PROJECT REPORT

1.INTRODUCTION

1.1 Overview:

Thyroid Disease Classification Using ML

Project Description:

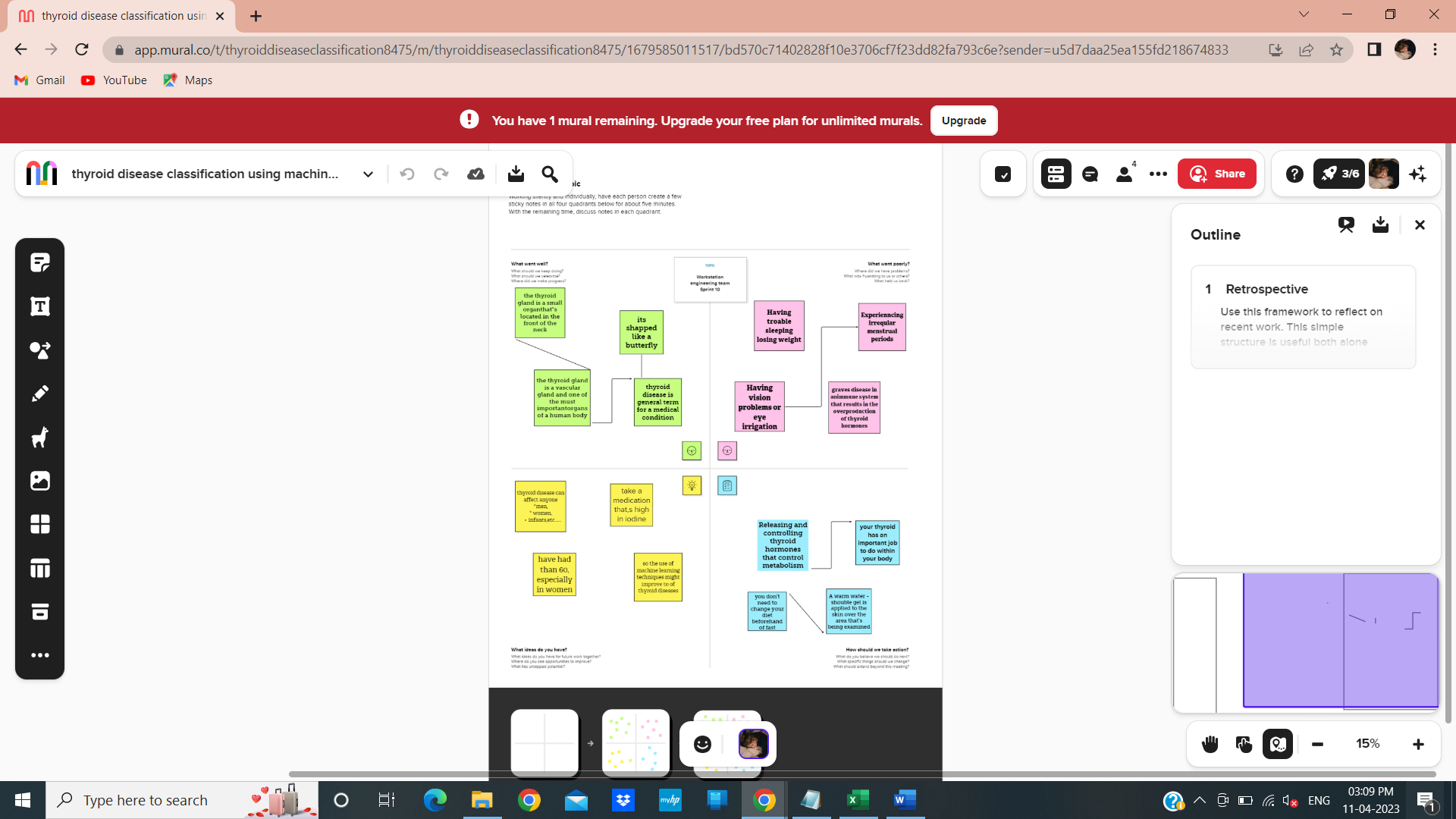
* Thyroid disease is a general term for a medical condition that keeps your thyroid from making the right amount of hormones. Your thyroid typically makes hormones that keep your body functioning normally. When the thyroid makes too much thyroid hormone, your body uses energy too quickly. This is called hyperthyroidism.

1.2purpose

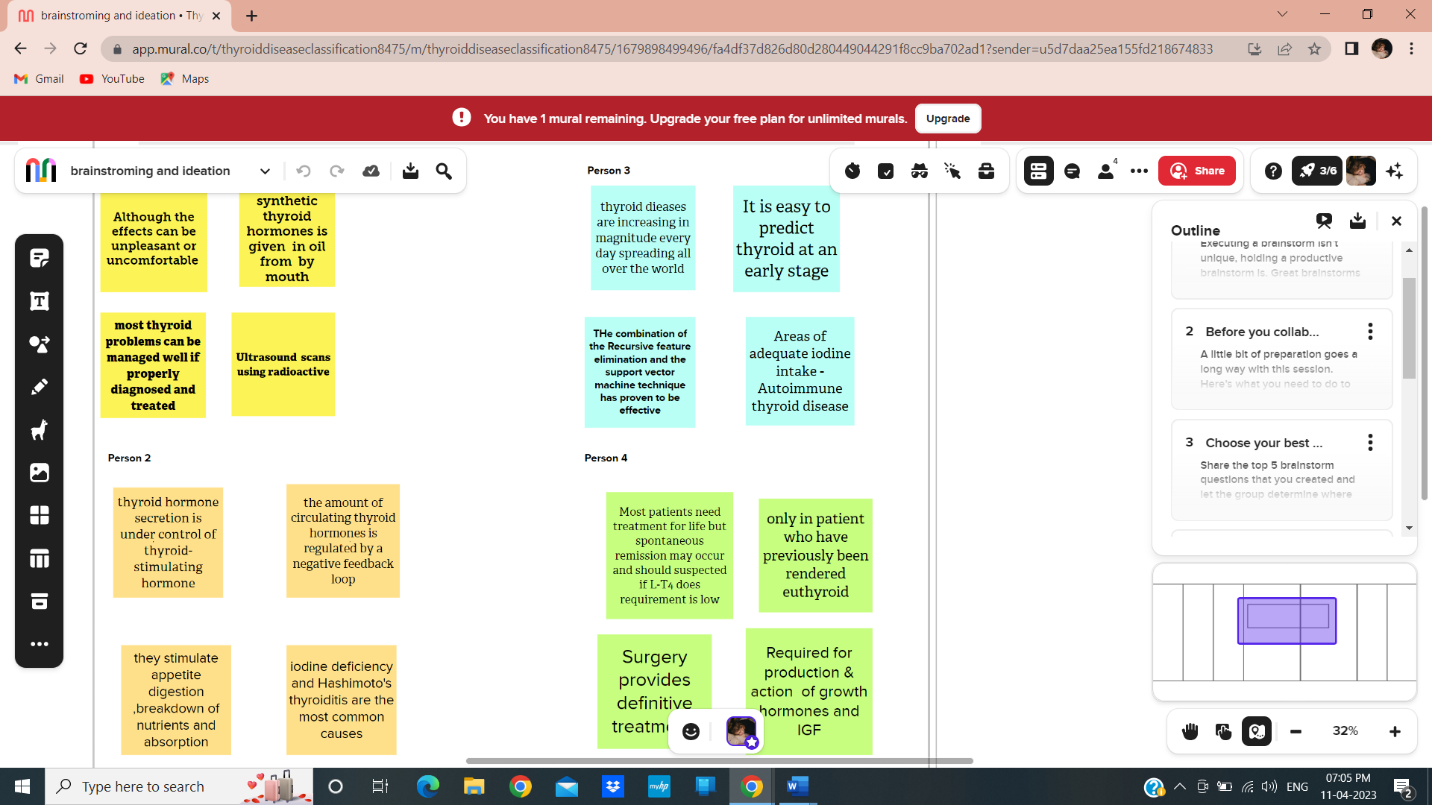
* The thyroid gland is avital hormone gland: It plays a major role in the metabolism , growth and development of the human body.it helps to regulate many body functions by constantly releasing a steady amount of thyroid hormones into the bloodstream.

