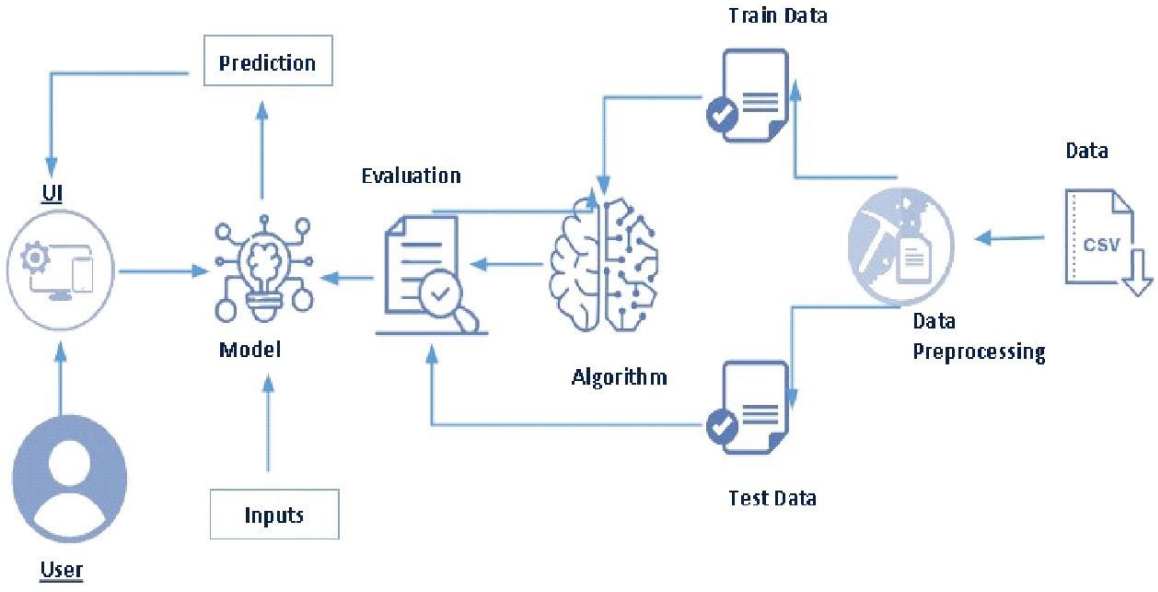
Intelligent admissions: Thyroid Disease Classification Using ML

OVERVIEW:

The Thyroid gland is a vascular gland and one of the most important organs of the human body. This gland secretes two hormones which help in controlling the metabolism of the body. The two types of Thyroid disorders are Hyperthyroidism and Hypothyroidism. When this disorder occurs in the body, they release certain types of hormones into the body which imbalances the eLearning plays a very deciding role in disease prediction. Machine Learning algorithms, SVM - support vector machine, Random Forest Classifier, XGB Classifier and ANN - Artificial Neural Networks are used to predict the patient's risk of getting thyroid disease. The web app is created to get data from users to predict the type of disease

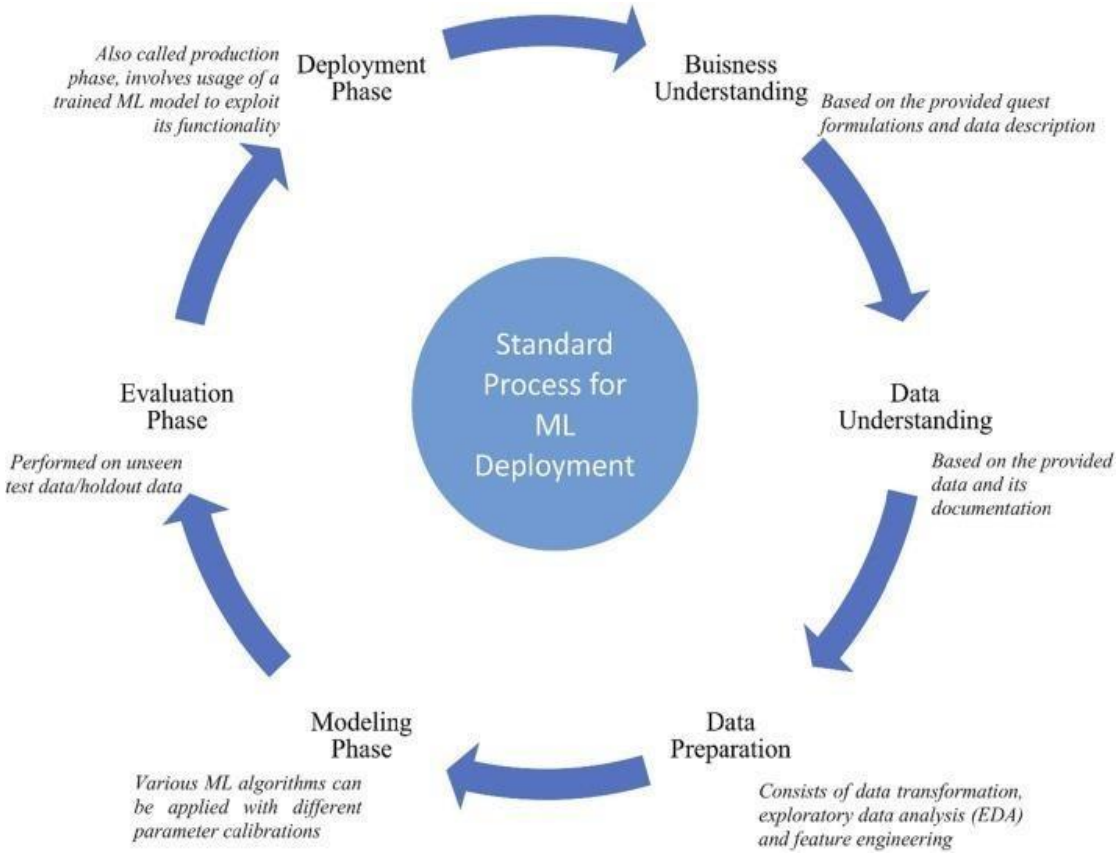
Technical Architecture:



Techinal About the project:

the vast amount of data and information difficult to deal with, especially in the health system, machine learning algorithms and data mining techniques have an important role in dealing with data. In our study, we used machine learning algorithms with thyroid disease.

The goal of this study is to categorize thyroid disease into three categories: hyperthyroidism, hypothyroidism, and normal, so we worked on this study using data from Iraqi people, some of whom have an overactive thyroid gland and others who have hypothyroidism, so we used all of the algorithms. Support vector machines, random forest, decision tree, naïve bayes, logistic regression, k-nearest neighbors, multilayer perceptron (MLP), linear discriminant analysis.



DESCRIPTION A PROJECT:

Problems with the thyroid include a variety of disorders that can result in the gland producing too little thyroid hormone (hypothyroidism) or too much (hyperthyroidism). Thyroid disorders can affect heart rate, mood, energy level, metabolism, bone health, pregnancy and many other functions.

The thyroid is a butterfly-shaped gland located in the front of the neck. It produces hormones that play a key role in regulating blood pressure, body temperature, heart rate, metabolism and the reaction of the body to other hormones.

The two main hormones produced by the thyroid are triiodothyronine, or T3, and thyroxine (T4). The gland also produces calcitonin, which helps bone cells process calcium and add it to the bones.

Project Flow:

● The user interacts with the UI to enter the input.

● Entered input is analysed by the model which is integrated.

● Once the model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

● Define Problem / Problem Understanding ○ Specify the business problem ○ Business requirements ○ Literature Survey ○ Social or Business Impact.

