GOVERNMENT OF TAMILNADU

Naan Muthalvan - Project-Based Experiential Learning

THYROID DISEASE CLASSIFICATION USING MACHINE LEARNING

Submitted by

R.AFIYASAHANA – (20626ER039)

S.BANU– (20626ER040)

A.ISHWARYA– (20626ER045)

S. PRIYA – (20626ER060)

Under the guidance of

Mrs.K.P.KAVITHA, MCA.,M.Phil.,(Ph.D)

Department of Computer Science

GOVERNMENT ARTS COLLEGE FOR WOMEN , NILAKOTTAI

B(Affiliated To Mother Teresa Women's University, Kodaikanal)
Reaccredited with "C" Grade by NAAC

NILAKOTTAI - 624208

APRIL - 2023

GOVERNMENT ARTS COLLEGE FOR WOMEN, NILAKOTTAI

(Affiliated to Mother Teresa Womens University, Kodaikanal)

Reaccredited with 'C' Grade by NAAC

Nilakottai - 624208

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of the project entitled **THYROID DISEASE CLASSIFICATION USING MACHINE LANGUAGE** done by **MS. R.AFIYASAHANA –(20626ER039), MS.S.BANU-(20626ER040), MS.A.ISHWARYA-(20626ER045), MS.S.PRIYA(20626ER060)** This is submitted in partial fulfillment for the award of the degree of **Bachelor of Science in Computer Science in GOVERNMENT ARTS COLLEGE FOR WOMEN, NILAKOTTAI** during the period of December 2022 to April 2023.

Project Mentor(s) **Head of the Department**

Submitted for viva-voce Examination held on __________________

INTERNAL EXAMINER **EXTERNALEXAMINER**

Thyroid disease classification using Machine Language

1. Introduction

The thyroid gland contains two main types of cells: follicular cells that produce thyroxin and C cells that produce calcitonin. A thin fibrous capsule with blood vessels, lymphatics, and nerves, most prominent at the poles, encloses the thyroid. The histologic appearance of thyroid follicles and colloid varies greatly as a reflection of secretory activity. Variation in follicle size is common, with the larger follicles tending to be at the periphery of the gland (Figure 1). Follicular epithelial lining ranges from flattened to cuboidal (Figure 2, Figure 3, Figure 4, and Figure 5). Active follicles are typically lined by more cuboidal epithelium (Figure 4 and Figure 5) and may have resorption vacuoles at the interface of the epithelia and the colloid (Figure 3). As mice age, their follicles become less active, more distended, and lined by flattened epithelium. Tinctorial variations in colloid are commonly seen.

1.1 Overview

The thyroid is a small gland, measuring about 2 inches (5 centimeters) across, that lies just under the skin below the Adam's apple in the neck. The two halves (lobes) of the gland are connected in the middle (called the isthmus), giving the thyroid gland the shape of a bow tie. Normally, the thyroid gland cannot be seen and can barely be felt. If it becomes enlarged, doctors can feel it easily, and a prominent bulge (goiter) may appear below or to the sides of the Adam's apple.

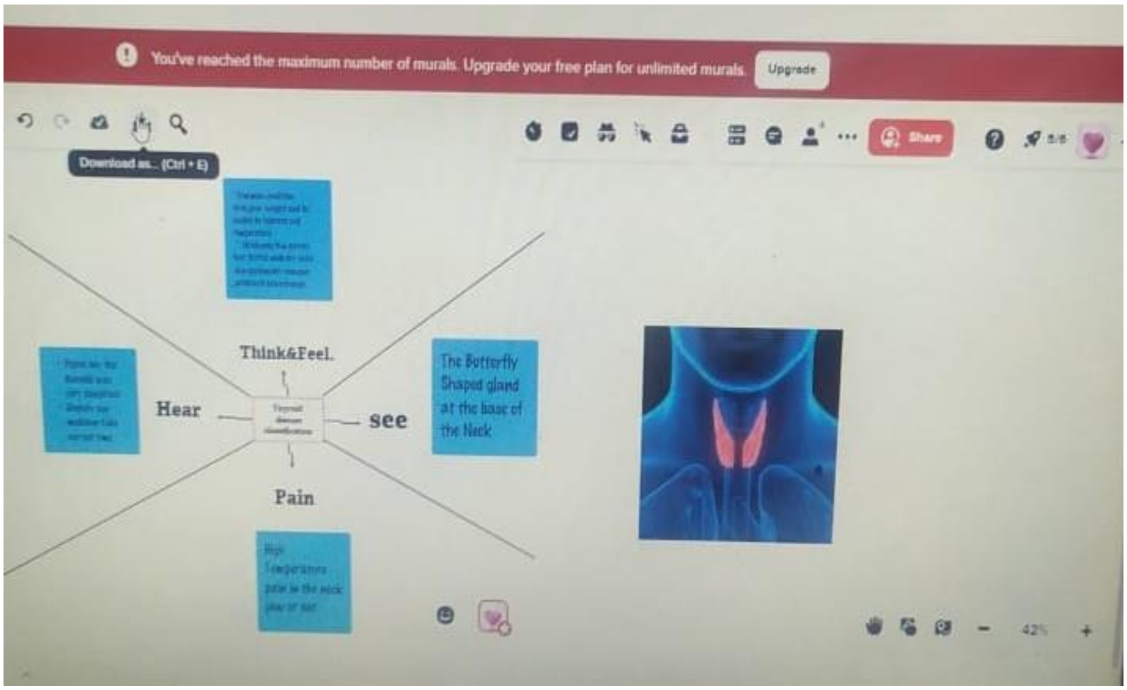
2.1 Purpose

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

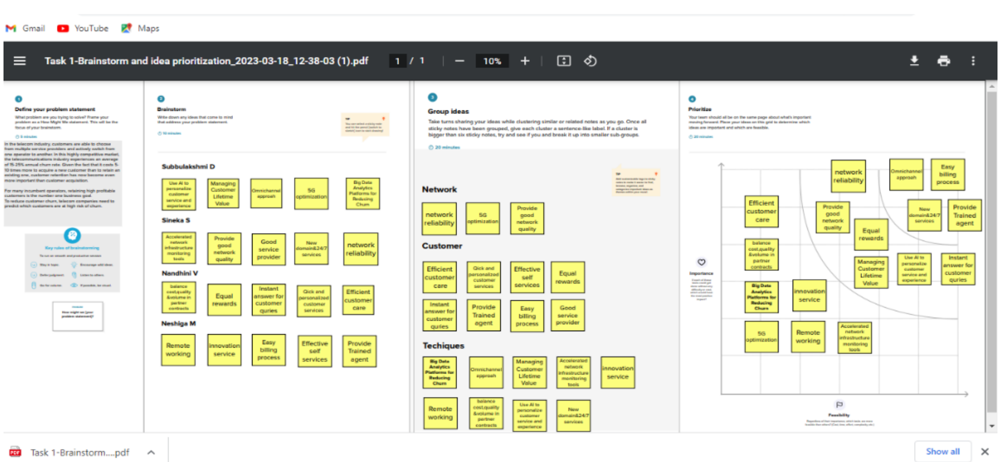
The thyroid weighs between 20 and 60 grams on average. It is surrounded by two fibrous capsules. The outer capsule is connected to the voice box muscles and many important vessels and nerves.

2. Problem definition & Design Thinking

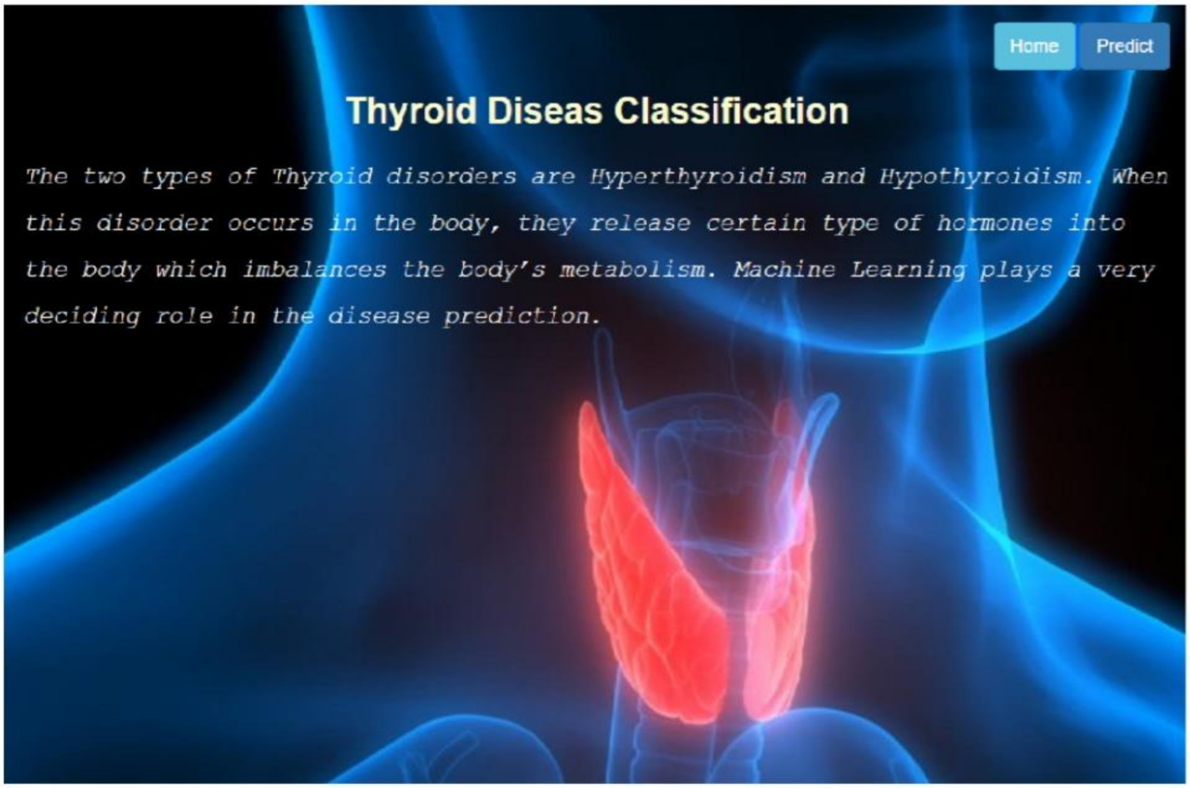
2.1Empathy map



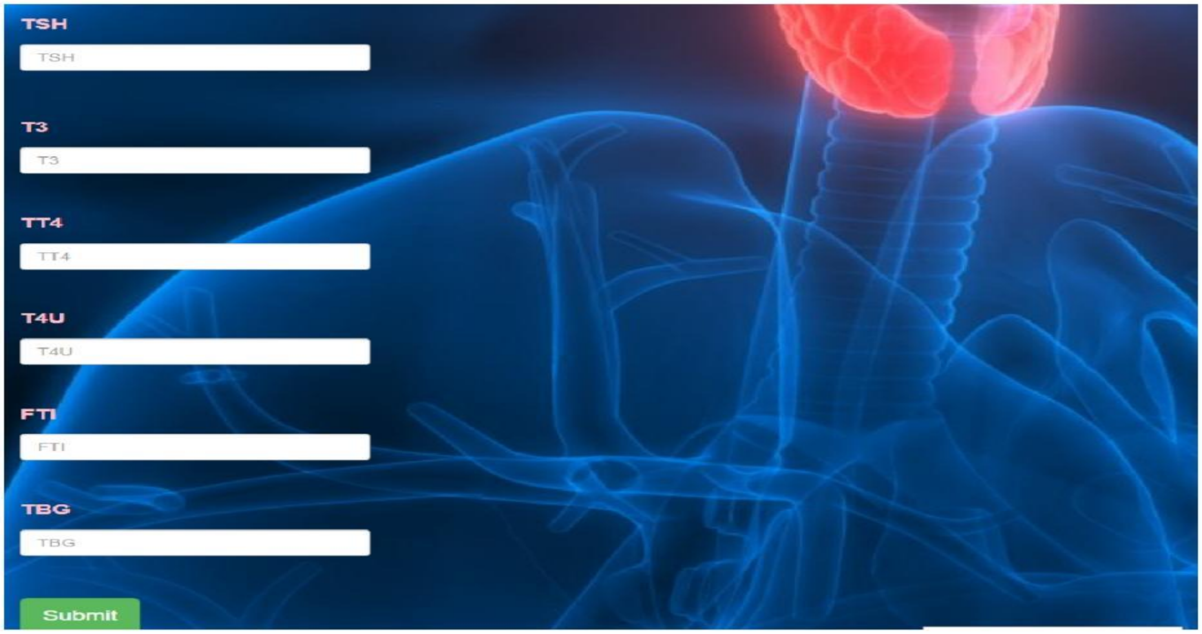
2.2 Ideation & Brainstorming Map

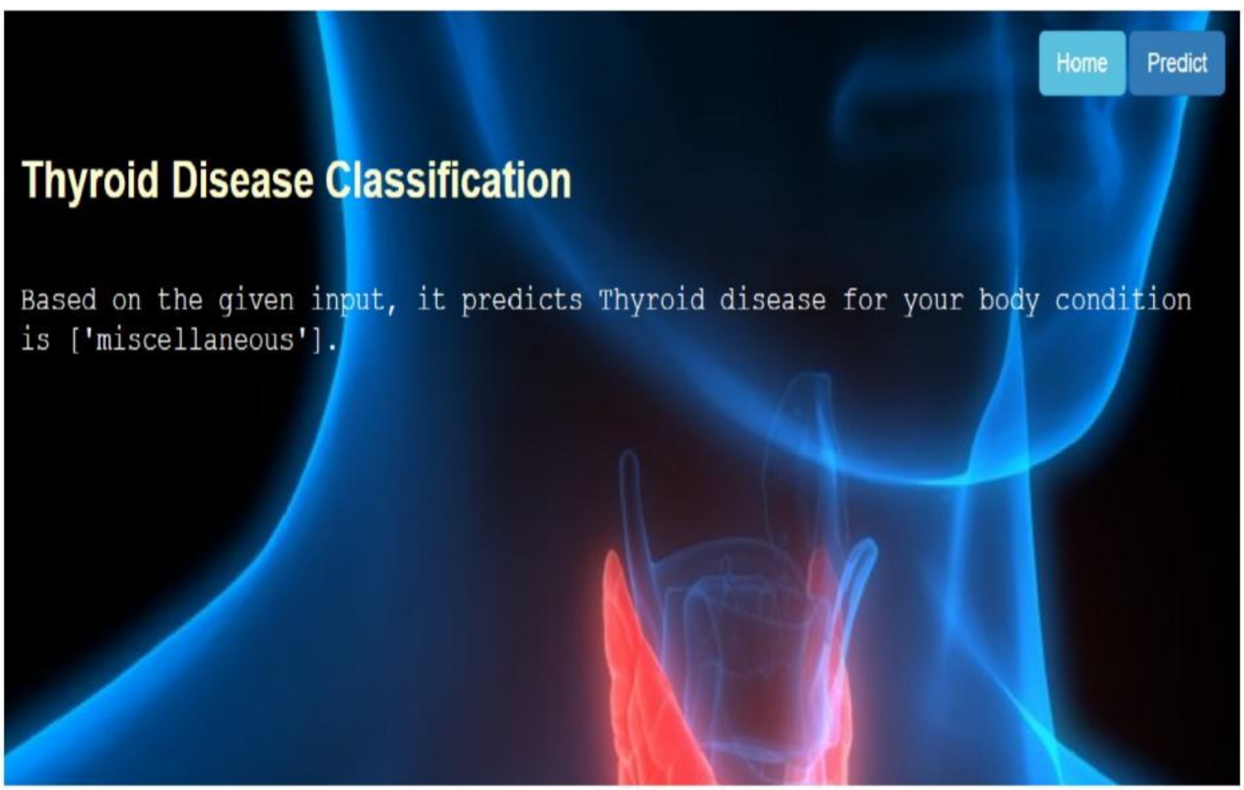


2. RESULT


Home
Predict
Thyroid Diseas Classification
The two types of Thyroid disorders are Hyperthyroidism and Hypothyroidism. When this disorder occurs in the body, they release certain type of hormones into the body which imbalances the body's metabolism. Machine Learning plays a very deciding role in the disease prediction.


Home
Predict
Thyroid Disease Classification
goitre
Male
tumor
Male
hypopituitary
Male
psych
Male
TSH
TSH


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


Home
Predict
Thyroid Disease Classification
Based on the given input, it predicts Thyroid disease for your body condition is ['miscellaneous'].

3.ADVANTAGES & DISADVANTAGES

ADVANTAGES

1. It helps to regulate many body functions by constantly releasing a steady amount of thyroid hormones into the bloodstream.

2. If the body needs more energy in certain situations – for instance, if it is growing or cold, or during pregnancy.

3. Surgery is excellent therapy for hyperthyroidism, with no mortality,and few complications or recurrences.

DISADVANTAGES

1.When the thyroid makes too much thyroid hormone, your body uses energy too quickly.

2. It can make your heart beat faster, cause you to lose weight without trying and even make you feel nervous.

3. Thyroid disorders can cause puberty and menstruation to occur abnormally early or late.

4. 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 .

5. APPLICATIONS

- **Studies with laboratory animals have shown that pertechnetate ion is selectively concentrated in the thyroid gland, salivary glands and stomach, and selectively excluded from the cerebrospinal fluid almost to the same extent as inorganic iodide.**
- **Since the radionuclide technetium-99m gives extremely low energy dissipation in tissue, human studies were undertaken to evaluate pertechnetate-99m as a clinical tracer material.**
- **After intravenous administration, approximately 30% of the technetium-99m is excreted in the urine over the first 24 hr. The pertechnetate remaining in the body is evidently slowly metabolized, since the urinary excretion decreases markedly while fecal excretion increases progressively to a total of about 20% of the injected dose at 72 hr.**

6.CONCLUSION

The incidence of thyroid cancer in multinodular goiter without any previous suspicion of malignancy was found to be 8.2%. Subtotal thyroidectomy resulted in a significantly higher rate of completion thyroidectomy for incidentally diagnosed thyroid cancer compared with total or near-total thyroidectomy. No permanent complications occurred, and the extent of surgical resection had no significant effect on the rate of temporary complications. We recommend total or near-total thyroidectomy in multinodular goiter to eliminate the need for completion thyroidectomy in case of a final diagnosis of thyroid cancer.

7.FUTURE SCOPE

- **Thyroid disease prediction has emerged as an important task recently. Despite existing approaches for its diagnosis, often the target is binary classification, the used datasets are small-sized and results are not validated either. Predominantly, existing approaches focus on model optimization and the feature engineering part is less investigated.**
- **One of the most used treatments is sodium levothyroxine, also known as LT4, a synthetic thyroid hormone used in the treatment of thyroid disorders and diseases**
- **The use of machine learning techniques can effectively support endocrinologists while monitoring patients, exploiting hormonal parameters related to the thyroid and other clinical data concerning the patient, to predict if the patient's treatment needs to be increased, decreased, or remain unchanged.**

8. APPENDIX

A. 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("/content/thyroidDF.csv")
data.head()

data.shape
data.isnull().sum()


diagnoses={'A': 'hyperthyroid coditions',
'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': 'antityroid treatment',
'R': 'miscellaneous',
'S': 'miscellaneous',
'T': 'miscellaneous'}
data['target']=data['target'].map(diagnoses)


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

data[data.age>100]
```

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

x
x['sex'].unique()

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

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

x['sex'].value_counts()


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.info()

from sklearn.preprocessing import OrdinalEncoder,LabelEncoder
Ordinal_encoder = OrdinalEncoder(dtype='int64')
x.iloc[:,1:16]= Ordinal_encoder.fit_transform(x.iloc[:,1:16])

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

x

label_encoder=LabelEncoder()

y_dt=label_encoder.fit_transform(y)

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

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


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()

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

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

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

x_bal

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',]
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()

x.head()

x_bal.drop(['age','sex','on_thyroxine','query_on_thyroxine','on_antithyroid_meds','sick','pregnant','thyroid_surgery'])

x_test_bal.drop(['age','sex','on_thyroxine','query_on_thyroxine','on_antithyroid_meds','sick','pregnant','thyroid_surgery'])
```

```
x_bal.head()

data.info()

import seaborn as sns
corrmat=x.corr()
f,ax=plt.subplots(figsize=(9,8))
sns.heatmap(corrmat,ax=ax,cmap="YIGnBu",linewidths=0.1)

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())

y_pred=rfr1.predict(x_test_os)

y_pred=rfr1.predict(x_test_os)

from sklearn.metrics import classification_report
print(classification_report(y_test_os,y_pred))

train_score=accuracy_score(y_os,rfr1.predict(x_os))
train_score
from xgboost import XGBClassifier
xgb1 = XGBClassifier()
xgb1.fit(x_os,y_os)


_pred=xgb1.predict(x_test_os)

print(classification_report(y_test_os,y_pred))

accuracy_score(y_test_os,y_pred)

from sklearn.svm import svc
from sklearn.metrics import accuracy_score,classification_report
sv=SVC()

sv.fit(x_bal,y_bal)

y_pred=sv.predict(x_test_bal)
```

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

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


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'))

model.add(Dense(units=1,activation='sigmoid'))

model.summary()

model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])


model.fit(x_bal,y_bal,validation_data=[x_test_bal,y_test_bal],epochs=15)

rfr1.predict([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]])

sv.predict([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]]

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)

model.predict([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]])

print(classification_report(y_test_bal,y_pred))

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

train_score

y_pred=xgb.predict(x_test_bal)

print(classification_report(y_test_bal,y_pred))

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

```
y_pred=sv.predict(x_test_bal)

print(classification_report(y_test_bal,y_pred))

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

y_pred=model.predict(x_test_bal)

print(classification_report(y_test_bal,y_pred))

accuracy_score(y_test_bal,y_pred)


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

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


random.svc.fit(x_bal,y_bal)

random_svc.best_params_

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

sv1.fit(x_bal,y_bal)

y_pred=sv1.predict(x_test_bal)

print(classification_report(y_test_bal,y_pred))

train_score=accuracy_score(y_bal,sv1.predict(x_bal))
trsin_score
import pickle
pickle.dump(sv1,open('thyroid_1_model.pk1','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)))
```

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

data['target'].unique()

y['target'].unique()

import pickle
pickle.dump(sv1,open('thyroid_1_model.pk1','wb'))

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

mode1=pickle.load(open(r"C:\Users\SmartBridge-
PC\Downloads\Thyroid\thyroid1_model.pkl",'rb'))
le=pickle.load(open("label_encoder.pkl",'rb'))
app=Flask(_name_)

@app.route("/")
 def about():
     return render_template('home.html')

@app.route("/pred",methods=['POST','GET'])
def predict():
  x=[[float(x)for x in reguest.from.value()]]
  print(x)
  col=['goitre','tumor','hypopituitary','psych','TSH','T3','TT4','T4U','FTI','TBG']
  x=pd.DataFrame(x,columns=col)
  print(x)
  pred=model.predict(x)
  pred=le.inverse_transform(pred)
  print(pred[0])
  return render_template('submit.html',prediction_text=str(pred))


if__name__=="__main__":
  app.run(debug=False)


 runfile('C:/Users/SmartBridge-
PC/Downloads/Thyroid/app.py',wdir='C:/users/SmartBridge-
PC/Downloads/Thyroid')
```