Problem Definition & Design Thinking

2.1 Empathy Map

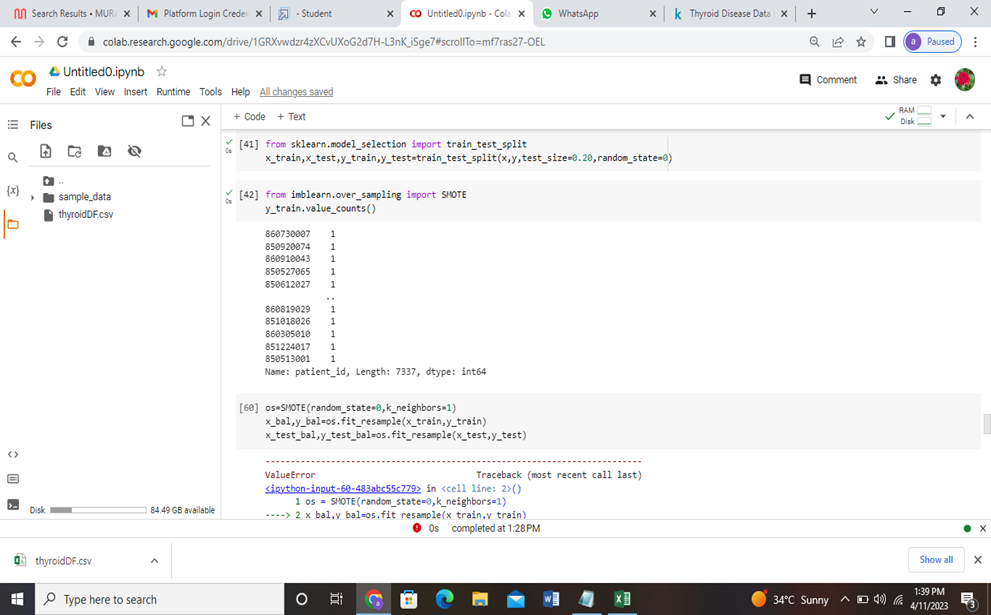


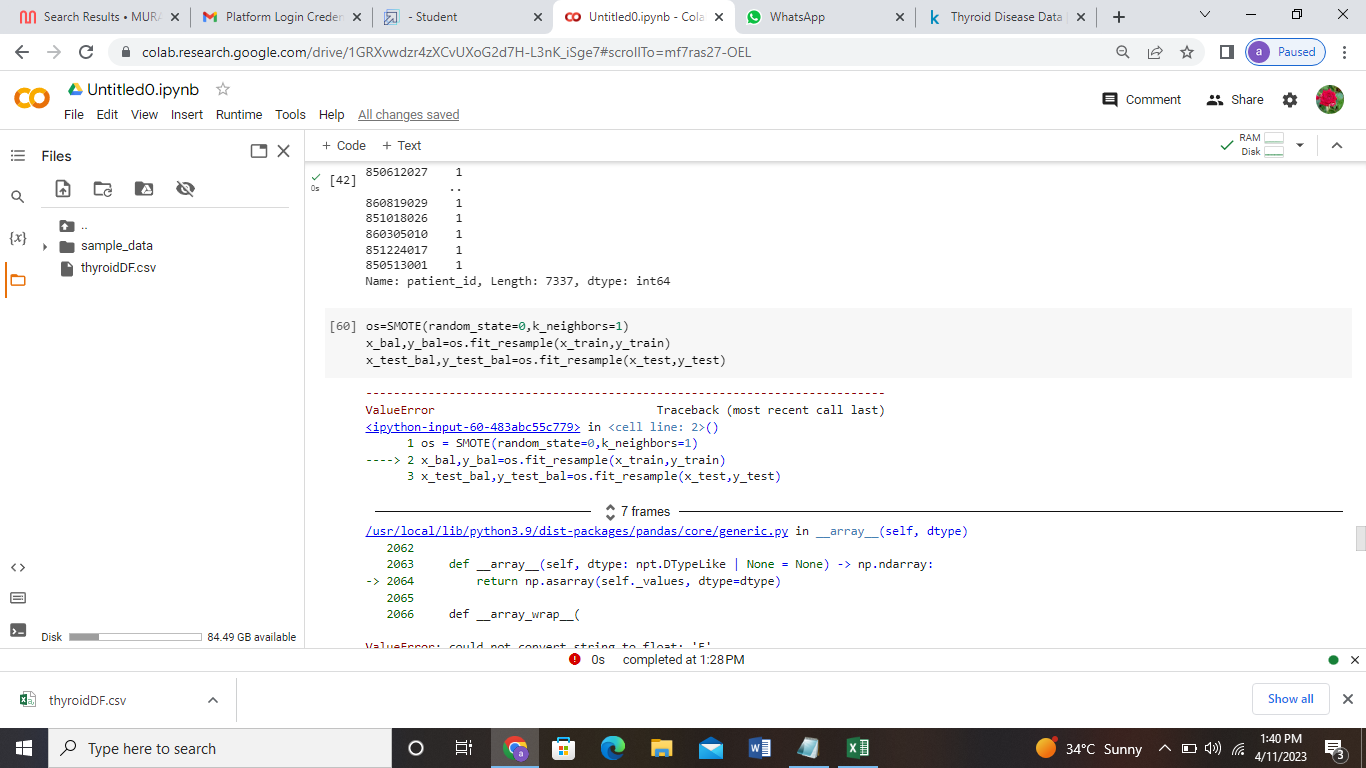
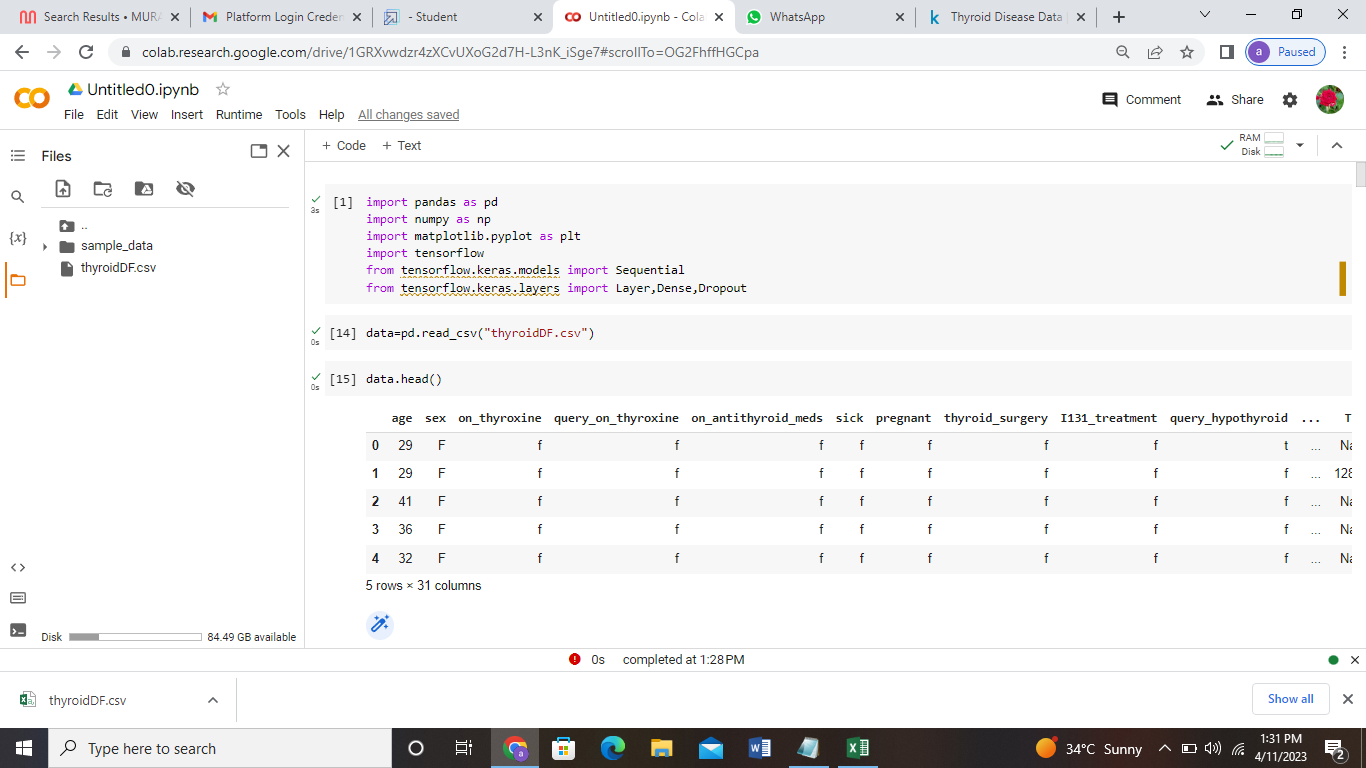
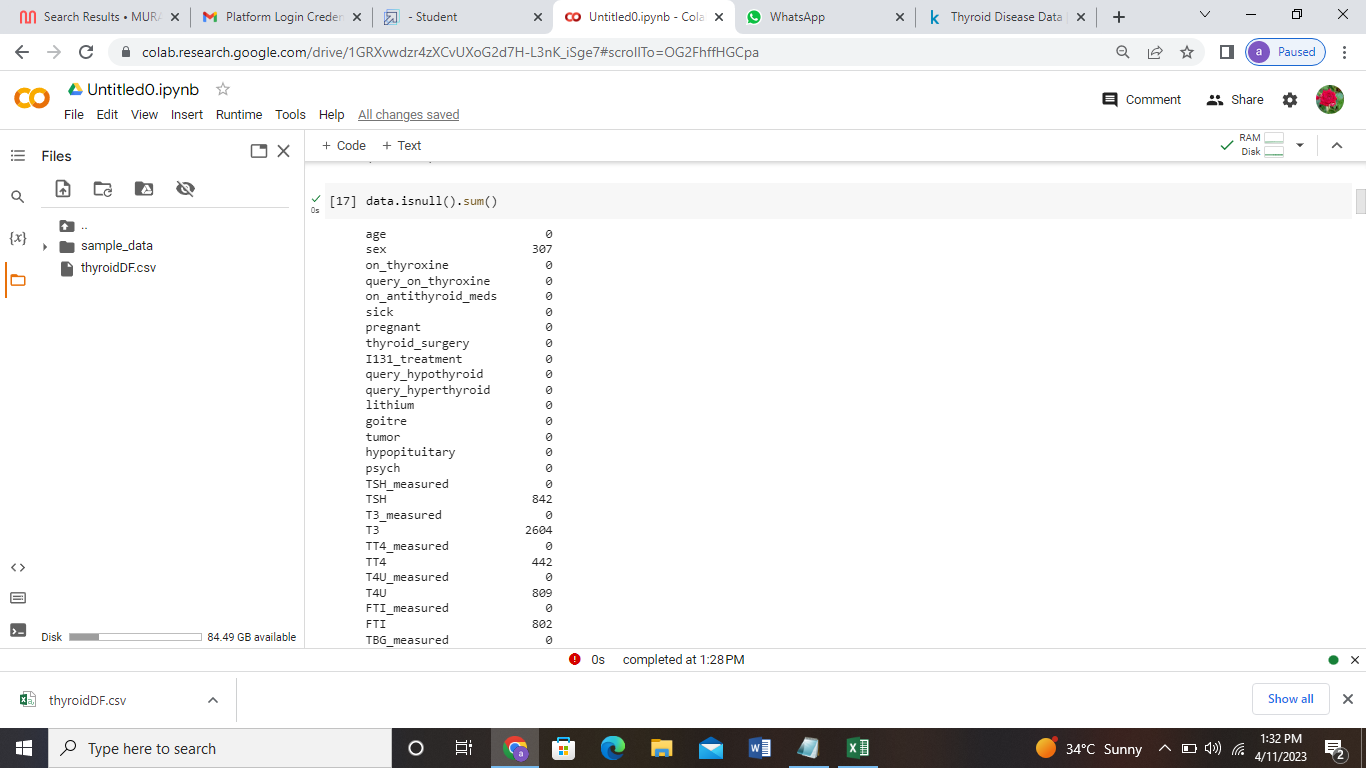
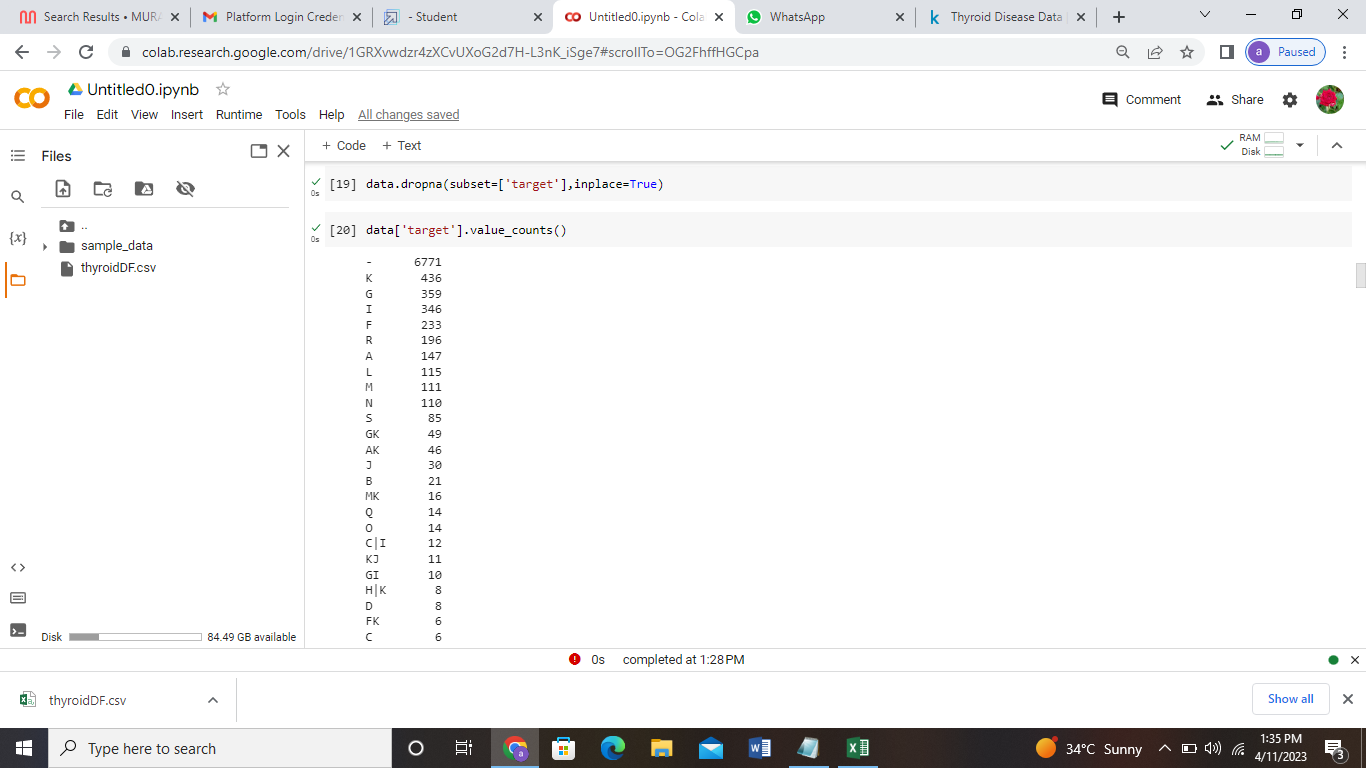
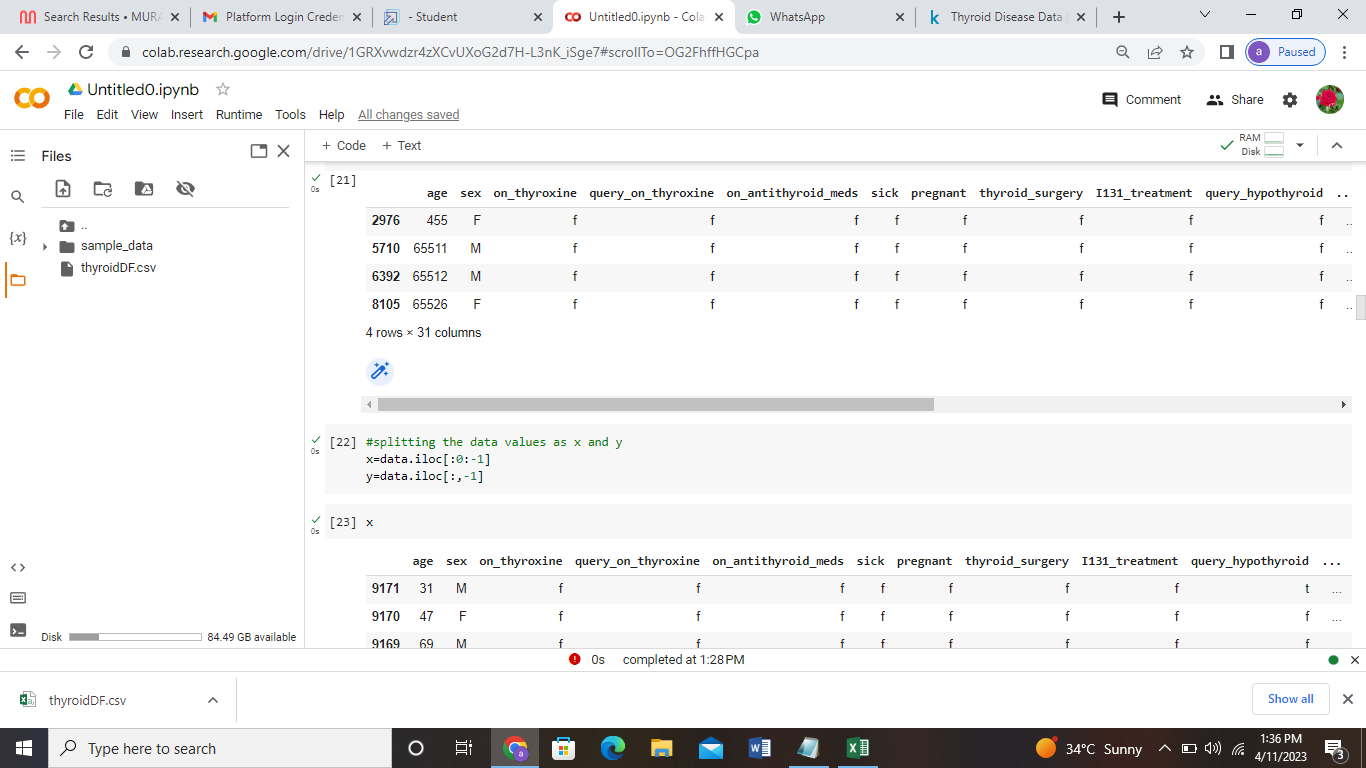
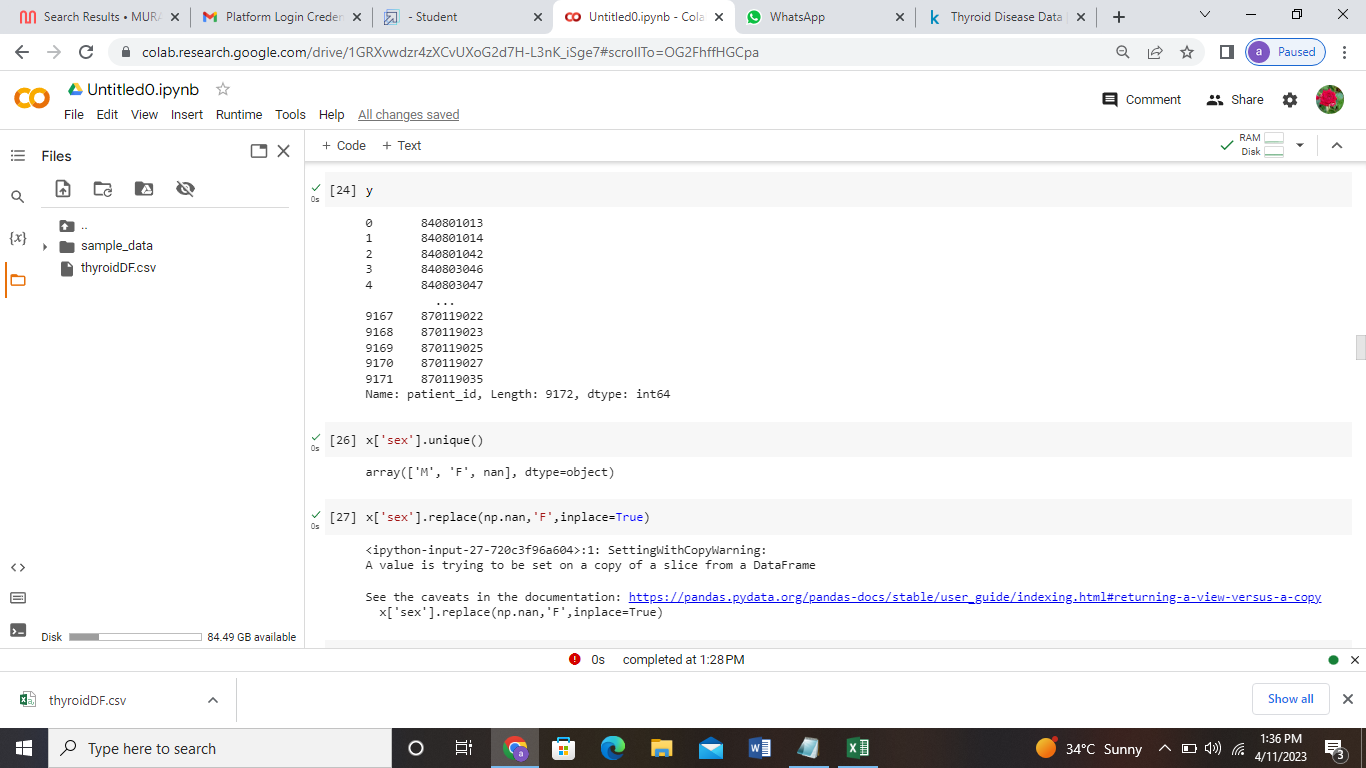
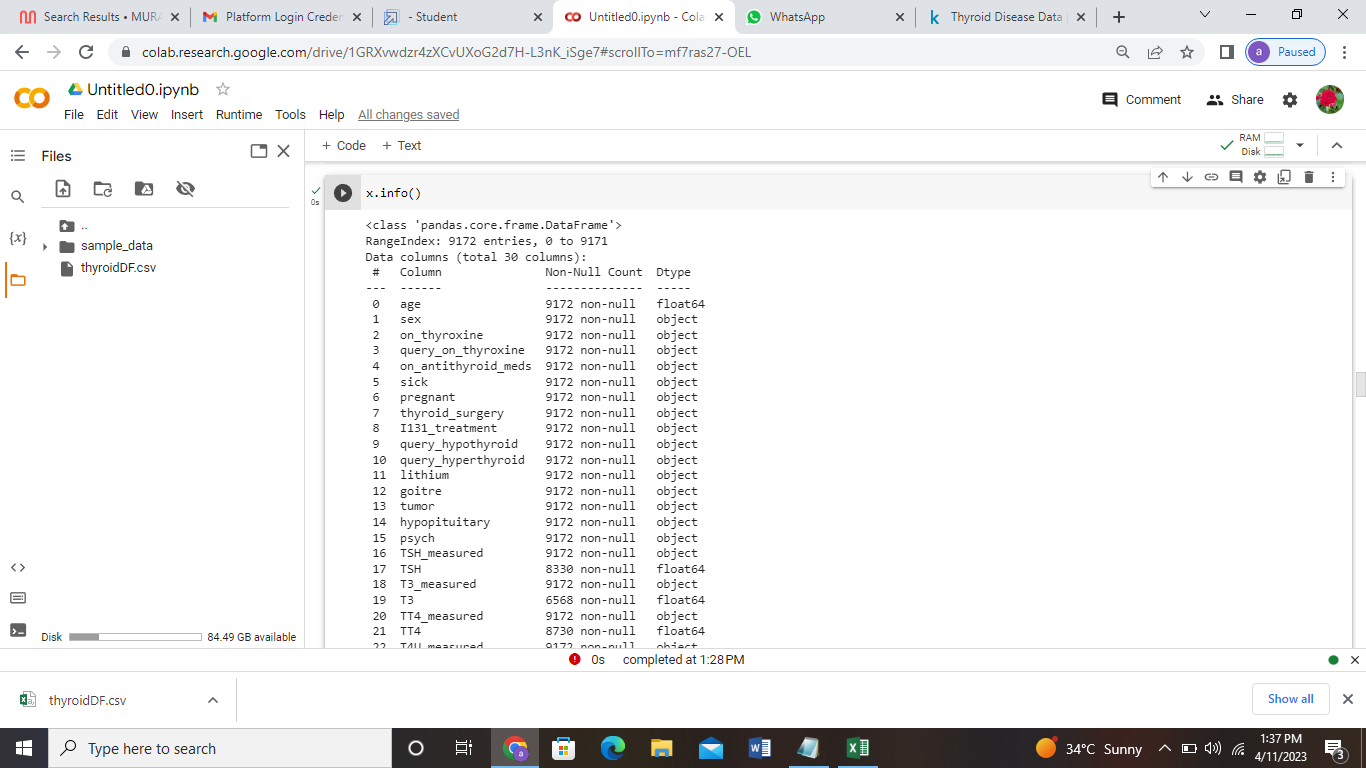
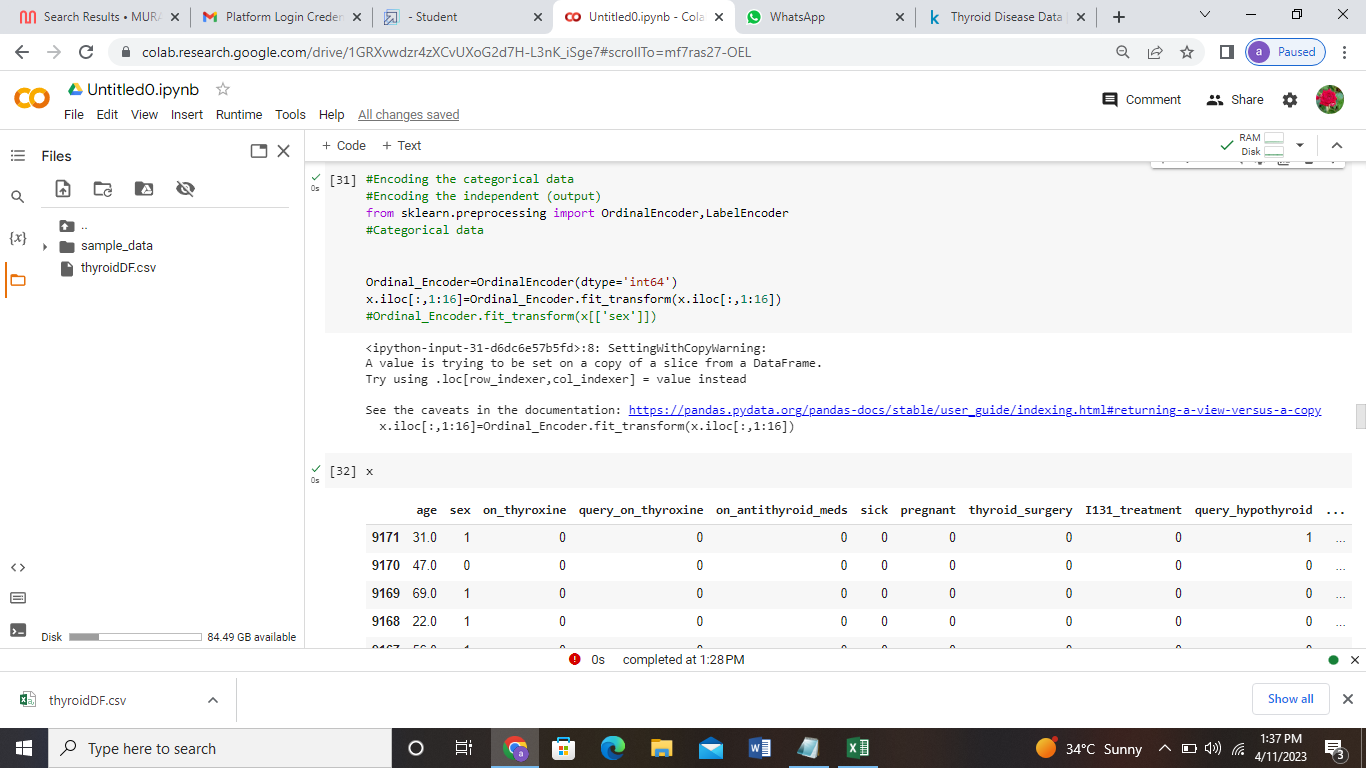
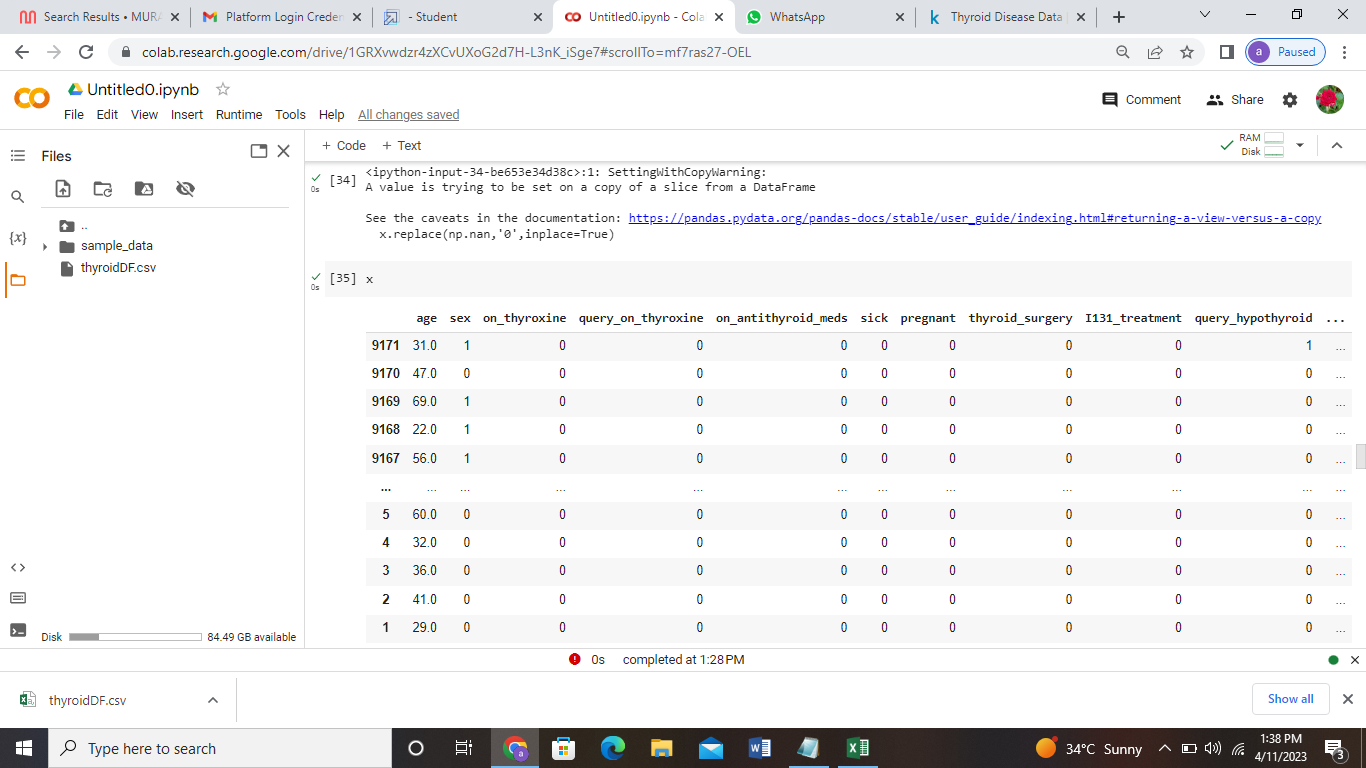
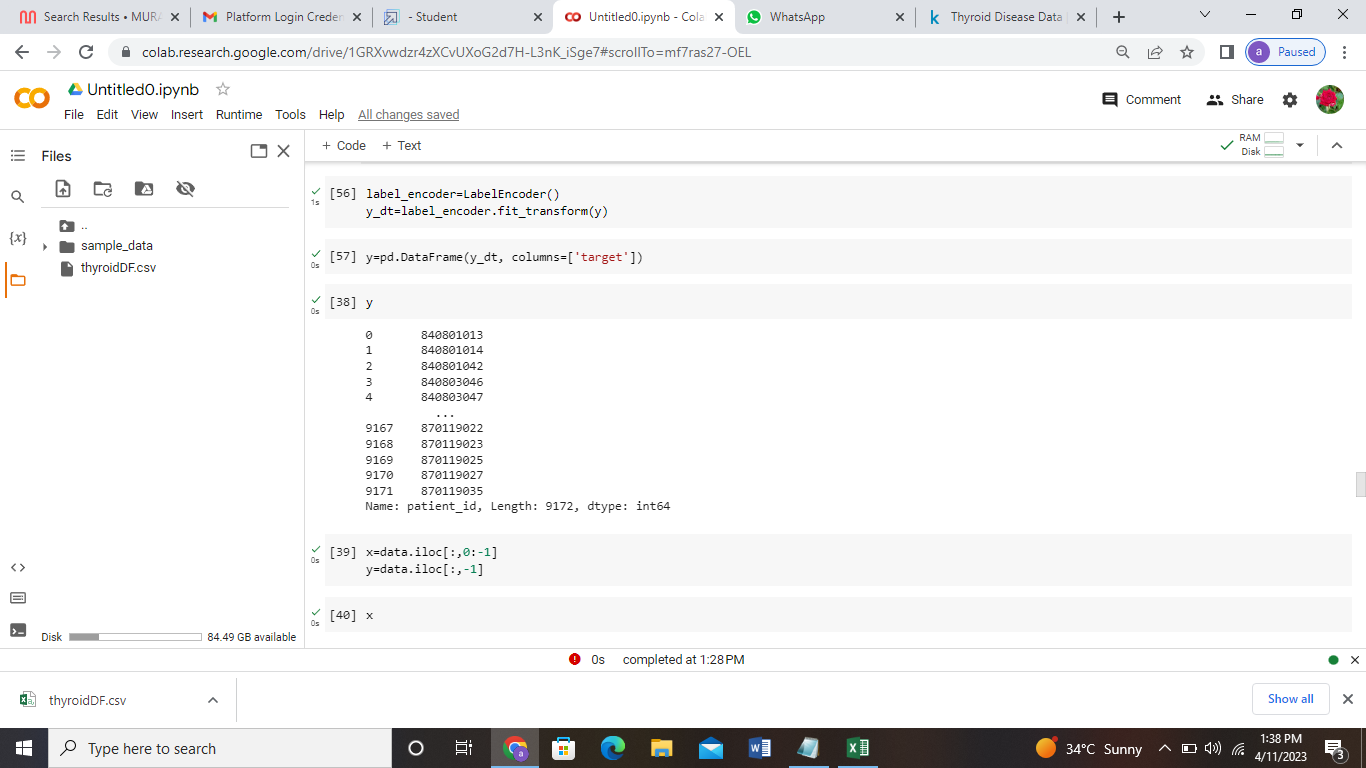