● Data Collection & Preparation ○ Collect the dataset ○ Data Preparation

● Exploratory Data Analysis ○ Descriptive statistical ○ Visual Analysis

● Model Building ○ Training the model in multiple algorithms ○ Testing the model

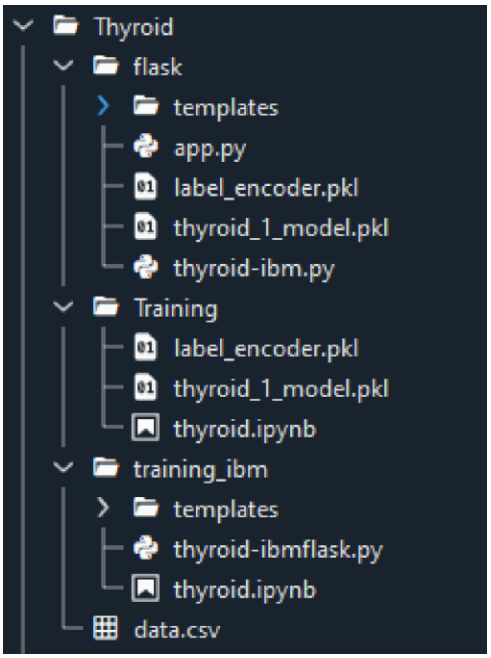
● Performance Testing & Hyperparameter Tuning ○ Testing model with multiple evaluation metrics ○ Comparing model accuracy before & after applying hyperparameter tuning

● Model Deployment ○ Save the best model ○ Integrate with Web Framework

● Project Demonstration & Documentation ○ Record explanation Video for project end to end solution ○ Project Documentation-Step by step project development procedure

Project Structure:

Create the Project folder which contains files as shown below.



● We are building a flask application which needs HTML pages stored in the templatesfolder and a python script app.py for scripting.

● thyroid_1_model.pkl is our saved model. Further, we will use this model for flaskintegration.

● Training folder contains model training files and the training_ibm folder contains IBMdeployment files

● We are building a flask application which needs HTML pages

1.2 PURPOSE:

Machine Learning has an additional benefit of processing large chunks of data that is sometimes tiresome for men to do and eventually lead to a failure in making the right decision. It is easily adaptable to new and complex data. After processing the data, it is capable of analyzing any flaws or eíioís. These also help in creating effective plans of Actions for improvement. There is a co- relation between inputs and outputs in the process of decision-making. These points are extremely useful for ventures that work mainly around risk management.

2. PROBLEM DEFINITION AMD DESIGN THINKING

Machine learning has become an increasingly popular tool in recent years, givenits ability to automatically detect patterns in data and make predictions about future

events. This can be extremely useful for making decisions in a wide range of domains, from financial trading to medical diagnoses.

2.1 EMPATHY MAP

2.2 Ideation and Brainstroming Map:

3. RESULT:

4. ADVANTAGE AND DISADVANTAGE:

Advantage:

1. Thyroid benefits — releasing and controlling thyroid hormones that control metabolism. Metabolism is a process where the food you take into your body is transformed into energy. This energy is used throughout your entire body to keep many of your body's systems working correctly.
2. It helps to regulate many body functions by constantly releasing a steady amount of thyroid hormones into the bloodstream.
3. If the body needs more energy in certain situations – for instance, if it is growing or cold, or during pregnancy – the thyroid gland produces more hormones.

Disadvantages:

1. Thyroid disorders can cause puberty and menstruation to occur abnormally early or late. In addition, abnormally high or low levels of thyroid hormone can cause very light or very heavy menstrual periods, very irregular menstrual periods, or absent menstrual periods .
2. When the thyroid makes too much thyroid hormone, your body uses energy too quickly. This is called hyperthyroidism.
3. Using energy too quickly will do more than make you tired — it can make your heart beat faster, cause you to lose weight

without trying and even make you feel nervous.

5.APPLICATION

1. The thyroid gland is a vital hormone gland: It plays a major role in the metabolism, growth and development of the human body. It helps to regulate many body functions by constantly releasing a steady amount of thyroid hormones into the bloodstream.
2. THYROXINE controls an underactive thyroid gland (hypothyroidism) and minimizes symptoms of low thyroid hormones like unknown weight gain, fatigue, sensitivity to the cold and many more.
3. Thus, it helps in replacing the body's own natural thyroid hormone essential for maintaining both mental and physical health.
4. It makes hormones that control the way the body uses energy. These hormones affect nearly every organ in your body and control many of your body's most important functions.
5. For example, they affect your breathing, heart rate, weight, digestion, and moods.

6.CONCLUSION

The thyroid gland maintains the level of metabolism in the tissues that is optimal for their normal function. Thyroid hormone stimulates the o2 consumption of most of the cells in the body, regulates lipid and carbohydrate metabolism, and is also necessary for normal growth and maturation.

