ARIGNAR ANNA GOVERNTMENT ARTS COLLEGE

VILLUPURAM

THYROID DISEASE CLASSIFICATION using machine learning

Department of computer science

TEAM LEADER :

1) D.Kavitha

MEMBERS:

2) V.Kaviya

3) V.Keerthi

4) S.Monishaa

INTRODUCTION

OVERVIEW :

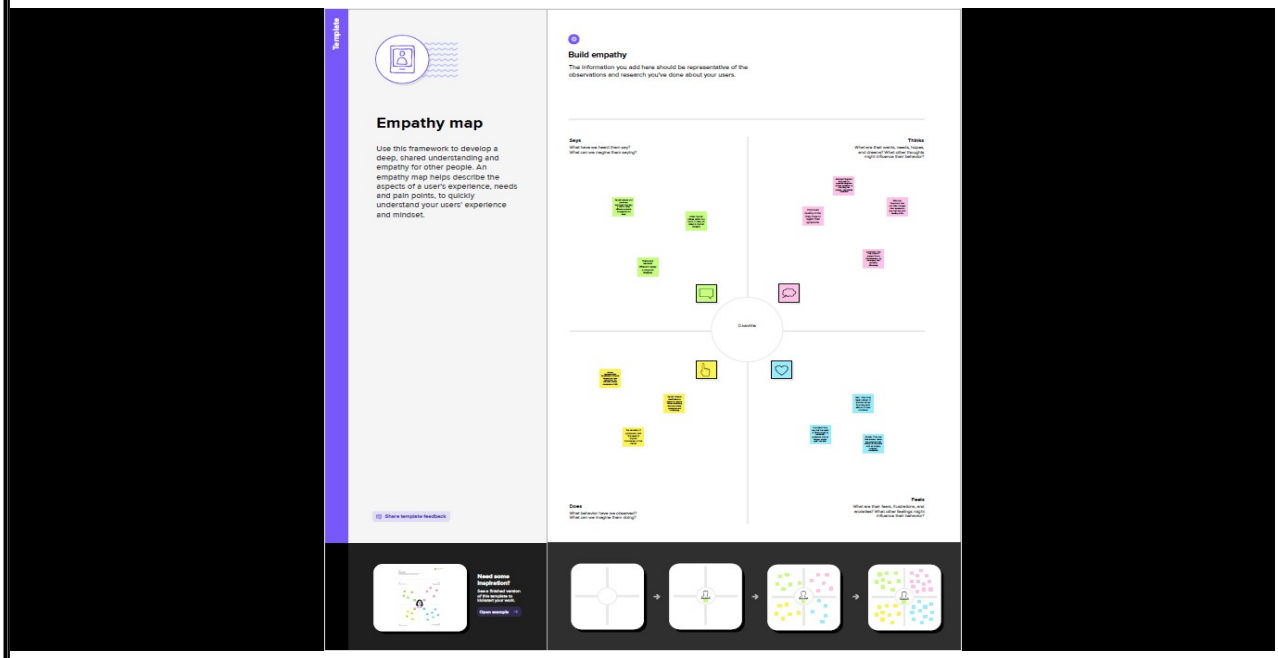
Thyroid disease classification project is used to know the thyroid disease. Thyroid gland secrets two hormones which help in controlling the metabolism of the body. When the disorder occurs in the body, they releases certain types of hormones into the body which imbalances the body's metabolism. A thyroid – related Blood test is used to detect this disease but it is often blurred and noise will be present. Data cleansing meyhods were used to make the data primitive enough for the analytics to show the risk of patients getting the disease.

PURPOSE :

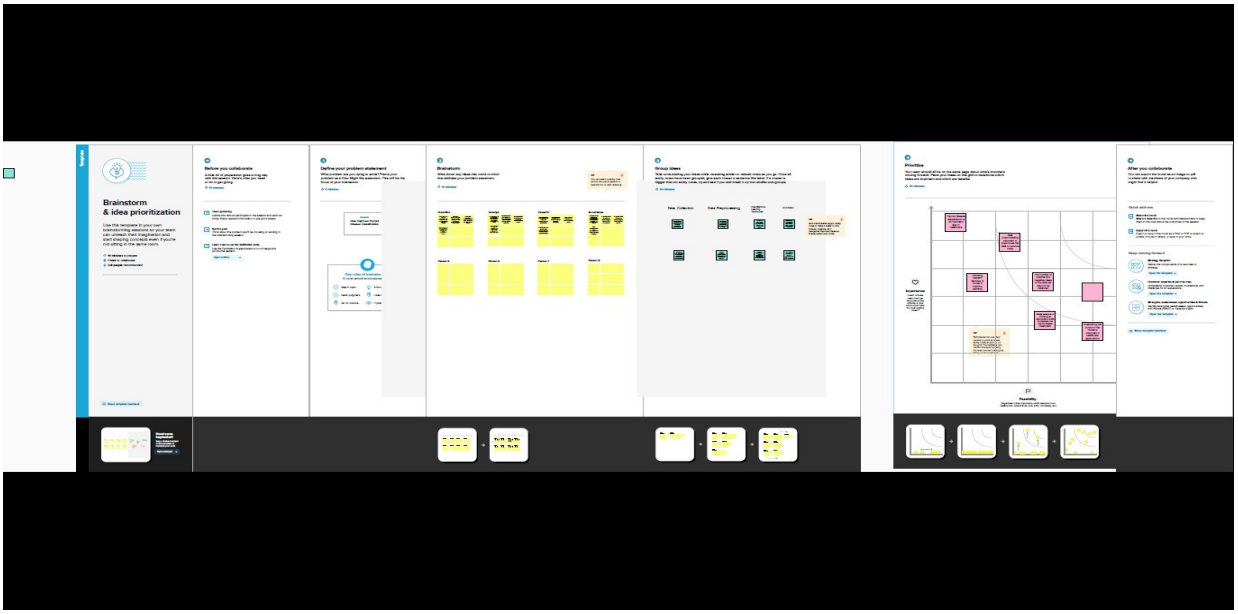
The proposed framework will take input in the form of dataset and then forward to the preprocessing module. Machine Learning plays a very deciding role in disease prediction. Support vector machines, random forest, decision, decision tree, naive bayes, logistics regression, k-nearset neighbors, multi-layer perceptron (MLP) , linear discriminant analysis To classification of thyroid disease.

