ARIGNAR ANNA GOVERNMENT ARTS COLLEGE

VILLUPURAM-605 602

Department of computer of applications

MACHINE LEARNING WITH PYTHON

Project Title: Thyroid Disease classification using ML

Team Id:

Team Leader:M.MUTHURAMAN

(4C9ED657E973412895E4DDB64B246A2E)

Team Members:M.PREMKUMAR

(3AE810950D4EEEEB68B45B1D58C2E69D)

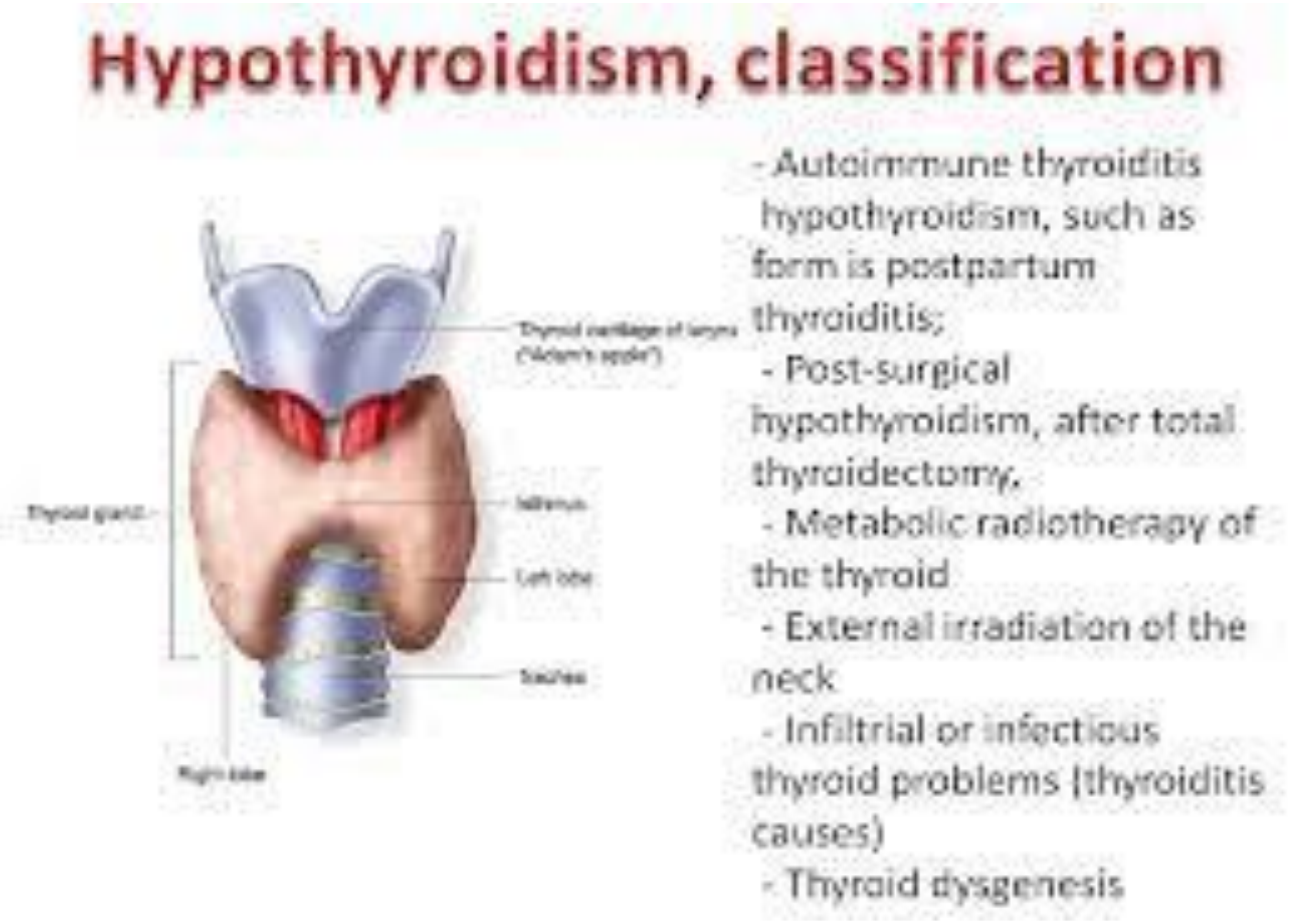
Team Members:J.RAGUNATH

(57E55B4164972BBB2FB4A732A0BDEA0E)

Team Members:R.RAJAMANI

(E1C4C54DC5D79669D8945CEF3F6A1A80)

Abstract:

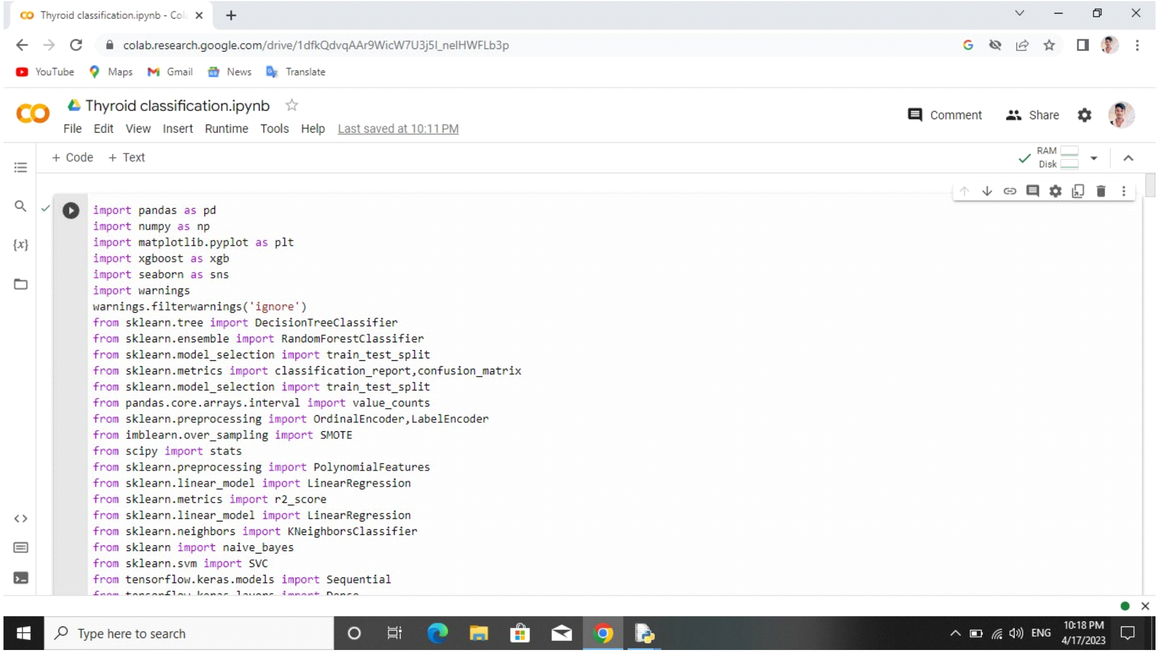


Hypothyroidism, classification

- Autoimmune thyroiditis hypothyroidism, such as form is postpartum thyroiditis;
- Post-surgical hypothyroidism, after total thyroidectomy,
- Metabolic radiotherapy of the thyroid
- External irradiation of the neck
- Infiltrial or infectious thyroid problems (thyroiditis causes)
- Thyroid dysgenesis

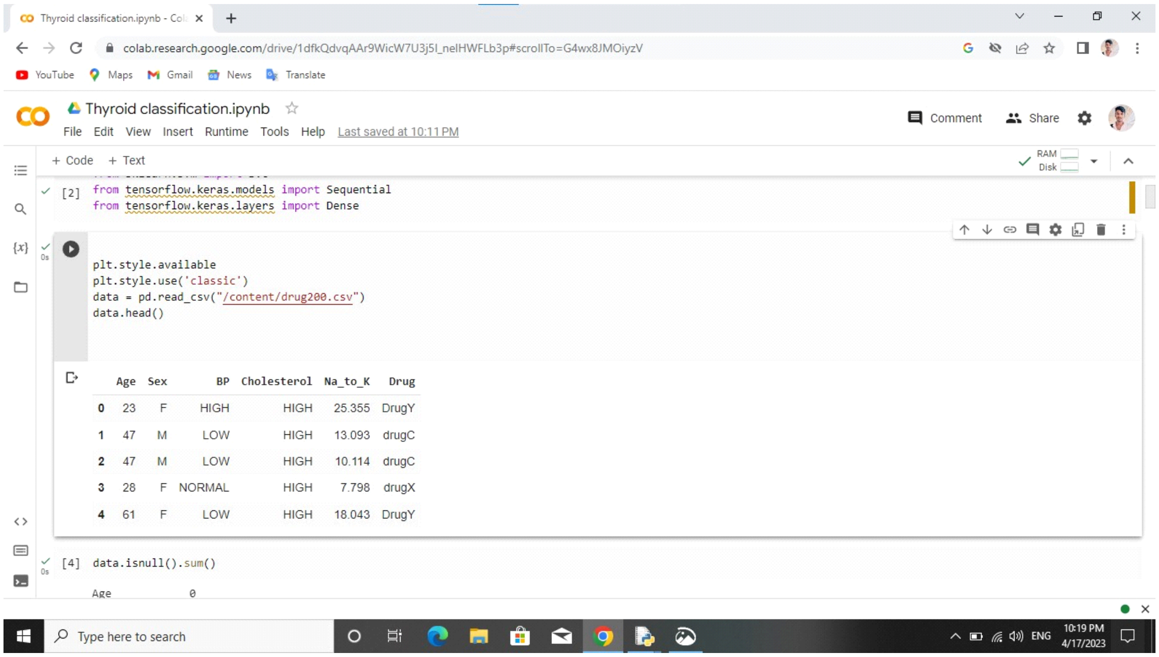
Thyroid disease is a general term for a medical condition that keeps your thyroid from making the right amount of hormones. Your thyroid typically makes hormones that keep your body functioning normally. When the thyroid makes too much thyroid hormone, your body uses energy too quickly. This is called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

Import libraries


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p
YouTube Maps Gmail News Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment Share
+ Code + Text
RAM Disk
Type here to search
ENG 10:18 PM 4/17/2023

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import xgboost as xgb
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.model_selection import train_test_split
from pandas.core.arrays.interval import value_counts
from sklearn.preprocessing import OrdinalEncoder,LabelEncoder
from imblearn.over_sampling import SMOTE
from scipy import stats
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn import naive_bayes
from sklearn.svm import SVC
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

Reading .CSV File


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=G4wx8JMOiyzV
YouTube Maps Gmail News Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment Share
+ Code + Text
RAM Disk
Type here to search
ENG 10:19 PM 4/17/2023

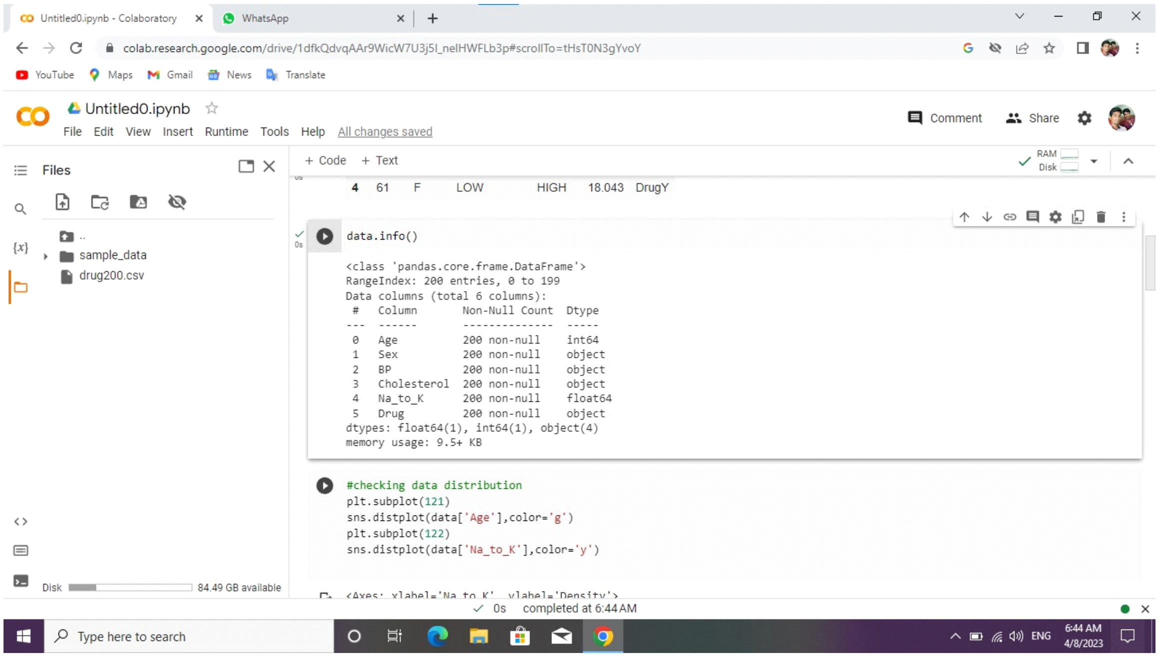
```
[2] from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense
```

```
plt.style.available
plt.style.use('classic')
data = pd.read_csv("/content/drug200.csv")
data.head()
```

```
   Age  Sex      BP  Cholesterol  Na_to_K   Drug
0   23    F    HIGH         HIGH   25.355  DrugY
1   47    M     LOW         HIGH   13.093  drugC
2   47    M     LOW         HIGH   10.114  drugC
3   28    F  NORMAL         HIGH    7.798  drugX
4   61    F     LOW         HIGH   18.043  DrugY
```

```
[4] data.isnull().sum()

    Age          0
```

data info:



Untitled0.ipynb - Colaboratory
WhatsApp
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=tHsT0N3gYvoY
Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
drug200.csv
Disk
84.49 GB available
+ Code + Text
RAM
Disk
completed at 6:44 AM
Type here to search
6:44 AM
4/8/2023

```
4   61   F   LOW   HIGH   18.043   DrugY
```

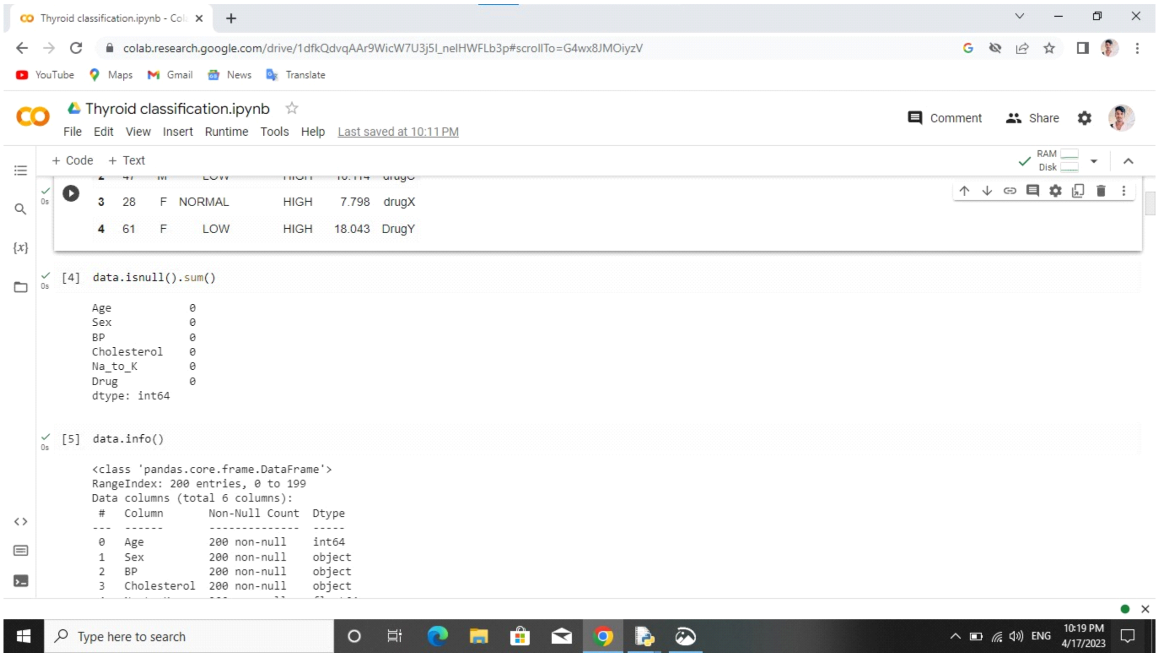
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          200 non-null    int64  
 1   Sex          200 non-null    object 
 2   BP           200 non-null    object 
 3   Cholesterol  200 non-null    object 
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

```
#checking data distribution
plt.subplot(121)
sns.distplot(data['Age'],color='g')
plt.subplot(122)
sns.distplot(data['Na_to_K'],color='y')
```

```
<Axes: xlabel='Na_to_K', ylabel='Density'>
```

Checking null values:



Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=G4wx8JMOiyzV
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment
Share
+ Code + Text
RAM
Disk
Type here to search
10:19 PM
4/17/2023

```
3   28   F   NORMAL   HIGH    7.798   drugX
4   61   F   LOW      HIGH   18.043   DrugY
```

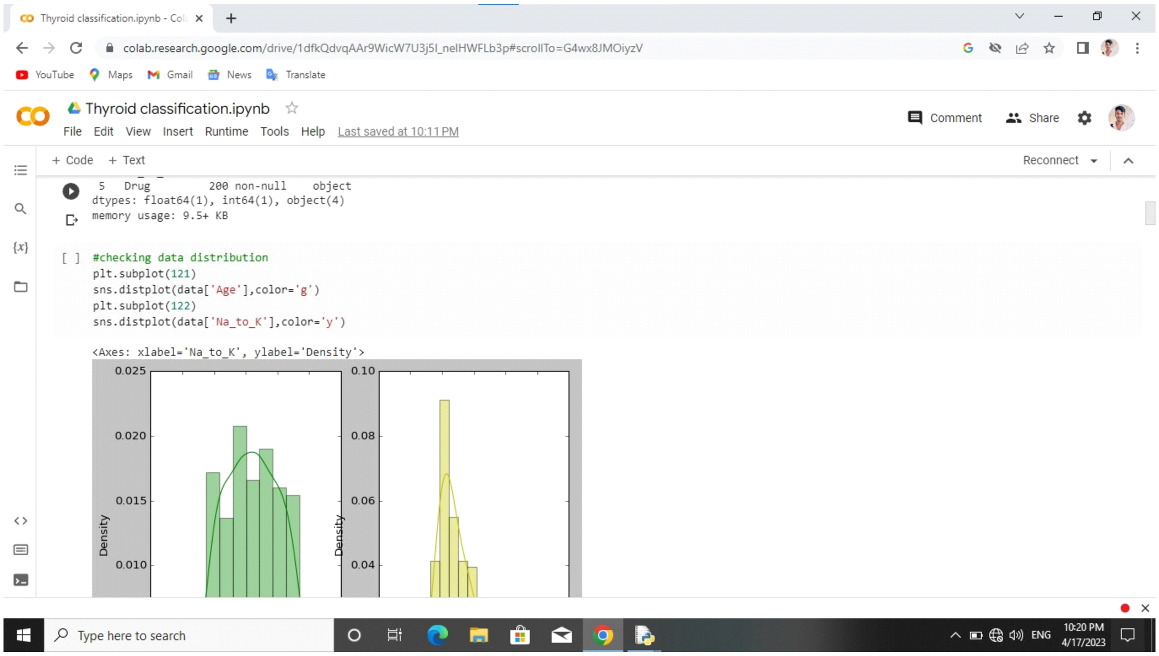
```
[4] data.isnull().sum()
```

```
Age            0
Sex            0
BP             0
Cholesterol    0
Na_to_K        0
Drug           0
dtype: int64
```

```
[5] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          200 non-null    int64  
 1   Sex          200 non-null    object 
 2   BP           200 non-null    object 
 3   Cholesterol  200 non-null    object 
```

Checking data   distribution:

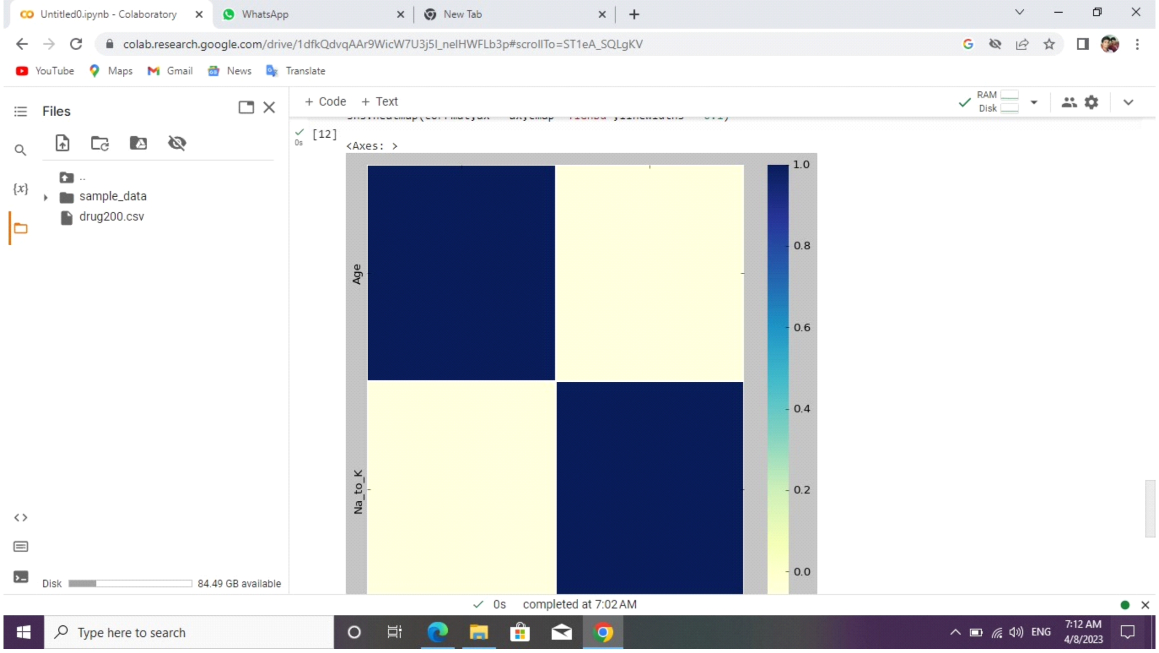


Thyroid classification.ipynb - Co
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=G4wx8JMOiyzV
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment
Share
+ Code + Text
Reconnect
0.025
0.020
0.015
0.010
Density
0.10
0.08
0.06
0.04
Density
Type here to search
ENG
10:20 PM
4/17/2023

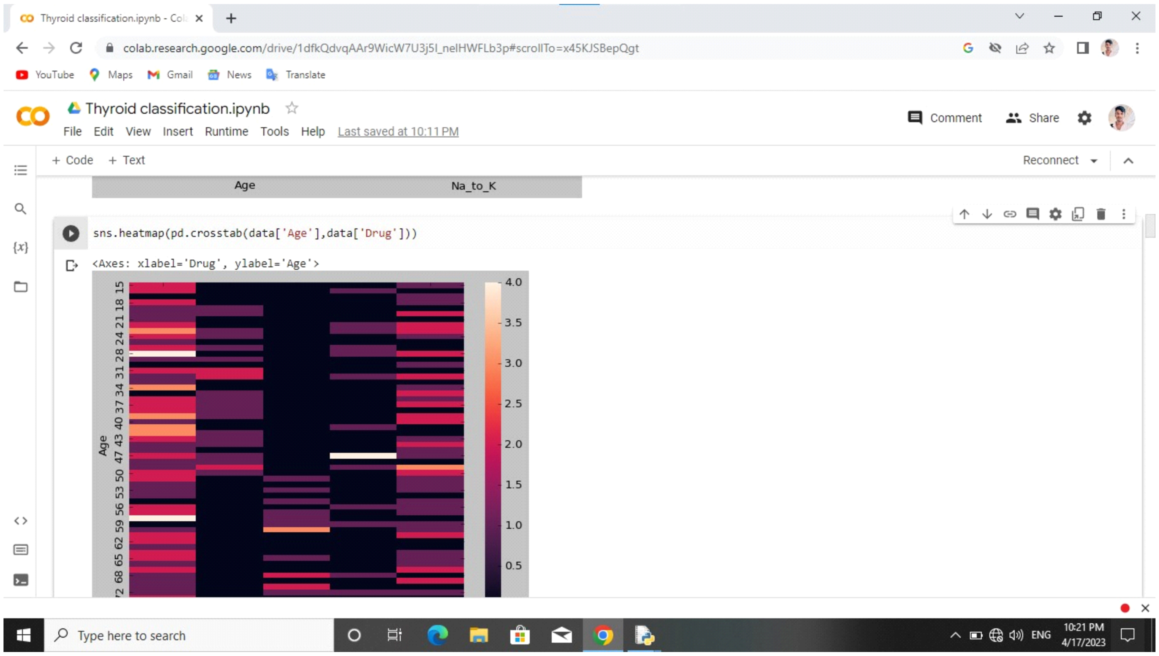
```
 5   Drug         200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

```
#checking data distribution
plt.subplot(121)
sns.distplot(data['Age'],color='g')
plt.subplot(122)
sns.distplot(data['Na_to_K'],color='y')
```

```
<Axes: xlabel='Na_to_K', ylabel='Density'>
```



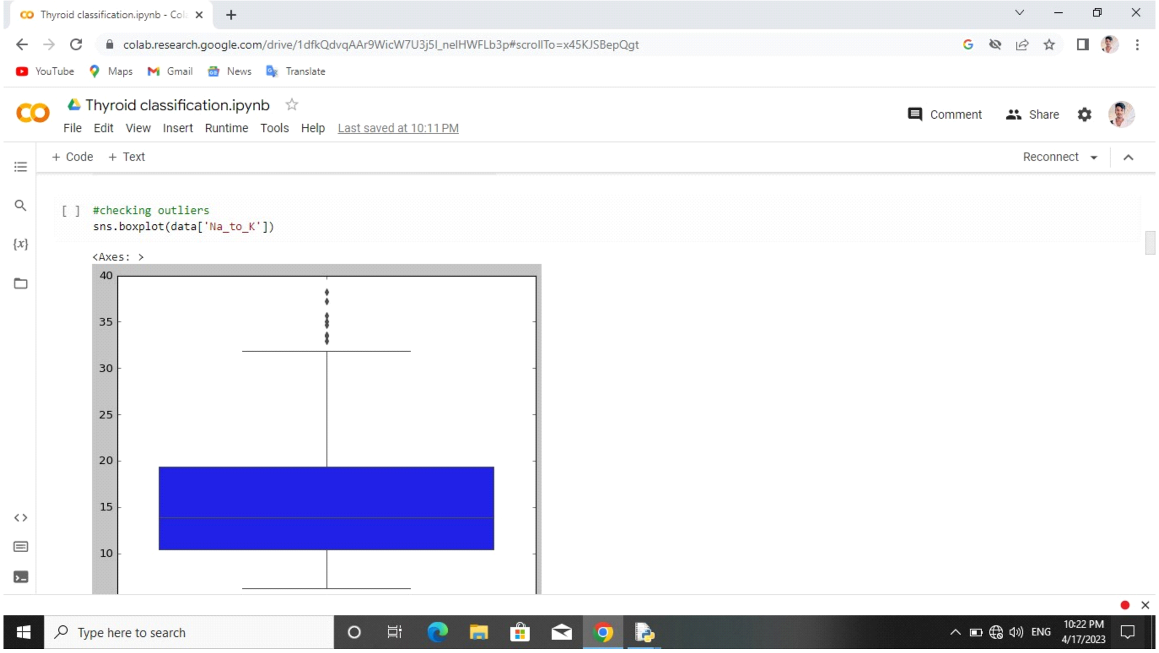
```
[12]
<Axes: >
```


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=x45KJSBepQgt
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment
Share
+ Code + Text
Reconnect
Age
Na_to_K
Age
15 18 21 24 28 31 34 37 40 43 47 50 53 56 59 62 65 68 72
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
Type here to search
ENG
10:21 PM
4/17/2023

```
sns.heatmap(pd.crosstab(data['Age'],data['Drug']))
```

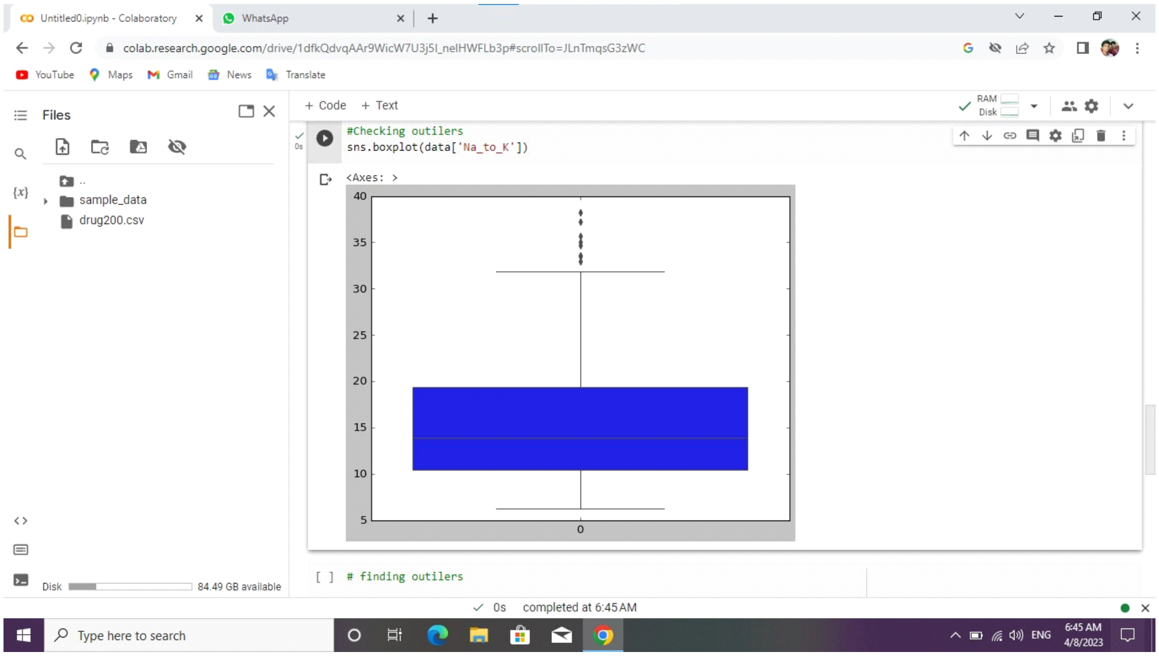
```
<Axes: xlabel='Drug', ylabel='Age'>
```

Checking outlier:


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=x45KJSBepQgt
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment
Share
+ Code + Text
Reconnect
40
35
30
25
20
15
10
Type here to search
ENG
10:22 PM
4/17/2023

```
#checking outliers
sns.boxplot(data['Na_to_K'])
```

```
<Axes: >
```

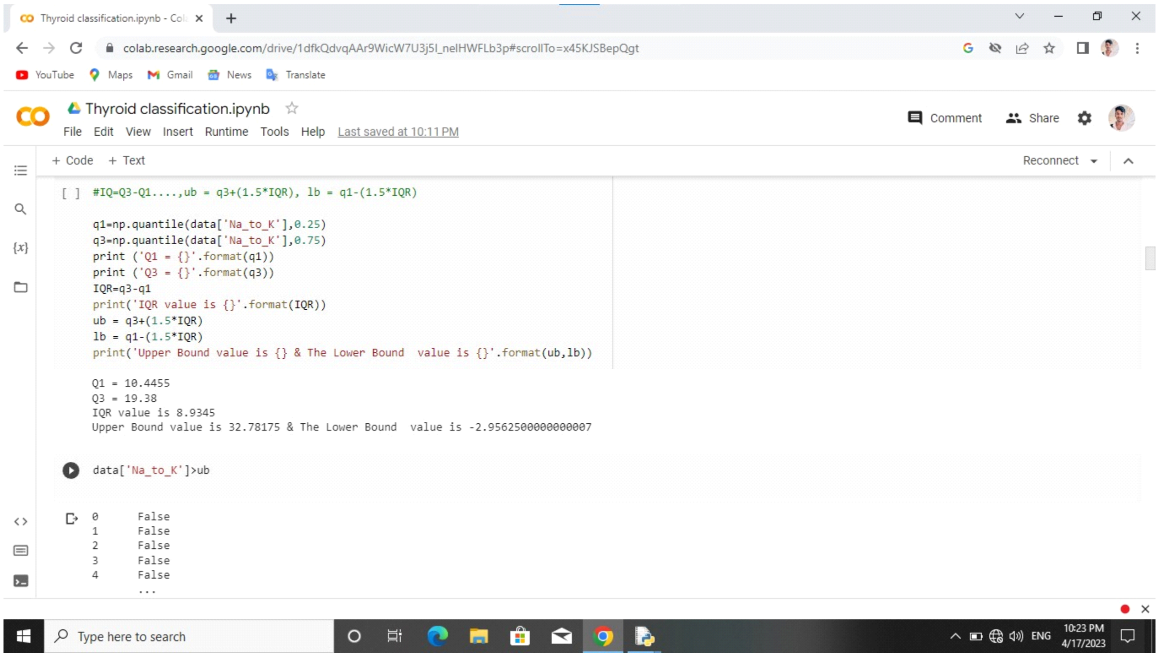


```
#Checking outilers
sns.boxplot(data['Na_to_K'])
```

```
<Axes: >
```

```
# finding outilers
```

Deleting outliers:



Thyroid classification.ipynb

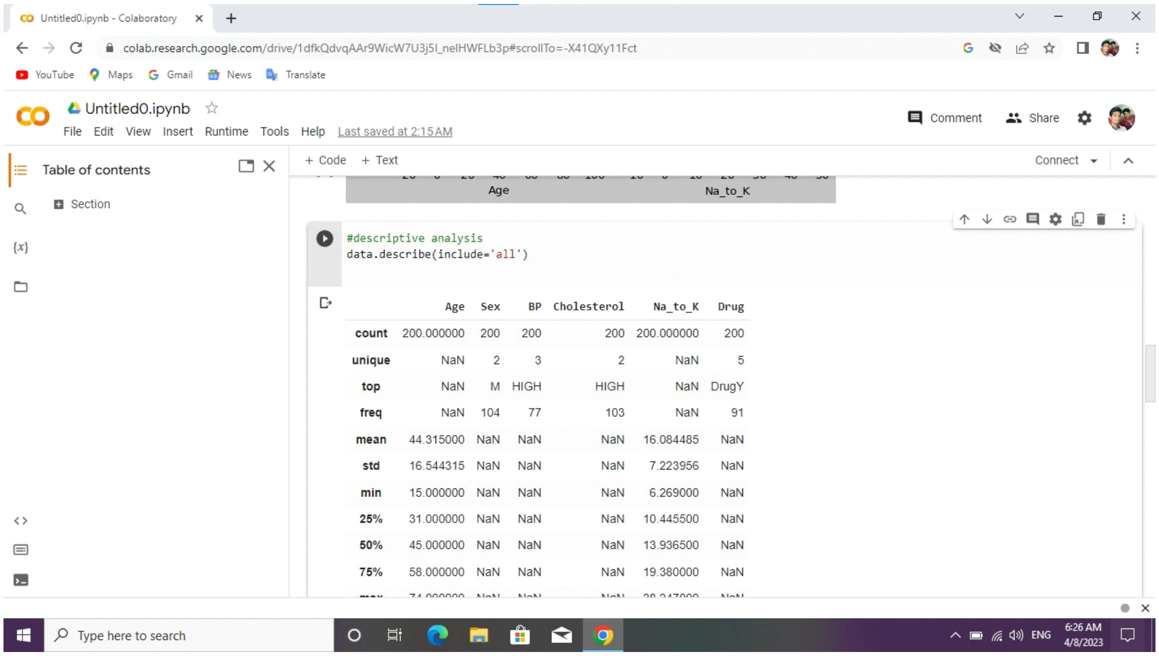
```
#IQ=Q3-Q1....,ub = q3+(1.5*IQR), lb = q1-(1.5*IQR)

q1=np.quantile(data['Na_to_K'],0.25)
q3=np.quantile(data['Na_to_K'],0.75)
print ('Q1 = {}'.format(q1))
print ('Q3 = {}'.format(q3))