Your thyroid is **a small, butterfly-shaped gland in the front of your**

neck. It makes hormones that control the way the body uses energy. These hormones affect nearly every organ in your body and control many of your body's most important functions.

that “undiagnosed hypothyroidism in pregnant women may adversely affect their fetuses; therefore screening for thyroid deficiency during pregnancy is warranted”. Several important obstetrical complications are the increased risk of spontaneous miscarriage, stillbirth and perinatal death.

7.FUTURE SCOPE

The future scope of thyroid disease is promising due to ongoing research and advancements in medical technology. One area of focus is developing more personalized treatment options for thyroid disorders. Genetic testing and precision medicine can help identify the underlying causes of thyroid disease and tailor treatment to the individual patient.

Another area of research is the use of stem cells to regenerate damaged thyroid tissue. This approach has shown promise in animal studies and may eventually be applied to human patients.

In addition, there is growing interest in the role of the gut microbiome in thyroid health. Research has suggested that imbalances in gut bacteria may contribute to the development of thyroid disorders, and that restoring a healthy microbiome may help improve thyroid function.

8.APPENDIX

Source Code:

```
Import pandas as pd
Import numpy as np
Import matplotlib. pyplot as plt
Import tensorflow
```

From tensorflow. keras. models import sequential

From tensorflow. Keras. layers import Layer, Dense, Dropout

data = pd.read_csv(“data.csv”)

```
data = pd.read_csv("data.csv")
```

```
data.head()
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	query_hypothyroid	...	TT4	T4U_measu
0	29	F	f	f	f	f	f	f	f	t	...	NaN	
1	29	F	f	f	f	f	f	f	f	f	...	128.0	
2	41	F	f	f	f	f	f	f	f	f	...	NaN	
3	36	F	f	f	f	f	f	f	f	f	...	NaN	
4	32	F	f	f	f	f	f	f	f	f	...	NaN	

5 rows × 31 columns

```
data.shape
```

```
(9172, 31)
```

data.info()

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2237 entries, 4 to 9169
Data columns (total 22 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   age                  2237 non-null   float64
 1   sex                  2237 non-null   int64
 2   on_thyroxine         2237 non-null   int64
 3   query_on_thyroxine   2237 non-null   int64
 4   on_antithyroid_meds  2237 non-null   int64
 5   sick                 2237 non-null   int64
 6   pregnant             2237 non-null   int64
 7   thyroid_surgery      2237 non-null   int64
 8   I131_treatment       2237 non-null   int64
 9   query_hypothyroid    2237 non-null   int64
 10  query_hyperthyroid   2237 non-null   int64
 11  lithium              2237 non-null   int64
 12  goitre               2237 non-null   int64
 13  tumor                2237 non-null   int64
 14  hypopituitary        2237 non-null   int64
 15  psych                2237 non-null   int64
 16  TSH                  2237 non-null   object
 17  T3                   2237 non-null   object
 18  TT4                  2237 non-null   object
 19  T4U                  2237 non-null   object
 20  FTI                  2237 non-null   object
 21  TBG                  2237 non-null   object
dtypes: float64(1), int64(15), object(6)
memory usage: 402.0+ KB
```

data.isnull().sum()

```
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.dropna(subset=[‘target’],inplace=True)

```
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
```

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

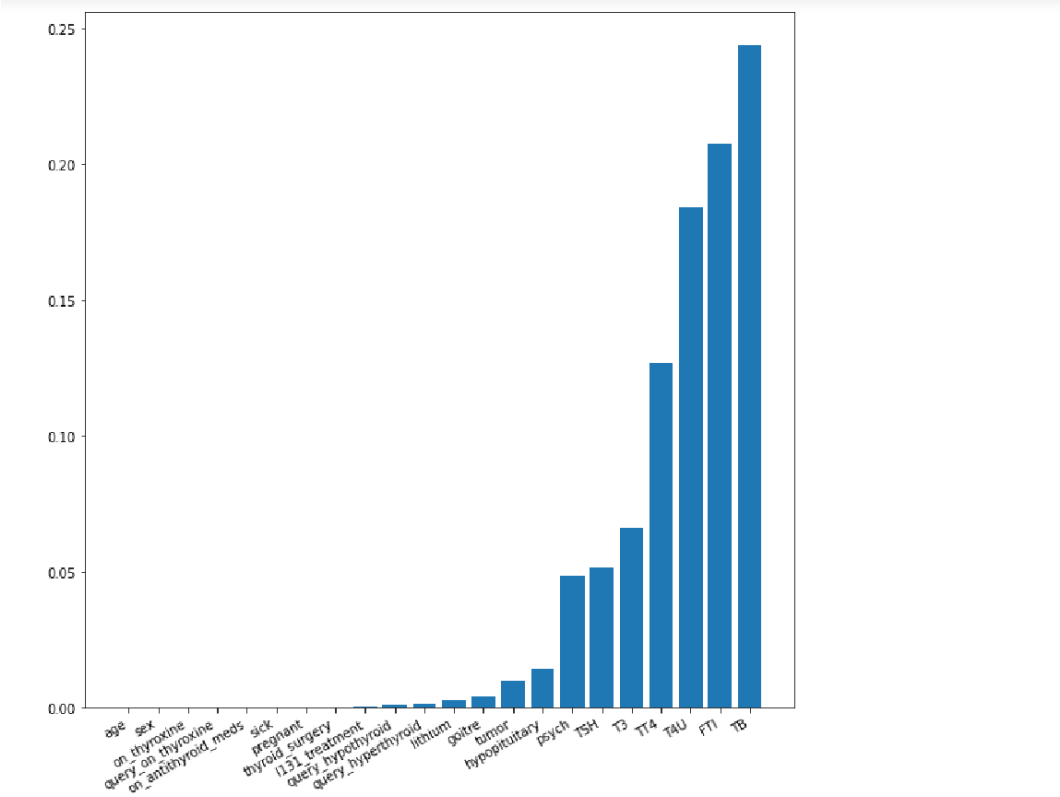
```
x
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	query_hypothyroid	...	goitre	tumor
4	32.0	0	0	0	0	0	0	0	0	0	...	0	0
18	63.0	0	1	0	0	1	0	0	0	0	...	0	0
32	41.0	1	0	0	0	0	0	0	0	0	...	0	0
33	71.0	0	1	0	0	0	0	0	0	0	...	0	0
39	55.0	0	1	0	0	0	0	0	0	1	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9153	64.0	1	0	0	0	0	0	0	0	0	...	0	0
9157	60.0	1	0	0	1	0	0	0	0	0	...	0	0
9158	64.0	1	0	0	0	0	0	0	0	1	...	0	0
9162	36.0	0	0	0	0	0	0	0	0	0	...	0	0
9169	69.0	1	0	0	0	0	0	0	0	0	...	0	0

2237 rows × 22 columns

visual analysis

Univariate



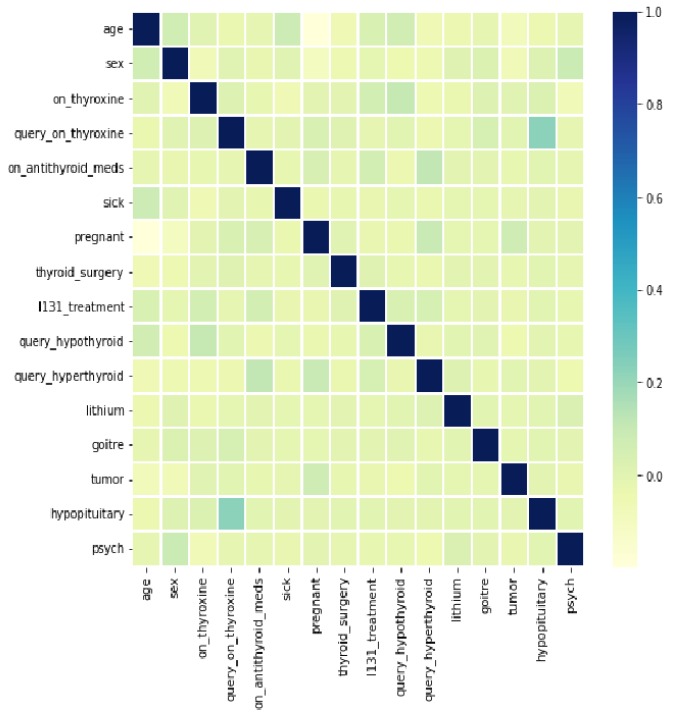
analysis

Visual analysis

```
#checking correlation using Heatmap
import seaborn as sns
corrmat = x.corr()

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

