)
```

In [52]:
```
sns.countplot(data = new_df, x = "Class")
plot.show()
```




In [54]:
```
x = new_df.drop(["Class"], axis = 1)
y = new_df["Class"]
rdsmple = RandomOverSampler()
x_sampled, y_sampled = rdsmple.fit_resample(x, y)
```

In [55]:
```
x_sampled.shape
```

Out[55]: (13924, 20)

In [56]:
```
y_sampled.shape
```

Out[56]: (13924,)

In [62]:
```
df_y = pd.DataFrame(y_sampled, columns=["Class"])
```

In [63]:
```
df_y
```

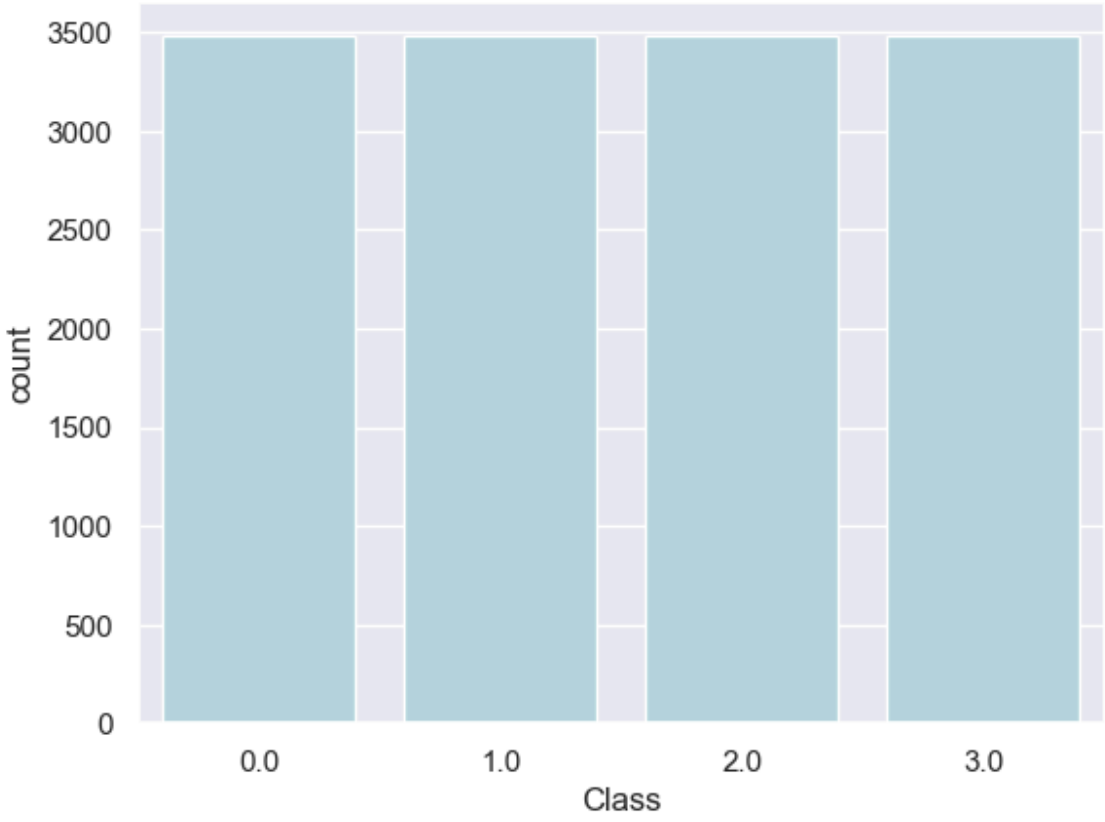
Out[63]:

| | Class |
|---|---|
| **0** | 1.0 |
| **1** | 1.0 |
| **2** | 1.0 |
| **3** | 1.0 |
| **4** | 1.0 |
| **...** | ... |
| **13919** | 3.0 |
| **13920** | 3.0 |
| **13921** | 3.0 |
| **13922** | 3.0 |
| **13923** | 3.0 |

13924 rows × 1 columns

In [67]:
```
sns.countplot(data = df_y, x = "Class", color="lightblue")
```

Out[67]:
```
<AxesSubplot: xlabel='Class', ylabel='count'>
```




In [68]:
```
df_x = pd.DataFrame(x_sampled, columns = x.columns)
```

In [69]:
```
df_x
```

Out[69]:

| nant | thyroid_surgery | I131_treatment | query_hypothyroid | query_hyperthyroid | lithium | goitre |
|---|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

In [71]:
```
merged_df = pd.merge(df_x, df_y, left_index=True, right_index=True)
```

In [72]:
```
merged_df
```

Out[72]:

| | age | sex | on_thyroxine | query_on_thyroxine | on_antithyroid_medication | sick | pregna |
|---|---|---|---|---|---|---|---|
| **0** | 42.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| **1** | 24.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| **2** | 47.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| **3** | 71.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0 |
| **4** | 71.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | |
| **13919** | 42.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| **13920** | 47.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| **13921** | 42.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| **13922** | 47.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| **13923** | 47.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |

13924 rows × 21 columns

In [81]:
```
merged_df.to_csv("Data/preprocessed_data.csv", index=False)
```

In [76]:
```
# for saving the label encoder for model output printing to front end
from joblib import dump
dump(lbn, "Encoder/label_encoder_class.joblib")
```

Out[76]:
```
['Encoder/label_encoder_class.joblib']
```

In [ ]:

In [ ]: