# INTRODUCTION:

Overview:

Thyroid disease is a general term for a medical condition that keeps your thyroid from making the right amount of hormones. Your thyroid typically makes hormones that keep your body functioning normally. When the thyroid makes too much thyroid hormone, your body uses energy too quickly. This is called hyperthyroidism. 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. On the flip-side of this, your thyroid can make too little thyroid hormone. This is called hypothyroidism. When you have too little thyroid hormone in your body, it can make you feel tired, you might gain weight and you may even be unable to tolerate cold temperatures.

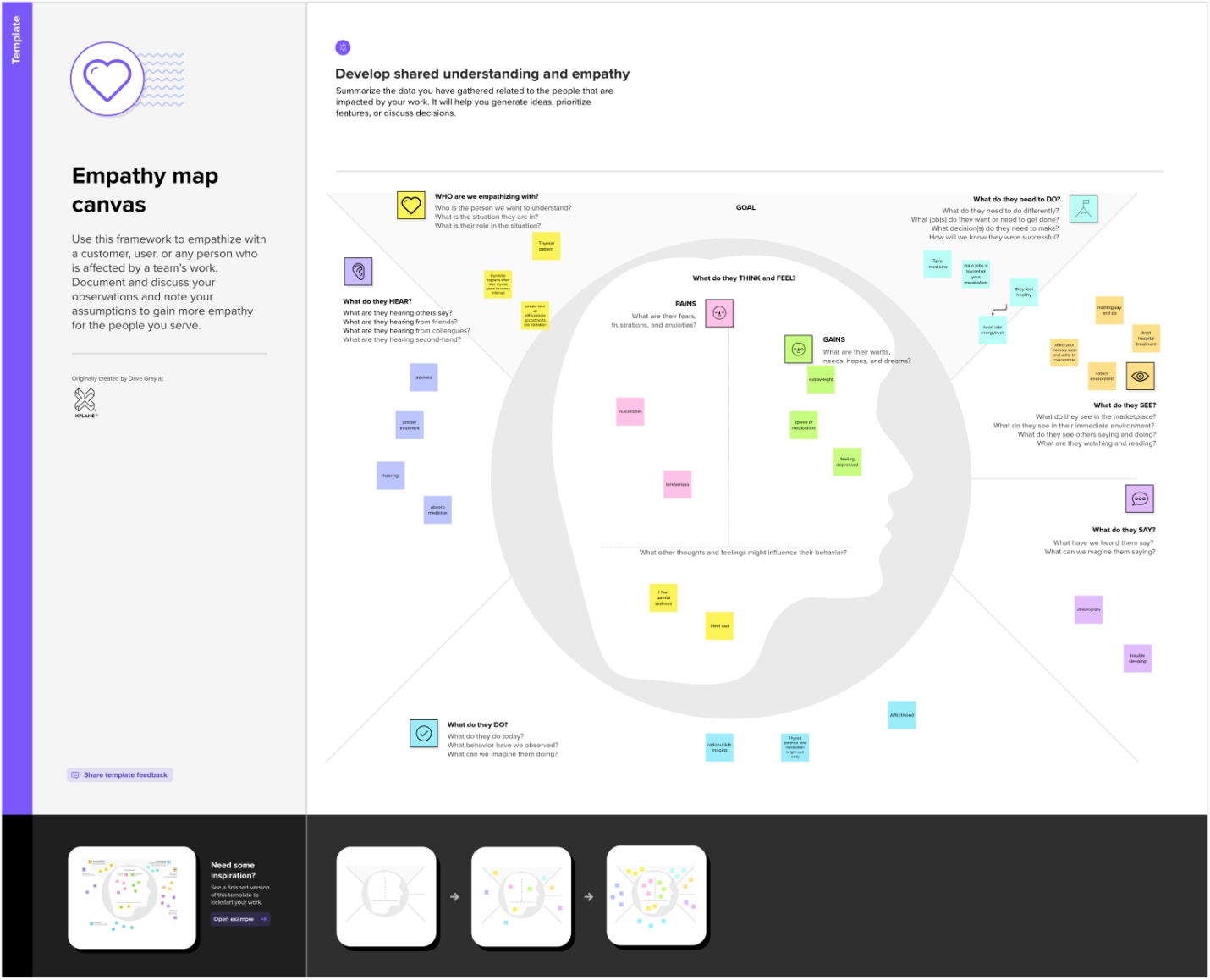
The two main types of thyroid disease are hypothyroidism and hyperthyroidism. Both conditions can be caused by other diseases that impact the way the thyroid gland works. Conditions that can cause hypothyroidism include: Thyroids: This condition is an inflammation (swelling) of the thyroid gland.