PROBLEM DEFINITON & DESIGN THINKING

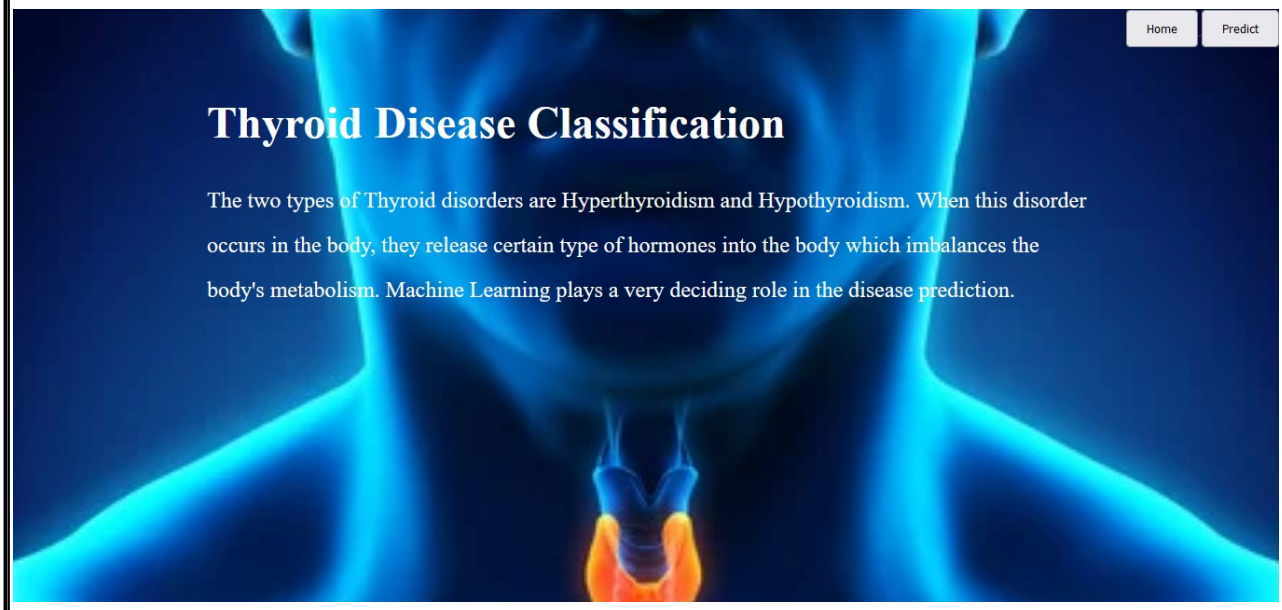
EMPATHY MAP

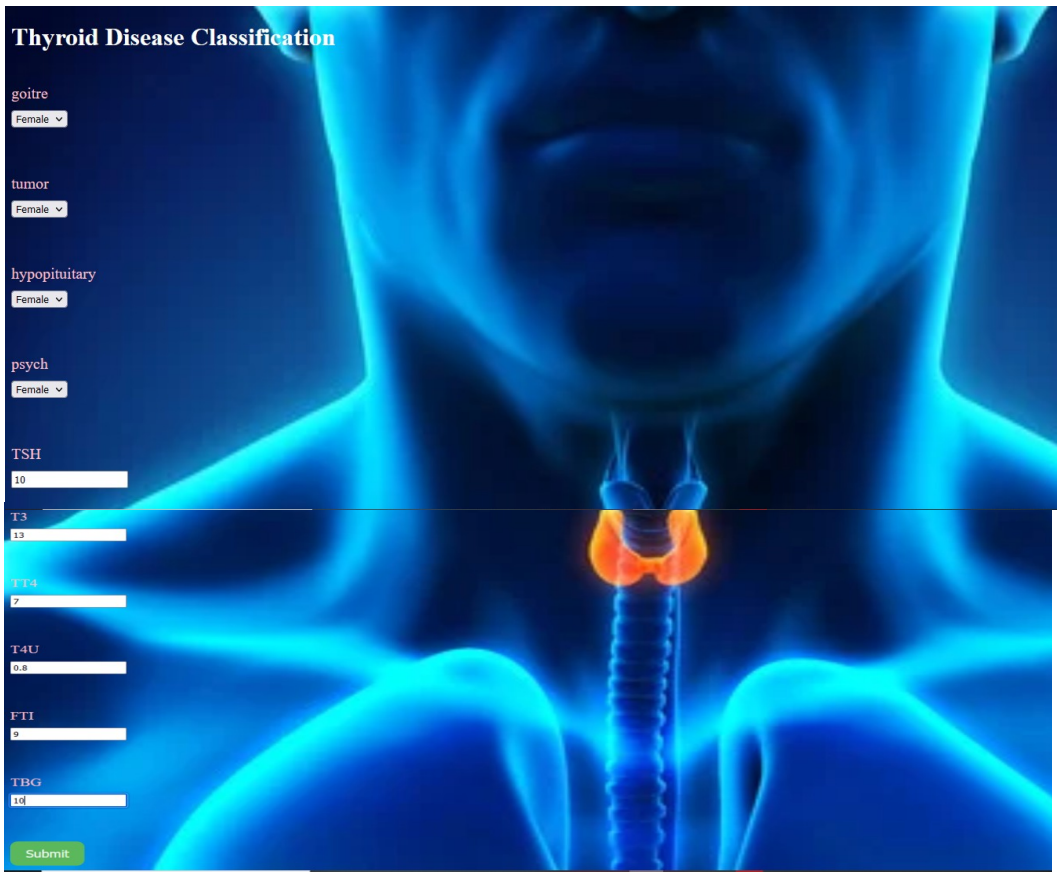


IDEATION & BRAINSTROMING MAP



RESULT





Result:

Based on the given input, it predicts Thyroid disease for your body condition is ['hyperthyroid conditions']

ADVANTAGES:

Thyroid disease classification is an important task in healthcare.

1). Improved accuracy: Machine learning algorithms can learn from large datasets and identify patterns that may not be apparent to human observers. With a well-trained model, it may be possible to accurately classify thyroid diseases with a high degree of accuracy.

2).Faster diagnosis: Traditional methods for diagnosing thyroid diseases can be time-consuming and costly. Machine learning models can process large amounts of data in a short period of time, potentially leading to faster diagnosis and treatment.

3).Personalized treatment: As mentioned earlier, different thyroid diseases require different treatment approaches. By accurately classifying thyroid diseases, a machine learning model can help healthcare providers develop more personalized treatment plans for their patients

DISADVANTAGES

1).limited Data Availability: The availability of labeled data is crucial for building accurate machine learning models. In the case of thyroid disease classification, there may be limited labeled data available for training, especially for rare or uncommon types of thyroid diseases.

2).Data Imbalance: The distribution of different types of thyroid diseases may be uneven in the dataset, leading to an imbalanced classification problem. This can result in a biased model that performs well on the majority class but poorly on the minority classes.

3).Feature Engineering: The choice and quality of features used to train the machine learning model can significantly impact its performance. In the case of thyroid disease classification, identifying relevant features from the available data may require expert knowledge, which may not always be available

APPLICATION

→ DISEASE PREDICTION

→ HEALTH CARE

→ MORE ACCURATE HEALTH RECORDS

→ VISUALIZATION OF MEDICAL DATA

→ IMPROVED DIAGNOSIS

CONCLUSION

machine learning has the potential to assist in the classification of thyroid diseases, but there are several challenges that need to be addressed to build accurate and clinically relevant models. Limited data availability, data imbalance, feature engineering, model interpretability, generalization, and ethical concerns are some of the main challenges that must be taken into account when developing machine learning models for thyroid disease classification. Despite these challenges, machine learning can offer valuable insights into the diagnosis and treatment of thyroid diseases and has the potential to improve patient outcomes. It is important to continue exploring and developing machine learning approaches for thyroid disease classification while addressing the limitations and challenges to ensure safe and effective clinical use

FUTURE SCOPE

Improved Data Collection: To build more accurate and robust machine learning models for thyroid disease classification, a large and diverse dataset with well-labeled samples is needed. There is a need for more standardized data collection protocols and collaboration between healthcare institutions to create a larger pool of data.

Integration with Electronic Health Records (EHRs): EHRs contain a wealth of patient data, including medical history, laboratory test results, and imaging data, which can be used to train machine learning models. Integrating machine learning algorithms with EHRs can help clinicians diagnose thyroid diseases more accurately and efficiently

APPENDIX

SOURCE CODE

app - Notepad

File Edit Format View Help

```
from flask import Flask, render_template, request, url_for
import numpy as np
import pickle
import pandas as pd

model = pickle.load(open('thyroid_1_model.pkl', 'rb'))
le = pickle.load(open("label_encoder.pkl", "rb"))

app = Flask(__name__)

# home page
@app.route("/")
@app.route("/home")
def home():
        return render_template("home.html")


# predict page
@app.route("/predict")
def formPage():
        return render_template("predict.html")

# submit page
@app.route("/submit", methods=['POST'])
def predict():

        goitre = request.form.get("goitre")
        tumor = request.form.get("tumor")
        hypopituitary = request.form.get("hypopituitary")
        psych = request.form.get("psych")
        TSH = request.form.get("TSH")
        T3 = request.form.get("T3")
        TT4 = request.form.get("TT4")
        T4U = request.form.get("T4U")
        FTI = request.form.get("FTI")
        TBG = request.form.get("TBG")
```

Ln 20, Col 16 100% Windows (CRLF) UTF-8

app - Notepad

File Edit Format View Help

```
# predict page
@app.route("/predict")
def formPage():
        return render_template("predict.html")

# submit page
@app.route("/submit", methods=['POST'])
def predict():

        goitre = request.form.get("goitre")
        tumor = request.form.get("tumor")
        hypopituitary = request.form.get("hypopituitary")
        psych = request.form.get("psych")
        TSH = request.form.get("TSH")
        T3 = request.form.get("T3")
        TT4 = request.form.get("TT4")
        T4U = request.form.get("T4U")
        FTI = request.form.get("FTI")
        TBG = request.form.get("TBG")

        x = [[float(goitre), float(tumor), float(hypopituitary), float(psych), float(TSH), float(T3), float(TT4), float(T4U), float(FTI), float(TBG)]]

        col = ['goitre', 'tumor', 'hypopituitary', 'psych', 'TSH', 'T3', 'TT4', 'T4U', 'FTI', 'TBG']
        x = pd.DataFrame(x, columns=col)

        pred = model.predict(x)
        pred = le.inverse_transform(pred)

        return render_template("submit.html", result=str(pred))


# running flask app
if __name__ == "__main__":
        app.run(debug=True)
```

Ln 20, Col 16 100% Windows (CRLF) UTF-8

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv("/content/data.csv")

data.head()
```

```
   age sex on_thyroxine query_on_thyroxine on_antithyroid_meds sick
pregnant  \
0   29   F            f                  f                   f    f
f
1   29   F            f                  f                   f    f
f
2   41   F            f                  f                   f    f
f
3   36   F            f                  f                   f    f
f
4   32   F            f                  f                   f    f
f

  thyroid_surgery I131_treatment query_hypothyroid  ...    TT4
T4U_measured  \
0               f              f                 t  ...    NaN
f
1               f              f                 f  ...  128.0
f
2               f              f                 f  ...    NaN
f
3               f              f                 f  ...    NaN
f
4               f              f                 f  ...    NaN
f

  T4U FTI_measured FTI TBG_measured   TBG  referral_source target
patient_id
0 NaN            f NaN            f   NaN            other      -
840801013
1 NaN            f NaN            f   NaN            other      -
840801014
2 NaN            f NaN            t  11.0            other      -
840801042
3 NaN            f NaN            t  26.0            other      -
840803046
4 NaN            f NaN            t  36.0            other      S
840803047

[5 rows x 31 columns]
```

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

```
array(['-', 'S', 'F', 'AK', 'R', 'I', 'M', 'N', 'G', 'K', 'A', 'KJ',
'L',
       'MK', 'Q', 'J', 'C|I', 'O', 'LJ', 'H|K', 'D', 'GK', 'MI', 'P',
       'FK', 'B', 'GI', 'C', 'GKJ', 'OI', 'D|R', 'E'], dtype=object)
```

```
data.shape
```

```
(9172, 31)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9172 entries, 0 to 9171
Data columns (total 31 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   age                   9172 non-null   int64  
 1   sex                   8865 non-null   object 
 2   on_thyroxine          9172 non-null   object 
 3   query_on_thyroxine    9172 non-null   object 
 4   on_antithyroid_meds   9172 non-null   object 
 5   sick                  9172 non-null   object 
 6   pregnant              9172 non-null   object 
 7   thyroid_surgery       9172 non-null   object 
 8   I131_treatment        9172 non-null   object 
 9   query_hypothyroid     9172 non-null   object 
 10  query_hyperthyroid    9172 non-null   object 
 11  lithium               9172 non-null   object 
 12  goitre                9172 non-null   object 
 13  tumor                 9172 non-null   object 
 14  hypopituitary         9172 non-null   object 
 15  psych                 9172 non-null   object 
 16  TSH_measured          9172 non-null   object 
 17  TSH                   8330 non-null   float64
 18  T3_measured           9172 non-null   object 
 19  T3                    6568 non-null   float64
 20  TT4_measured          9172 non-null   object 
 21  TT4                   8730 non-null   float64
 22  T4U_measured          9172 non-null   object 
 23  T4U                   8363 non-null   float64
 24  FTI_measured          9172 non-null   object 
 25  FTI                   8370 non-null   float64
 26  TBG_measured          9172 non-null   object 
 27  TBG                   349 non-null    float64
 28  referral_source       9172 non-null   object 
 29  target                9172 non-null   object 
 30  patient_id            9172 non-null   int64  
dtypes: float64(6), int64(2), object(23)
memory usage: 2.2+ MB
```

```
data.isnull().sum()
```

```
age                        0
sex                      307
on_thyroxine               0
query_on_thyroxine         0
on_antithyroid_meds        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_measured               0
TSH                      842
T3_measured                0
T3                      2604
TT4_measured               0
TT4                      442
T4U_measured               0
T4U                      809
FTI_measured               0
FTI                      802
TBG_measured               0
TBG                     8823
referral_source            0
target                     0
patient_id                 0
dtype: int64
```

```
data.drop(['TSH_measured', 'T3_measured', 'TT4_measured',
'T4U_measured', 'FTI_measured', 'TBG_measured', 'referral_source',
'patient_id'], axis=1, inplace = True)
```

```
data.head()
```

```
   age sex on_thyroxine query_on_thyroxine on_antithyroid_meds sick
pregnant  \
0   29   F            f                  f                   f    f
f
1   29   F            f                  f                   f    f
f
2   41   F            f                  f                   f    f
f
3   36   F            f                  f                   f    f
f
4   32   F            f                  f                   f    f
f
```

```
  thyroid_surgery I131_treatment query_hypothyroid  ... tumor
hypopituitary  \
0               f              f                 t  ...     f
f
1               f              f                 f  ...     f
f
2               f              f                 f  ...     f
f
3               f              f                 f  ...     f
f
4               f              f                 f  ...     f
f

  psych  TSH   T3    TT4  T4U  FTI   TBG  target
0     f  0.3  NaN    NaN  NaN  NaN   NaN       -
1     f  1.6  1.9  128.0  NaN  NaN   NaN       -
2     f  NaN  NaN    NaN  NaN  NaN  11.0       -
3     f  NaN  NaN    NaN  NaN  NaN  26.0       -
4     f  NaN  NaN    NaN  NaN  NaN  36.0       S

[5 rows x 23 columns]
```

```
data['target']
```

```
0       -
1       -
2       -
3       -
4       S
       ..
9167    -
9168    -
9169    I
9170    -
9171    -
Name: target, Length: 9172, dtype: object
```

```
diagnoses = {'A': 'hyperthyroid conditions',
             'B': 'hyperthyroid conditions',
             'C': 'hyperthyroid conditions',
             'D': 'hyperthyroid conditions',
             'E': 'hypothyroid conditions',
             'F': 'hypothyroid conditions',
             'G': 'hypothyroid conditions',
             'H': 'hypothyroid conditions',
             'I': 'binding protein',
             'J': 'binding protein',
             '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)

data
```

```
      age sex on_thyroxine query_on_thyroxine on_antithyroid_meds sick
\
0      29   F            f                   f                    f    f

1      29   F            f                   f                    f    f

2      41   F            f                   f                    f    f

3      36   F            f                   f                    f    f

4      32   F            f                   f                    f    f

...   ...  ..          ...                 ...                  ...  ...

9167   56   M            f                   f                    f    f

9168   22   M            f                   f                    f    f

9169   69   M            f                   f                    f    f

9170   47   F            f                   f                    f    f

9171   31   M            f                   f                    f    f


     pregnant thyroid_surgery I131_treatment query_hypothyroid  ...
tumor  \
0           f               f              f                 t  ...
f
1           f               f              f                 f  ...
f
2           f               f              f                 f  ...
f
3           f               f              f                 f  ...
f
4           f               f              f                 f  ...
```

```
f
...       ...             ...            ...                ...  ...
...
9167        f               f              f                  f  ...
f
9168        f               f              f                  f  ...
f
9169        f               f              f                  f  ...
f
9170        f               f              f                  f  ...
f
9171        f               f              f                  t  ...
f

     hypopituitary psych  TSH   T3    TT4   T4U   FTI   TBG
target
0                f     f  0.3  NaN    NaN   NaN   NaN   NaN
NaN
1                f     f  1.6  1.9  128.0   NaN   NaN   NaN
NaN
2                f     f  NaN  NaN    NaN   NaN   NaN  11.0
NaN
3                f     f  NaN  NaN    NaN   NaN   NaN  26.0
NaN
4                f     f  NaN  NaN    NaN   NaN   NaN  36.0
miscellaneous
...            ...   ...  ...  ...    ...   ...   ...   ...
...
9167             f     f  NaN  NaN   64.0  0.83  77.0   NaN
NaN
9168             f     f  NaN  NaN   91.0  0.92  99.0   NaN
NaN
9169             f     f  NaN  NaN  113.0  1.27  89.0   NaN  binding
protein
9170             f     f  NaN  NaN   75.0  0.85  88.0   NaN
NaN
9171             f     f  NaN  NaN   66.0  1.02  65.0   NaN
NaN

[9172 rows x 23 columns]
```

```
data.isnull().sum()
```

```
age                       0
sex                     307
on_thyroxine              0
query_on_thyroxine        0
on_antithyroid_meds       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                     842
T3                     2604
TT4                     442
T4U                     809
FTI                     802
TBG                    8823
target                 6935
dtype: int64
```

```
data.dropna(subset=['target'], inplace=True)
```

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

```
hypothyroid conditions     593
general health             436
binding protein            376
replacement therapy        336
miscellaneous              281
hyperthyroid conditions    182
antithyroid treatment       33
Name: target, dtype: int64
```

```
data['target'].isnull().sum()
```

```
0
```

```
data.head()
```

```
    age sex on_thyroxine query_on_thyroxine on_antithyroid_meds sick
pregnant  \
4    32   F            f                   f                   f    f
f
18   63   F            t                   f                   f    t
f
32   41   M            f                   f                   f    f
f
33   71   F            t                   f                   f    f
f
39   55   F            t                   f                   f    f
f

   thyroid_surgery I131_treatment query_hypothyroid  ... tumor
hypopituitary  \
```

```
4                f              f                f  ...     f
f
18               f              f                f  ...     f
f
32               f              f                f  ...     f
f
33               f              f                f  ...     f
f
39               f              f                t  ...     f
f

   psych        TSH   T3    TT4   T4U   FTI   TBG
target
4      f        NaN  NaN    NaN   NaN   NaN  36.0
miscellaneous
18     f  68.000000  NaN   48.0  1.02  47.0   NaN  hypothyroid
conditions
32     f   0.050000  1.6   39.0  1.00  39.0   NaN
miscellaneous
33     f   0.050000  NaN  126.0  1.38  91.0   NaN         binding
protein
39     f   9.599999  2.4  136.0  1.48  92.0   NaN     replacement
therapy

[5 rows x 23 columns]
```

```
data.describe()
```

```
               age          TSH           T3          TT4          T4U
\
count  2237.000000  2087.000000  1643.000000  2140.000000  2059.000000

mean     52.792579    14.930791     1.961875   116.390495     1.013439

std      19.677450    46.204092     1.452238    60.351600     0.280222

min       1.000000     0.005000     0.050000     2.000000     0.170000

25%      36.000000     0.255000     1.000000    76.000000     0.850000

50%      56.000000     2.000000     1.700000   109.000000     0.960000

75%      69.000000     8.799999     2.500000   156.000000     1.120000

max      95.000000   530.000000    18.000000   600.000000     2.330000


               FTI         TBG
count  2060.000000   98.000000
```

```
mean    120.363369   47.717347
std      70.996728   32.398750
min       1.400000    9.299999
25%      83.000000   32.000000
50%     109.000000   36.000000
75%     157.000000   46.750000
max     881.000000  200.000000
```

```
data[data.age > 100]
```

```
Empty DataFrame
Columns: [age, sex, on_thyroxine, query_on_thyroxine,
on_antithyroid_meds, sick, pregnant, thyroid_surgery, I131_treatment,
query_hypothyroid, query_hyperthyroid, lithium, goitre, tumor,
hypopituitary, psych, TSH, T3, TT4, T4U, FTI, TBG, target]
Index: []

[0 rows x 23 columns]
```

```
data['age']=np.where((data.age > 100), np.nan, data.age)
```

```
data
```

```
       age sex on_thyroxine query_on_thyroxine on_antithyroid_meds
sick  \
4     32.0   F            f                   f                    f
f
18    63.0   F            t                   f                    f
t
32    41.0   M            f                   f                    f
f
33    71.0   F            t                   f                    f
f
39    55.0   F            t                   f                    f
f
...    ...  ..          ...                 ...                  ...  ..
.
9153  64.0   M            f                   f                    f
f
9157  60.0   M            f                   f                    t
f
9158  64.0   M            f                   f                    f
f
9162  36.0   F            f                   f                    f
f
9169  69.0   M            f                   f                    f
f

     pregnant thyroid_surgery I131_treatment query_hypothyroid  ...
tumor  \
4           f               f              f                   f  ...
```

```
f
18          f               f              f                 f  ...
f
32          f               f              f                 f  ...
f
33          f               f              f                 f  ...
f
39          f               f              f                 t  ...
f
...       ...             ...            ...               ...  ...
...
9153        f               f              f                 f  ...
f
9157        f               f              f                 f  ...
f
9158        f               f              f                 t  ...
f
9162        f               f              f                 f  ...
f
9169        f               f              f                 f  ...
f

     hypopituitary psych        TSH   T3    TT4   T4U   FTI   TBG  \
4                 f     f        NaN  NaN    NaN   NaN   NaN  36.0
18                f     f  68.000000  NaN   48.0  1.02  47.0   NaN
32                f     f   0.050000  1.6   39.0  1.00  39.0   NaN
33                f     f   0.050000  NaN  126.0  1.38  91.0   NaN
39                f     f   9.599999  2.4  136.0  1.48  92.0   NaN
...             ...   ...        ...  ...    ...   ...   ...   ...
9153              f     f   0.810000  NaN   31.0  0.55  56.0   NaN
9157              f     f   0.180000  NaN   28.0  0.87  32.0   NaN
9158              f     f        NaN  NaN   44.0  0.53  83.0   NaN
9162              f     f        NaN  NaN   84.0  1.26  67.0   NaN
9169              f     f        NaN  NaN  113.0  1.27  89.0   NaN

                     target
4              miscellaneous
18    hypothyroid conditions
32             miscellaneous
33           binding protein
39       replacement therapy
...                      ...
9153          general health
9157          general health
9158         binding protein
9162         binding protein
9169         binding protein

[2237 rows x 23 columns]
```

```
x = data.iloc[:,0:-1]
y = data.iloc[:, -1]
```

```
data.isnull().sum()
```

```
age                         0
sex                        90
on_thyroxine                0
query_on_thyroxine          0
on_antithyroid_meds         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                       150
T3                        594
TT4                        97
T4U                       178
FTI                       177
TBG                      2139
target                      0
dtype: int64
```

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

```
array(['F', 'M', nan], dtype=object)
```

```
x['sex'].replace(np.nan, 'F', inplace=True)
```

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

```
F    1701
M     536
Name: sex, dtype: int64
```

```
x.isnull().sum()
```

```
age                         0
sex                         0
on_thyroxine                0
query_on_thyroxine          0
on_antithyroid_meds         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                     150
T3                      594
TT4                      97
T4U                     178
FTI                     177
TBG                    2139
dtype: int64
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2237 entries, 4 to 9169
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   age                   2237 non-null   float64
 1   sex                   2147 non-null   object 
 2   on_thyroxine          2237 non-null   object 
 3   query_on_thyroxine    2237 non-null   object 
 4   on_antithyroid_meds   2237 non-null   object 
 5   sick                  2237 non-null   object 
 6   pregnant              2237 non-null   object 
 7   thyroid_surgery       2237 non-null   object 
 8   I131_treatment        2237 non-null   object 
 9   query_hypothyroid     2237 non-null   object 
 10  query_hyperthyroid    2237 non-null   object 
 11  lithium               2237 non-null   object 
 12  goitre                2237 non-null   object 
 13  tumor                 2237 non-null   object 
 14  hypopituitary         2237 non-null   object 
 15  psych                 2237 non-null   object 
 16  TSH                   2087 non-null   float64
 17  T3                    1643 non-null   float64
 18  TT4                   2140 non-null   float64
 19  T4U                   2059 non-null   float64
 20  FTI                   2060 non-null   float64
 21  TBG                   98 non-null     float64
 22  target                2237 non-null   object 
dtypes: float64(7), object(16)
memory usage: 419.4+ KB
```

```
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')

from sklearn.preprocessing import OrdinalEncoder, LabelEncoder

ordinal_encoder = OrdinalEncoder(dtype = 'int64')
x.iloc[:, 1:16] = ordinal_encoder.fit_transform(x.iloc[:, 1:16])
```

```
<ipython-input-30-6681d58b2586>:4: DeprecationWarning: In a future
version, `df.iloc[:, i] = newvals` will attempt to set the values
inplace instead of always setting a new array. To retain the old
behavior, use either `df[df.columns[i]] = newvals` or, if columns are
non-unique, `df.isetitem(i, newvals)`
  x.iloc[:, 1:16] = ordinal_encoder.fit_transform(x.iloc[:, 1:16])
```

```
x.head()
```

```
     age  sex  on_thyroxine  query_on_thyroxine  on_antithyroid_meds
sick  \
4   32.0    0             0                   0                    0
0
18  63.0    0             1                   0                    0
1
32  41.0    1             0                   0                    0
0
33  71.0    0             1                   0                    0
0
39  55.0    0             1                   0                    0
0

    pregnant  thyroid_surgery  I131_treatment  query_hypothyroid  ...
goitre  \
4          0                0               0                  0  ...
0
18         0                0               0                  0  ...
0
32         0                0               0                  0  ...
0
33         0                0               0                  0  ...
0
39         0                0               0                  1  ...
0

    tumor  hypopituitary  psych        TSH   T3    TT4   T4U   FTI
TBG
4       0              0      0        NaN  NaN    NaN   NaN   NaN
36.0
18      0              0      0  68.000000  NaN   48.0  1.02  47.0
```

```
NaN
32      0              0      0   0.050000  1.6   39.0  1.00  39.0
NaN
33      0              0      0   0.050000  NaN  126.0  1.38  91.0
NaN
39      0              0      0   9.599999  2.4  136.0  1.48  92.0
NaN

[5 rows x 22 columns]
```

```
x.replace(np.nan, '0', inplace=True)
x.head()
```

```
     age  sex  on_thyroxine  query_on_thyroxine  on_antithyroid_meds
sick  \
4   32.0    0             0                   0                    0
0
18  63.0    0             1                   0                    0
1
32  41.0    1             0                   0                    0
0
33  71.0    0             1                   0                    0
0
39  55.0    0             1                   0                    0
0

    pregnant  thyroid_surgery  I131_treatment  query_hypothyroid  ...
goitre  \
4          0                0               0                  0  ...
0
18         0                0               0                  0  ...
0
32         0                0               0                  0  ...
0
33         0                0               0                  0  ...
0
39         0                0               0                  1  ...
0

    tumor  hypopituitary  psych       TSH   T3    TT4   T4U   FTI
TBG
4       0              0      0         0    0      0     0     0
36.0
18      0              0      0      68.0    0   48.0  1.02  47.0
0
32      0              0      0      0.05  1.6   39.0   1.0  39.0
0
33      0              0      0      0.05    0  126.0  1.38  91.0
0
39      0              0      0  9.599999  2.4  136.0  1.48  92.0
```

```
0

[5 rows x 22 columns]
```

```
label_encoder = LabelEncoder()
y_dt = label_encoder.fit_transform(y)
```

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

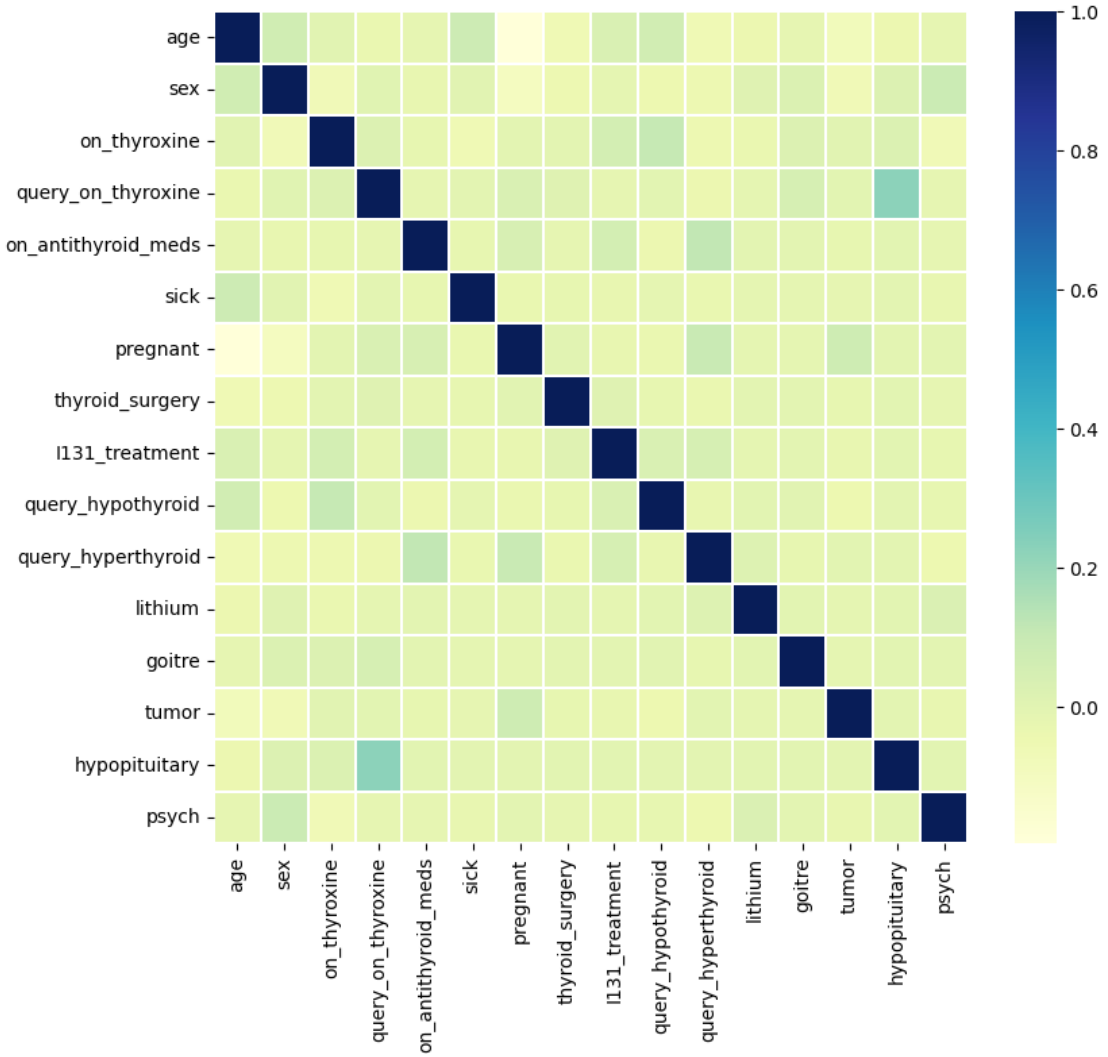
```
      target
0          5
1          4
2          5
3          1
4          6
...      ...
2232       2
2233       2
2234       1
2235       1
2236       1

[2237 rows x 1 columns]
```

```
y.value_counts(normalize=True)
```

```
target
4         0.265087
2         0.194904
1         0.168082
6         0.150201
5         0.125615
3         0.081359
0         0.014752
dtype: float64
```

```
import seaborn as sns
corrmat = x.corr()

f, ax = plt.subplots(figsize=(9, 8))
sns.heatmap(corrmat, ax = ax, cmap = "YlGnBu", linewidths = 0.1)
```

```
<ipython-input-36-64415348dfec>:2: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  corrmat = x.corr()

<Axes: >
```




```
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)

y_train.value_counts()
```

```
target
4          471
2          351
1          302
6          265
5          230
3          144
0           26
dtype: int64
```

```
from imblearn.over_sampling import SMOTE
os = SMOTE(random_state = 0, k_neighbors = 1)
```

```
x_bal, y_bal = os.fit_resample(x_train, y_train)
x_test_bal, y_test_bal = os.fit_resample(x_test, y_test)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_bal = sc.fit_transform(x_bal)
x_test_bal = sc.transform(x_test_bal)
```

```
x_bal
```

```
array([[-1.62721505, -0.44060477, -0.4238    , ..., -2.50870684,
        -1.40088079,  3.29445097],
       [-0.11561403, -0.44060477,  2.35960359, ..., -0.26259147,
         0.0720981 , -0.19494049],
       [ 1.1874903 ,  2.26960776, -0.4238    , ...,  0.17039463,
        -0.19352104, -0.19494049],
       ...,
       [ 1.395987  , -0.44060477,  2.35960359, ...,  0.43615031,
         0.06101022, -0.19494049],
       [ 0.72802783, -0.44060477,  2.35960359, ...,  0.143333  ,
         0.89086631, -0.19494049],
       [ 1.15628145, -0.44060477,  2.35960359, ...,  0.39723515,
        -0.26588659, -0.19494049]])
```

```
x_test_bal
```

```
array([[-1.5229667 , -0.44060477, -0.4238    , ...,  1.06342846,
         0.13246609, -0.19494049],
       [-0.89747663, -0.44060477, -0.4238    , ...,  1.76703086,
        -0.30218342, -0.19494049],
       [-0.9496008 ,  2.26960776, -0.4238    , ..., -0.39789962,
        -0.90586329, -0.19494049],
       ...,
       [ 1.39013447, -0.44060477,  2.35960359, ...,  0.81835453,
         0.70094189, -0.19494049],
       [ 1.33846247, -0.44060477,  2.35960359, ...,  0.81987378,
         0.67327619, -0.19494049],
       [-0.19842352, -0.44060477, -0.4238    , ...,  0.24830842,
         0.37610348, -0.19494049]])
```

```
y_bal.value_counts()
```

```
target
0         471
1         471
2         471
3         471
4         471
5         471
6         471
dtype: int64
```

```
columns = ['age', 'sex', 'on_thyroxine', 'query_on_thyroxine',
'on_antithyroid_meds', 'sick', 'pregnant', 'thyroid_surgery',
'I131_treatment', 'query_hypothyroid', 'query_hypothyroid', 'lithium',
'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH', 'T3', 'TT4',
'T4U', 'FTI', 'TBG']

x_test_bal = pd.DataFrame(x_test_bal, columns=columns)

x_bal = pd.DataFrame(x_bal, columns=columns)

x_bal
```

```
           age       sex  on_thyroxine  query_on_thyroxine  \
0    -1.627215 -0.440605     -0.423800           -0.105069
1    -0.115614 -0.440605      2.359604           -0.105069
2     1.187490  2.269608     -0.423800           -0.105069
3    -1.366594 -0.440605     -0.423800           -0.105069
4    -0.167738 -0.440605     -0.423800           -0.105069
...        ...       ...           ...                 ...
3292  0.546923 -0.440605      2.359604           -0.105069
3293  0.383062 -0.440605      2.359604           -0.105069
3294  1.395987 -0.440605      2.359604           -0.105069
3295  0.728028 -0.440605      2.359604           -0.105069
3296  1.156281 -0.440605      2.359604           -0.105069

      on_antithyroid_meds      sick  pregnant  thyroid_surgery  \
0               -0.158703 -0.141815 -0.137297        -0.239601
1               -0.158703 -0.141815 -0.137297        -0.239601
2               -0.158703 -0.141815 -0.137297        -0.239601
3               -0.158703 -0.141815 -0.137297        -0.239601
4               -0.158703 -0.141815 -0.137297        -0.239601
...                   ...       ...       ...              ...
3292            -0.158703 -0.141815 -0.137297        -0.239601
3293            -0.158703 -0.141815 -0.137297        -0.239601
3294            -0.158703 -0.141815 -0.137297        -0.239601
3295            -0.158703 -0.141815 -0.137297        -0.239601
3296            -0.158703 -0.141815 -0.137297        -0.239601

      I131_treatment  query_hypothyroid  ...    goitre     tumor  \
0          -0.162675          -0.230986  ... -0.052319 -0.137297
1          -0.162675          -0.230986  ... -0.052319 -0.137297
2          -0.162675          -0.230986  ... -0.052319 -0.137297
3          -0.162675          -0.230986  ... -0.052319  7.283487
4          -0.162675          -0.230986  ... -0.052319 -0.137297
...              ...                ...  ...       ...       ...
3292       -0.162675          -0.230986  ... -0.052319 -0.137297
3293       -0.162675          -0.230986  ... -0.052319 -0.137297
3294       -0.162675          -0.230986  ... -0.052319 -0.137297
3295       -0.162675          -0.230986  ... -0.052319 -0.137297
3296       -0.162675          -0.230986  ... -0.052319 -0.137297
```

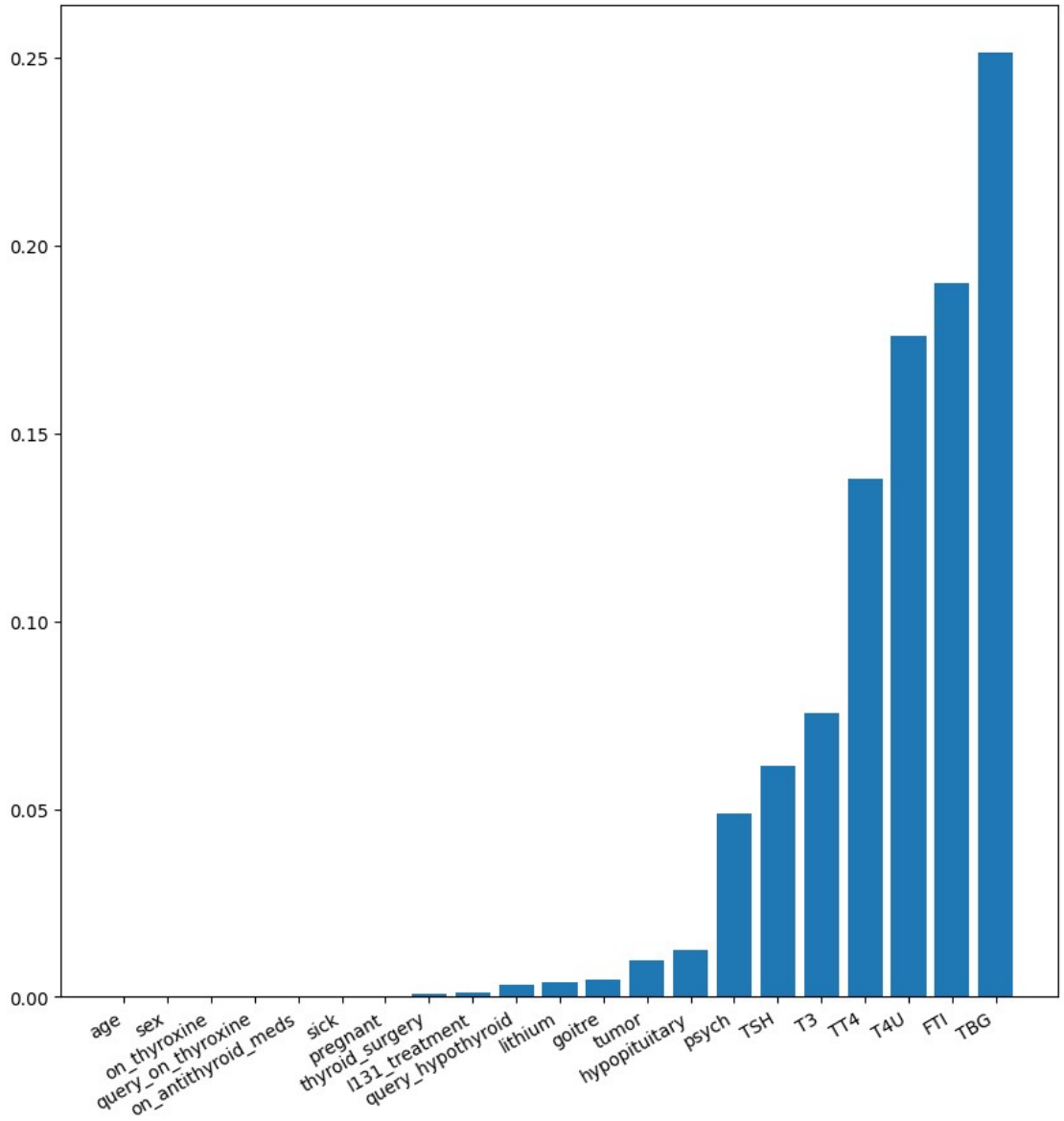
```
      hypopituitary     psych       TSH        T3       TT4       T4U
\
0         -0.024637 -0.107982 -0.315458 -1.035358 -1.704935 -2.508707

1         -0.024637 -0.107982 -0.090056  0.155233 -0.197223 -0.262591

2         -0.024637 -0.107982 -0.278907 -0.471394 -0.227079  0.170395

3         -0.024637 -0.107982 -0.284999  0.969848  0.041622  0.495134

4         -0.024637 -0.107982 -0.306321  4.541622  1.459767 -0.127283

...             ...       ...       ...       ...       ...       ...

3292      -0.024637 -0.107982 -0.114424  0.343221 -0.148122 -0.146517

3293      -0.024637 -0.107982 -0.309176 -0.856540  0.565143 -0.513902

3294      -0.024637 -0.107982 -0.095452 -0.172405  0.248906  0.436150

3295      -0.024637 -0.107982 -0.311566  0.087864  1.071643  0.143333

3296      -0.024637 -0.107982 -0.072439  0.079407 -0.200359  0.397235


           FTI       TBG
0    -1.400881  3.294451
1     0.072098 -0.194940
2    -0.193521 -0.194940
3    -0.133153 -0.194940
4     1.496783 -0.194940
...        ...       ...
3292  0.040168 -0.194940
3293  1.085434 -0.194940
3294  0.061010 -0.194940
3295  0.890866 -0.194940
3296 -0.265887 -0.194940

[3297 rows x 22 columns]
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
rfr = RandomForestClassifier().fit(x_bal, y_bal)
y_pred = rfr.predict(x_test_bal)
accuracy_score(y_test_bal, y_pred)
x_bal.shape, y_bal.shape, x_test_bal.shape, y_test_bal.shape
```

```
<ipython-input-48-0d8934587252>:3: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the
```

```
shape of y to (n_samples,), for example using ravel().
  rfr = RandomForestClassifier().fit(x_bal, y_bal)

((3297, 22), (3297, 1), (854, 22), (854, 1))

test_score = accuracy_score(y_test_bal, y_pred)
test_score

0.905152224824356

train_score = accuracy_score(y_bal, rfr.predict(x_bal))
train_score

1.0

from sklearn.inspection import permutation_importance
results = permutation_importance(rfr, x_bal, y_bal,
scoring='accuracy')

feature_importance = ['age', 'sex', 'on_thyroxine',
'query_on_thyroxine', 'on_antithyroid_meds', 'sick', 'pregnant',
'thyroid_surgery', 'I131_treatment', 'query_hypothyroid',
'query_hypothyroid', 'lithium', 'goitre', 'tumor', 'hypopituitary',
'psych', 'TSH', 'T3', 'TT4', 'T4U', 'FTI', 'TBG']
importance = results.importances_mean
importance = np.sort(importance)

for i, v in enumerate(importance):
  i = feature_importance[i]
  print('feature: {:<20} Score: {}'.format(i, v))

plt.figure(figsize=(10, 10))
plt.bar(x = feature_importance, height = importance)
plt.xticks(rotation = 30, ha = 'right')
plt.show()

feature: age                  Score: 0.0
feature: sex                  Score: 0.0
feature: on_thyroxine         Score: 0.0
feature: query_on_thyroxine   Score: 0.0
feature: on_antithyroid_meds  Score: 0.0
feature: sick                 Score: 0.00024264482863207705
feature: pregnant             Score: 0.0003033060357900963
feature: thyroid_surgery      Score: 0.0008492569002122918
feature: I131_treatment       Score: 0.0012132241431604962
feature: query_hypothyroid    Score: 0.0015165301789505925
feature: query_hypothyroid    Score: 0.0032757051865332175
feature: lithium              Score: 0.003760994843797394
feature: goitre               Score: 0.00461025174400973
feature: tumor                Score: 0.009766454352441657
feature: hypopituitary        Score: 0.012617531088868672
feature: psych                Score: 0.048892932969366074
```