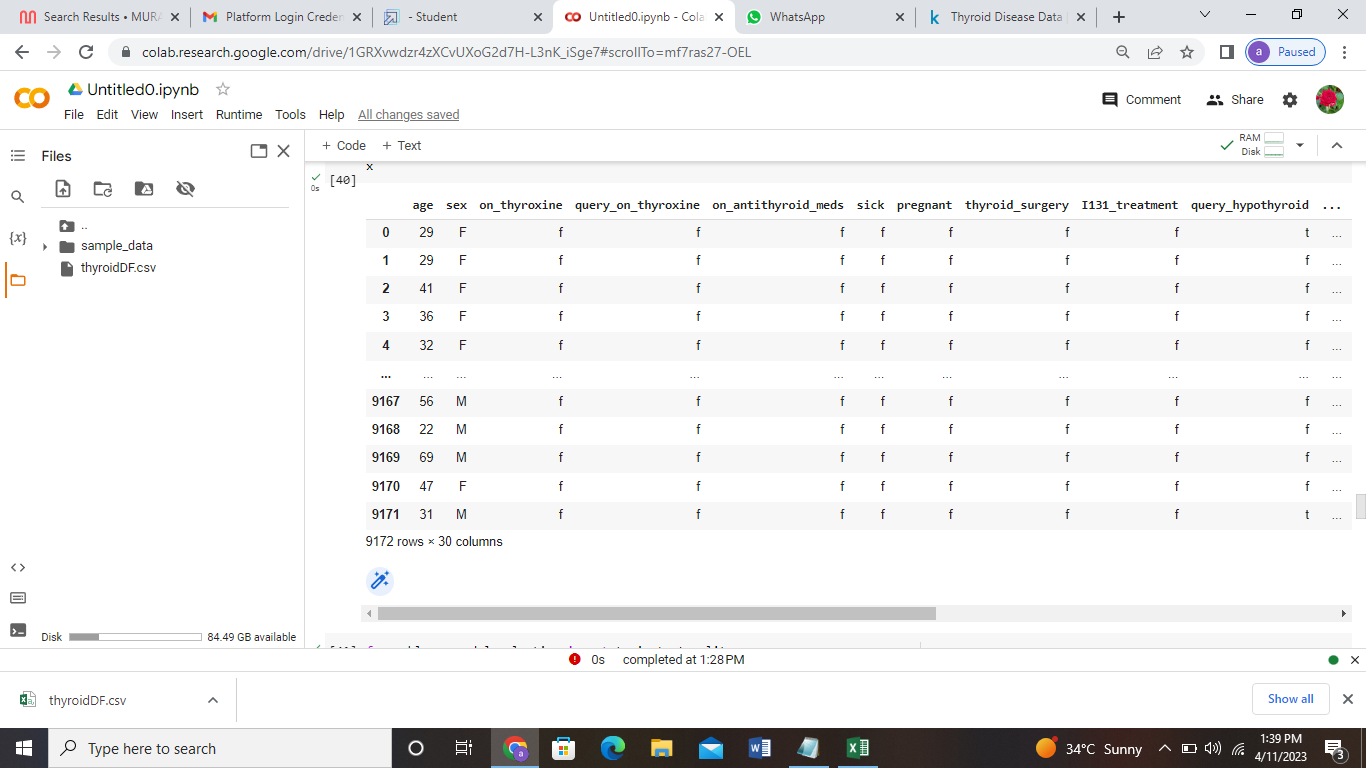
2.2 ideation & Brainstorming Map