Purpose:

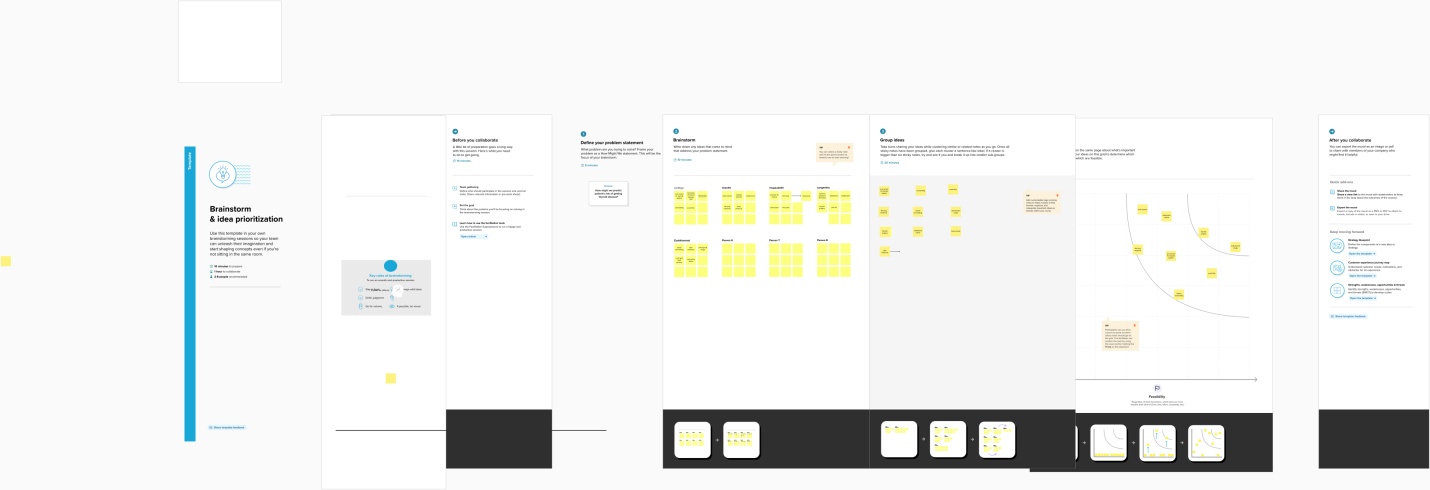
It makes hormones that control the way the body uses energy. These hormones affect nearly every organ in your body and control many of your body's most important functions. For example, they affect your breathing, heart rate, weight, digestion, and moods.

PROBLEM DEFINITION AND DESIGN THINKING:

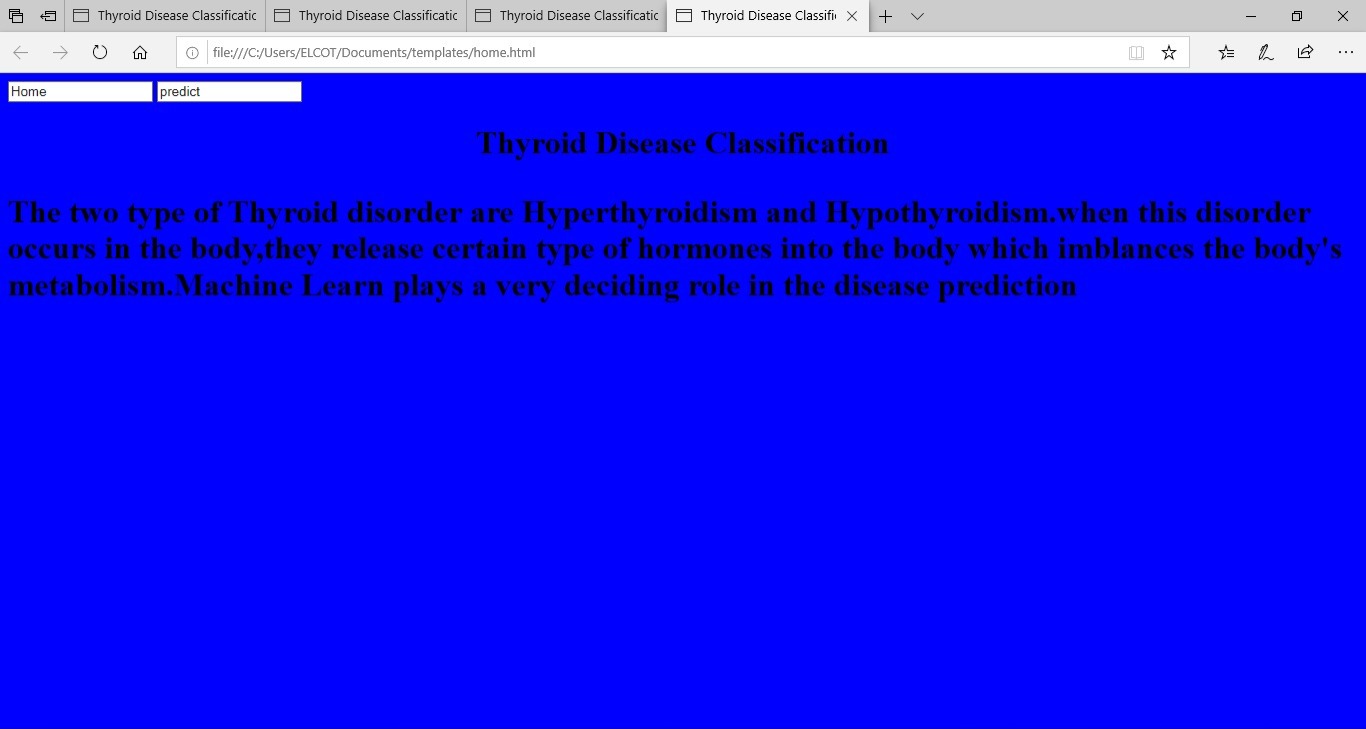
Empathy map:



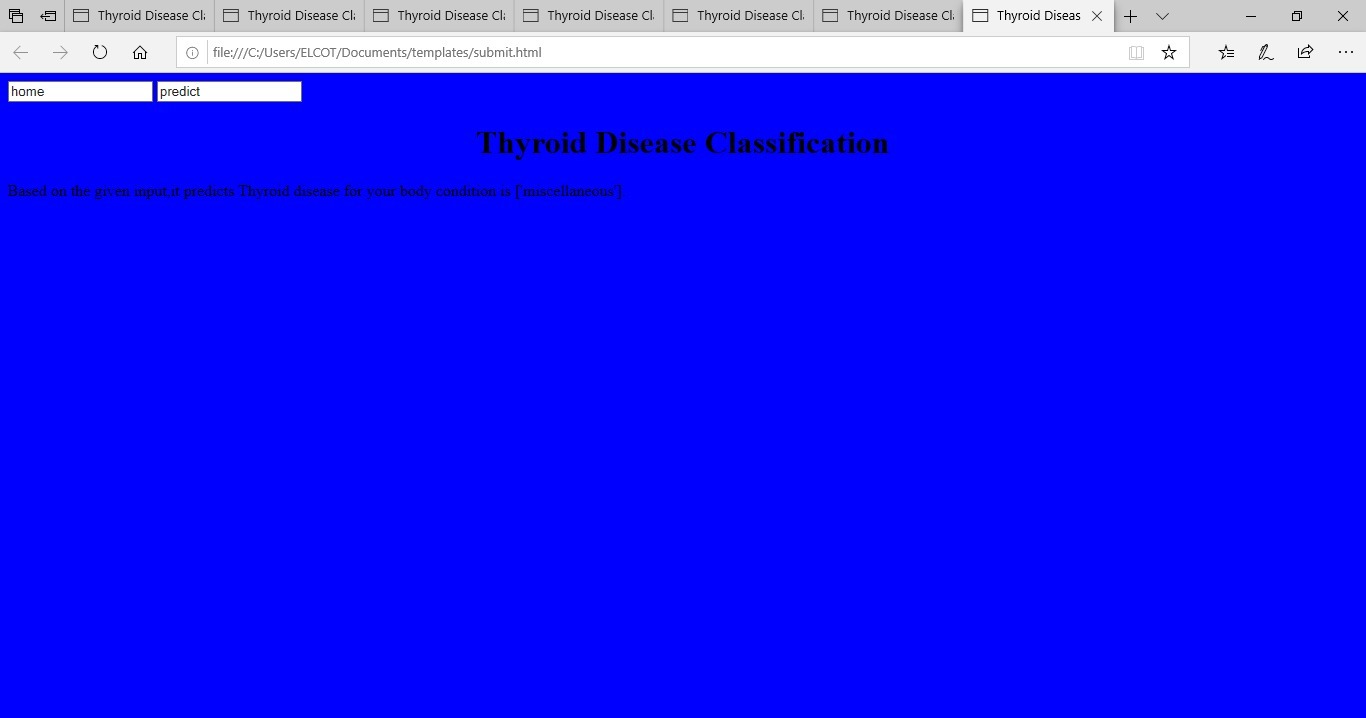
IDEATION AND BRAINSTORMING MAP:



RESULTS:







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. The main function of the gland is to synthesize and secrete the thyroid hormones, namely, thyroxine ( T4), triiodo thyronine (T3) and calcetonine. The thyroid gland plays an important role in the normal metabolic rate and that is why a precise control system is operating to provide the right amounts of thyroid hormones at different conditions, both supra thyroid and auto regulatory mechanisms are involved in this control system. The secretion of thyro tropin ( TSH), The major modulator of thyroid function, is regulated at the level of the pituitary thyro troph by the antagonistic effects of thyroid hormones and the TSH releasing hormone (TRH), the former inhibits and the latter stimulates the synthesis and secretion of ( TSH ). Therefore, excess thyroid hormone leads to decreased secretion of (TSH), and thyroid hormone insufficiency is associated with (TSH) hyper secretion, TSH stimulates all steps of thyroid hormone synthesis and secretion. Ischemic heart disease is defined by world health organization as : myocardial impairment due to imbalance between coronary blood flow and myocardial requirement, caused by changes in the coronary circulation. In the last 20 years, epidemiolgic and experimental studies have provided considerable evidence linking certain risk .The aim of the work is to study the thyroid functions in patients with ischemic heart disease in its different clinical presentations. The relation between the severity of IHD and its complication with thyroid function tests.

FUTURE SCOPE:

The goal of treatment for hypothyroidism is to return blood levels of thyroid-stimulating hormone (TSH) and thyroxine (T4) to the normal range and to alleviate symptoms. Medication

#### Thyroid Disease Prediction

Several thyroid disease detection and classification approaches have been presented in the literature. For example, Garcia et al. [[**9**](https://www.mdpi.com/2072-6694/14/16/3914#B9-cancers-14-03914)] predicted the high probable molecules initiating the thyroid hormone homeostasis using machine learning algorithms RF, LR, GBM, SVM, and deep neural networks (DNN). The early prediction of the molecules is helpful for further testing in the first stages of thyroid disease. The molecular events were obtained from ToxCast datasets for running the experiments. The article reported that Thyroid Peroxidase (TPO) and Thyroid Hormone receptor (TR) achieved the best predictive performance with an F1 score of 0.83 and 0.81, respectively. The authors in [[**10**](https://www.mdpi.com/2072-6694/14/16/3914#B10-cancers-14-03914)] utilized the image processing techniques and feature selection methods to pick the important features from the dataset and achieve the best performance for thyroid disease prediction.

The thyroid disease classification is also a significant problem to be solved in the health industry. Razia et al. compared the performance of various machine learning algorithms to classify Thyroid disease into normal, Hypothyroidism, or hyperthyroidism categories. The authors obtained the datasets from the University of California Irvine (UCI) machine learning library. The dataset contains 7200 samples, and each sample has 21 attributes. The authors reported that DT outperformed the SVM, NB, and multilinear regression (MLR) with 99.23%. However, multi-classification is limited to three categories, and limited information is provided on data preprocessing to assess the applicability of the results for real-time datasets. A multi-kernel SVM is proposed in the paper] to classify thyroid diseases. The authors mentioned that the multi-kernel SVM achieved 97.49% performance accuracy on UCI thyroid datasets. The improved gray wolf optimization performs the feature selection and enhances the performance.