```
<AxesSubplot:>
```



Activate Windows

Scalling the Data

```
from sklearn.ensemble import RandomForestClassifier
rfr1 = RandomForestClassifier().fit(x_os,y_os.values.ravel())
y_pred = rfr1.predict(x_test_os)

rfr1 = RandomForestClassifier()
```

```
rfr1.fit(x_os, y_os.values.ravel())
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
y_pred = rfr1.predict(x_test_os)
```

Splitting data into x and y

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

```
x
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	query_hypothyroid	...	goitre	tumor
4	32.0	F	f	f	f	f	f	f	f	f	...	f	f
18	63.0	F	t	f	f	t	f	f	f	f	...	f	f
32	41.0	M	f	f	f	f	f	f	f	f	...	f	f
33	71.0	F	t	f	f	f	f	f	f	f	...	f	f
39	55.0	F	t	f	f	f	f	f	f	t	...	f	f
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9153	64.0	M	f	f	f	f	f	f	f	f	...	f	f
9157	60.0	M	f	f	t	f	f	f	f	f	...	f	f
9158	64.0	M	f	f	f	f	f	f	f	t	...	f	f
9162	36.0	F	f	f	f	f	f	f	f	f	...	f	f
9169	69.0	M	f	f	f	f	f	f	f	f	...	f	f

2237 rows × 22 columns

```
y
```

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

2237 rows × 1 columns

```
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)
```

```
from imblearn.over_sampling import SMOTE
y_train.value_counts()
```

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

ANN model

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Layer, Dense, Dropout
```

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

sv= SVC()
```

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

```
C:\Users\SmartBridge-PC\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: 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

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
```

In [68]:
```
model = Sequential()
```

In [69]:
```
model.add(Dense(units = 128, activation='relu', input_shape=(10,)))
```

In [70]:
```
model.add(Dense(units = 128, activation='relu', kernel_initializer='random_uniform'))
model.add(Dropout(0.2))
model.add(Dense(units = 256, activation='relu', kernel_initializer='random_uniform'))
model.add(Dropout(0.2))
model.add(Dense(units = 128, activation='relu', kernel_initializer='random_uniform'))
```

In [71]:
```
model.add(Dense(units = 1, activation='sigmoid'))
```

In [72]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               1408

 dense_1 (Dense)             (None, 128)               16512

 dropout (Dropout)           (None, 128)               0

 dense_2 (Dense)             (None, 256)               33024

 dropout_1 (Dropout)         (None, 256)               0

 dense_3 (Dense)             (None, 128)               32896

 dense_4 (Dense)             (None, 1)                 129

=================================================================
Total params: 83,969
Trainable params: 83,969
Non-trainable params: 0
_________________________________________________________________
```

Testing the model

testing the models

In [115]:
```
rfr1.predict([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]])
```

```
C:\Users\Mahidhar reddy\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but
RandomForestClassifier was fitted with feature names
  warnings.warn(
```

Out[115]:
```
array([4])
```

In [130]:
```
sv.predict([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]])
```

```
C:\Users\Mahidhar reddy\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but
SVC was fitted with feature names
  warnings.warn(
```

Out[130]:
```
array([1])
```

In [143]:
```
col = ['goitre', 'tumor', 'hypopituitary', 'psych', 'TSH', 'T3', 'TT4', 'T4U', 'FTI', 'TBG']
da = [[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]]
da1 = pd.DataFrame(data = da, columns=col)
xgb1.predict(da1)
```

Out[143]:
```
array([4], dtype=int64)
```

In [140]:
```
model.predict([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]])
```