```
feature: TSH                    Score: 0.06138914164391873
feature: T3                     Score: 0.07540188049742189
feature: TT4                    Score: 0.13794358507734306
feature: T4U                    Score: 0.17585683955110704
feature: FTI                    Score: 0.18999090081892628
feature: TBG                    Score: 0.251319381255687
```




```
x_bal.drop(["age", "sex", "on_thyroxine", "query_on_thyroxine",
"on_antithyroid_meds", "sick", "pregnant", "thyroid_surgery",
"I131_treatment", "query_hypothyroid", "query_hypothyroid",
"lithium"], axis = 1, inplace=True)

x_test_bal.drop(["age", "sex", "on_thyroxine", "query_on_thyroxine",
"on_antithyroid_meds", "sick", "pregnant", "thyroid_surgery",
```

```
"I131_treatment", "query_hypothyroid", "query_hypothyroid",
"lithium"], axis = 1, inplace=True)

x_bal.head()
```

```
     goitre     tumor  hypopituitary     psych       TSH        T3
TT4  \
0 -0.052319 -0.137297      -0.024637 -0.107982 -0.315458 -1.035358 -
1.704935
1 -0.052319 -0.137297      -0.024637 -0.107982 -0.090056  0.155233 -
0.197223
2 -0.052319 -0.137297      -0.024637 -0.107982 -0.278907 -0.471394 -
0.227079
3 -0.052319  7.283487      -0.024637 -0.107982 -0.284999  0.969848
0.041622
4 -0.052319 -0.137297      -0.024637 -0.107982 -0.306321  4.541622
1.459767

        T4U       FTI       TBG
0 -2.508707 -1.400881  3.294451
1 -0.262591  0.072098 -0.194940
2  0.170395 -0.193521 -0.194940
3  0.495134 -0.133153 -0.194940
4 -0.127283  1.496783 -0.194940
```

```
x_test_bal.head()
```

```
     goitre     tumor  hypopituitary     psych       TSH        T3
TT4  \
0 -0.052319 -0.137297      -0.024637 -0.107982 -0.312412  0.593872
0.788014
1 -0.052319 -0.137297      -0.024637 -0.107982 -0.314240  0.781860
0.444674
2 -0.052319 -0.137297      -0.024637 -0.107982  1.298911 -0.408731 -
1.227244
3 -0.052319 -0.137297      -0.024637 -0.107982 -0.166205 -0.471394 -
0.227079
4 -0.052319 -0.137297      -0.024637 -0.107982 -0.227125 -0.346068 -
0.301718

        T4U       FTI      TBG
0  1.063428  0.132466 -0.19494
1  1.767031 -0.302183 -0.19494
2 -0.397900 -0.905863 -0.19494
3 -0.397900  0.132466 -0.19494
4 -0.830886  0.434306 -0.19494
```

RandomForest

```
rfr1 = RandomForestClassifier()
rfr1.fit(x_bal, y_bal)
y_pred = rfr1.predict(x_test_bal)
```

```
<ipython-input-57-24f1fecb0a9c>:2: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the
shape of y to (n_samples,), for example using ravel().
  rfr1.fit(x_bal, y_bal)
```

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

```
              precision    recall  f1-score   support

           0       0.83      0.16      0.26       122
           1       0.81      0.95      0.88       122
           2       0.92      0.98      0.95       122
           3       0.76      0.84      0.80       122
           4       0.48      0.89      0.63       122
           5       0.89      0.67      0.77       122
           6       0.58      0.51      0.54       122

    accuracy                           0.71       854
   macro avg       0.75      0.71      0.69       854
weighted avg       0.75      0.71      0.69       854
```

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

```
train_score
```

```
1.0
```

XGBClassifier

```
from xgboost import XGBClassifier
xgb = XGBClassifier()
xgb.fit(x_bal, y_bal)
```

```
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, ...)
```

```
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
   macro avg       0.76      0.73      0.72       854
weighted avg       0.76      0.73      0.72       854
```

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

```
1.0
```

SVC Model

```
# model 3
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
```

```
sv = SVC()
```

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

```
/usr/local/lib/python3.9/dist-packages/sklearn/utils/
validation.py:1143: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
SVC()
```

```
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
```

```
rfr_gs = RandomForestClassifier(criterion="entropy", max_depth = 16,
n_estimators = 200)
```

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

```
<ipython-input-71-9d9e92e85fd9>:1: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the
shape of y to (n_samples,), for example using ravel().
  rfr_gs.fit(x_bal, y_bal)
```

```
RandomForestClassifier(criterion='entropy', max_depth=16,
n_estimators=200)
```

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

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

```
              precision    recall  f1-score   support

           0       0.64      0.06      0.11       122
           1       0.82      0.95      0.88       122
           2       0.93      0.99      0.96       122
           3       0.76      0.84      0.80       122
           4       0.45      0.87      0.59       122
           5       0.90      0.68      0.78       122
           6       0.57      0.52      0.54       122

    accuracy                           0.70       854
   macro avg       0.72      0.70      0.66       854
weighted avg       0.72      0.70      0.66       854
```

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

```
1.0
```

```
xgb1 = XGBClassifier(booster="gbtree", gamma=0, learning_rate=0.1,
n_estimators=500)
```

```
xgb1.fit(x_bal, y_bal)
```

```
XGBClassifier(base_score=None, booster='gbtree', 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=0, gpu_id=None, grow_policy=None,
importance_type=None,
              interaction_constraints=None, learning_rate=0.1,
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=500, n_jobs=None, num_parallel_tree=None,
              objective='multi:softprob', predictor=None, ...)
```

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

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

```
              precision    recall  f1-score   support

           0       0.83      0.32      0.46       122
           1       0.83      0.93      0.88       122
           2       0.96      1.00      0.98       122
           3       0.77      0.84      0.80       122
           4       0.51      0.80      0.62       122
           5       0.83      0.70      0.76       122
           6       0.56      0.52      0.54       122

    accuracy                           0.73       854
   macro avg       0.75      0.73      0.72       854
weighted avg       0.75      0.73      0.72       854
```

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

```
1.0
```

```
sv1 = SVC(C=1000, gamma=1, kernel='rbf')
```

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

```
/usr/local/lib/python3.9/dist-packages/sklearn/utils/
validation.py:1143: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
SVC(C=1000, gamma=1)
```

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

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

```
              precision    recall  f1-score   support

           0       0.78      0.43      0.56       122
           1       0.65      0.90      0.75       122
           2       0.92      0.90      0.91       122
           3       0.68      0.63      0.65       122
           4       0.58      0.80      0.67       122
           5       0.82      0.67      0.74       122
           6       0.47      0.44      0.46       122

    accuracy                           0.68       854
   macro avg       0.70      0.68      0.68       854
weighted avg       0.70      0.68      0.68       854
```

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

```
0.9517743403093721
```

```
import pickle
pickle.dump(xgb1, 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']
```

```
type(features)
```

```
numpy.ndarray
```

```
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])
```