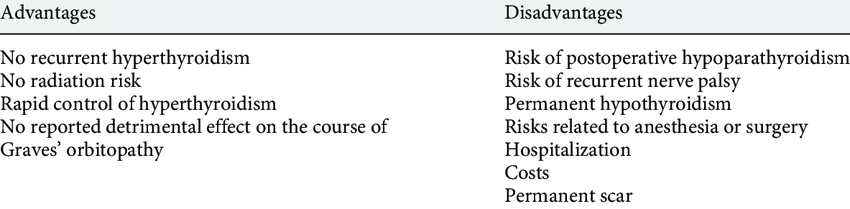
3.RESULT

Thyroid Disease Analysis Coding Output





4. Thyroid disease Advantages and Disadvantages



5.APPLICATION

* 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.
* Hospitals.
* Medical research Department.

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.

7.FUTURE SCOPE

* Hospitals.
* Specialty Clinics.
* Medical Research Department.
* Patient(Body).
* Getting enough Sleep.
* Exercising regularly.
* Taking Medications properly.
* Getting tested regularly.
* Watching your diet.

8.APPENDIX

Source code

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

"""Untitled0.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1GRXvwdzr4zXCvUXoG2d7H-L3nK\_iSge7

"""

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("thyroidDF.csv")

data.head()

data.shape

data.isnull().sum()

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

#remapping target valueas ton diagnostic group

dignoses={'A':'hypothyroid conditions',

'B':'hypothyroid conditions',

'C':'hypothyroid conditions',

'D':'hypothyroid conditions',

'E':'hypothyroid conditions',

'F':'hypothyroid conditions',

'G':'hypothyroid conditions',

'H':'hypothyroid conditions',

'I':'binding protein',

'J':'binding protein',

'K':'binding protein',

'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'].value\_counts()

data[data.age>100]

#splitting the data values as x and y

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

y=data.iloc[:,-1]

x

y

x['sex'].unique()

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

x['sex'].value\_counts()

#Converting the data type

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()

#Encoding the categorical data

#Encoding the independent (output)

from sklearn.preprocessing import OrdinalEncoder,LabelEncoder

#Categorical data

Ordinal\_Encoder=OrdinalEncoder(dtype='int64')

x.iloc[:,1:16]=Ordinal\_Encoder.fit\_transform(x.iloc[:,1:16])

#Ordinal\_Encoder.fit\_transform(x[['sex']])

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 StandardScalar

sc=StandardScalar()

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(rtf,x\_bal,y\_bal,scoring='accuracy')