```
1/1 [==============================] - 0s 238ms/step
```

Out[140]:
```
array([[1.]], dtype=float32)
```

Model Deployment::Save the bestmodel:

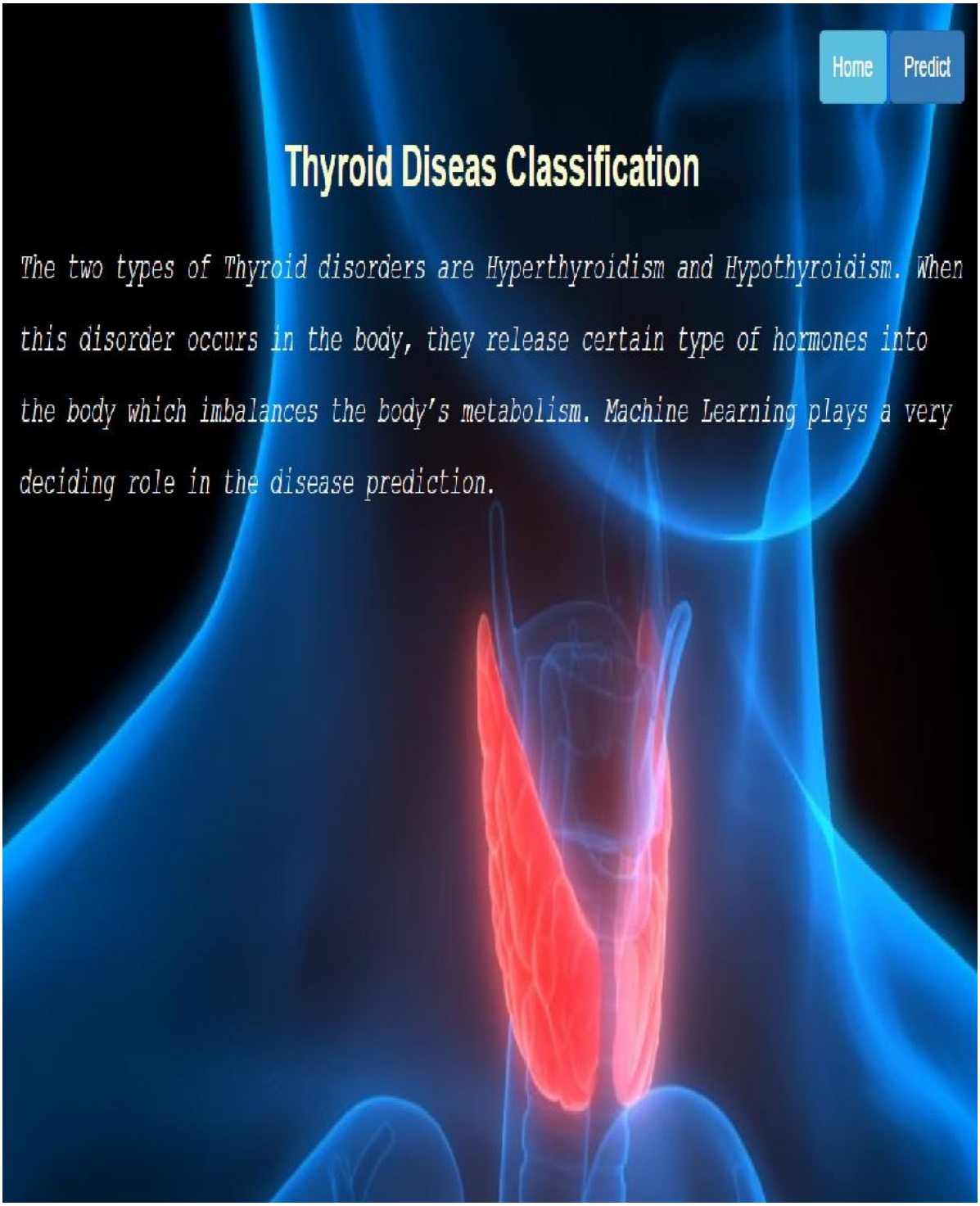
```
# saving the model
import pickle
pickle.dump(sv1,open('thyroid_1_model.pkl','wb'))
```

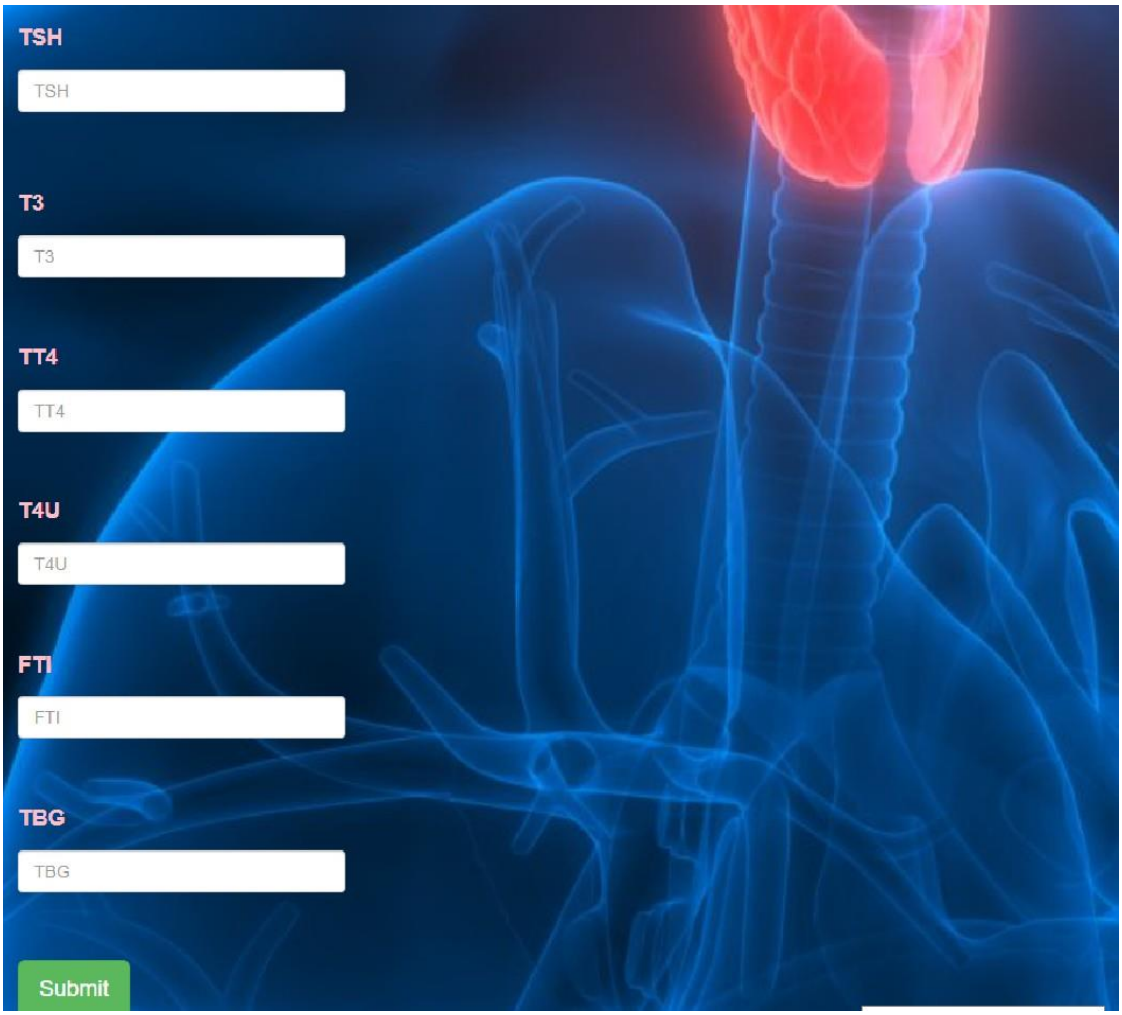
```
features = np.array([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]])
print(label_encoder.inverse_transform(xgb1.predict(features)))
```

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

```
import pickle
pickle.dump(sv1,open('thyroid_1_model.pkl','wb'))
```

Intergate with Web Framewok




TSH
TSH
T3
T3
TT4
TT4
T4U
T4U
FTI
FTI
TBG
TBG
Submit