A study performed multiclass hypothyroidism using selective features and machine learning algorithms. Hypothyroidism is classified into four categories. The results show that RF performed well with 99.81% accuracy compared to the SVM, KNN, and DT algorithms. However, the authors did not mention the performance of their proposed methodology for thyroid disease classification. Another study tested three feature selection methods along with SVM, DT, RF, LR, and Naive Bayes (NB) to make early predictions for hypothyroidism

APPENDIX:

A.Source code

***Thyroid\_prediction.py***

# -\*- coding: utf-8 -\*-

"""Thyroid\_Prediction.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1cC2jfIY9GVNlXigL\_9LRKy1qJqY2gXuk

"""

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 google.colab import drive

drive.mount('/content/drive')

with open('/content/drive/My Drive/Colab Notebooks/data.csv', 'r') as dataset:

data = pd.read\_csv(dataset)

data.head()

data.shape

data.isnull().sum()

#Removing Redundant attributes from dataset

#data.drop(['TSH\_measured','T3\_measured','TT4\_measured','T4U\_measured','FTI\_measured','TBG\_measured','referral\_source','patient\_id'])

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) #remapping

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

#data['target'].values\_counts()

data[data.age>100]

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

y=data.iloc[:,-1]

x

x['sex'].unique()

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

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.fit\_transform(x\_test\_bal)

#x\_bal

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']

#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','I131\_treatment','query\_hypothyroid',

'query\_hyperthyroid','lithium','goitre','tumor','hypopituitary','psych','TSH','T3','TT4','T4U','FTI','TB']

#importance = results.importance\_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','I131\_treatmant','query\_hypothyroid','query hyperthyroid','lithium'])

#x\_test\_bal\_drop(['age','sex','on\_thyroxine','query\_on\_thyroxine','on\_antithyroid\_ meds','sick','pregnant','thyroid\_surgery','I131\_treatment','query\_hypothyroid','query\_hyperthyroid','lithium'])

#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

#rfrl = RandomForestClassifier().fit(x\_os,y\_os.values.ravel())

#y\_pred = rfrl.predict(x\_test\_os)

#rfrl = RandomForestClassifier()

#rfrl.fit(x\_os,y\_os.values.ravel())

#y\_pred = rfrl.predict(x\_test\_os)

#y\_pred = rfrl.predict(x\_test\_os)

#print(classification\_report(y\_test\_os,y\_pred))

#train\_score = accuracy\_score(y\_os,rfrl.predict(x\_os))

#train\_score

from xgboost import XGBClassifier

xgb1 = XGBClassifier()

#xgb1.fit(x\_os,y\_os)

#y\_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 = Sequential()

model.add(Dense(units=128,activation='relu',input\_shape=(10,)))

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\_uniforms'))

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)

#rfrl.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.DataFrames(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,rfrl.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',gama=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))

#train\_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.pk1','wb'))

data['target'].unique()

#y['target'].unique()

App.py

# -\*- coding: utf-8 -\*-

"""app.py

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/139TaowLhwEW6u9NS-4ITDf9Yla7HiuND

"""

from flask import Flask, rendere\_template, request

import numpy as np

import pickle

import pandas as pd

# install pyngrok

!pip install pyngrok

#import ngrok

from pyngrok import ngrok

model=pickle.load(open("2OJOQeRY2mo0fC2Zg5lHXE2Nm2A\_3vdhJiTfFHmshHSoQadyC"))

public\_url=ngrok.connect(5000)

print("url:",public)

le=pickle.load("label\_encodder.pkl", 'rb'))

app=Flask(\_name\_)

@app.route("/")

def about():

return render\_template('home.html')

@app.route("|pred", methods=O['POST','GET'])

def preddict():

x=[[float(x) for x in request.form.value()]]

print(x)

col=['goitre','tumor','hypopituitary','psych','TSH','T3','TT4''FTI','TBG']

x=pd.DataFrame(x,columns=col)

#print(x.shape)

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)

ADVANTAGE AND DISADVANTAGE:

Advantage:

\*can help restore normal thyroid hormone levels and improve symptoms

\*may reduce the risk of complications, such as heart disease or osteoporosis

\*can be tailored to individual needs and preferences

\*can be effective for both hyperthyroidism and hypothyroidism

\*may improve overall health and quality of life

Disadvantages:

\*Can have side effects, such as nausea, headache, or hair loss

\*may require regular monitoring and adjustments

APPLICATIONS:

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.

thyroid controls how much energy your body uses (the metabolic rate). It's also involved in digestion, how your heart and muscles work, brain development and bone health. When the thyroid gland does not make enough thyroid (called hypothyroidism), many of the body's functions slow down