IQR=q3-q1
print('IQR value is {}'.format(IQR))
ub = q3+(1.5*IQR)
lb = q1-(1.5*IQR)
print('Upper Bound value is {} & The Lower Bound  value is {}'.format(ub,lb))
```

```
Q1 = 10.4455
Q3 = 19.38
IQR value is 8.9345
Upper Bound value is 32.78175 & The Lower Bound  value is -2.9562500000000007
```

```
data['Na_to_K']>ub
```

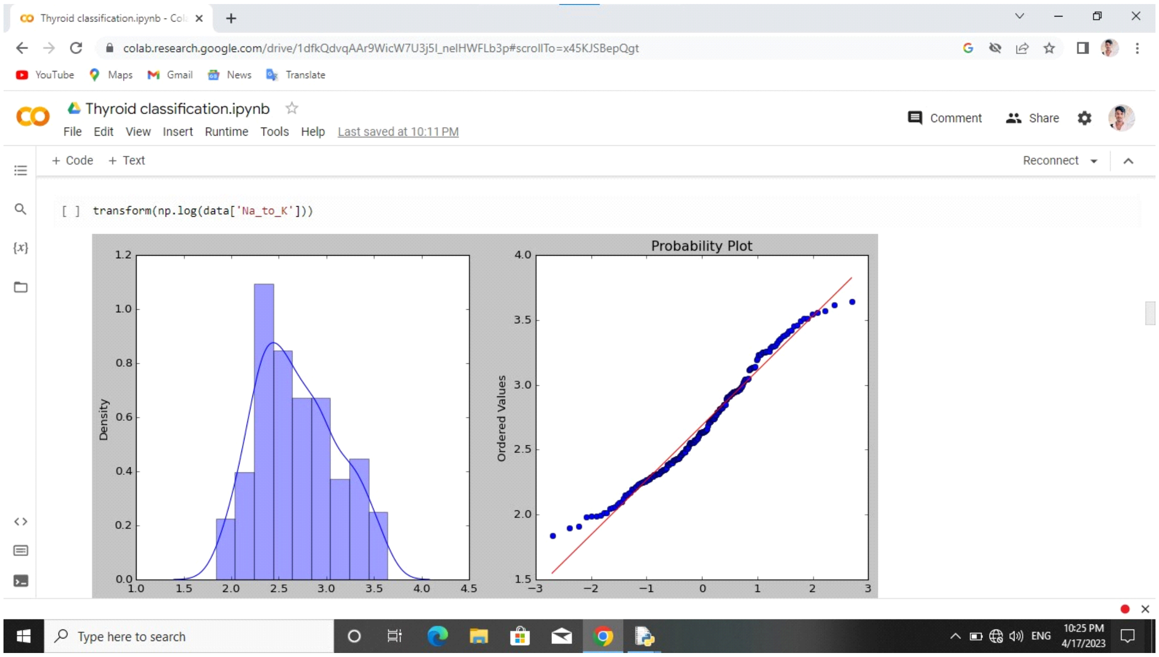
```
0      False
1      False
2      False
3      False
4      False
       ...
```

descripitive analysis:

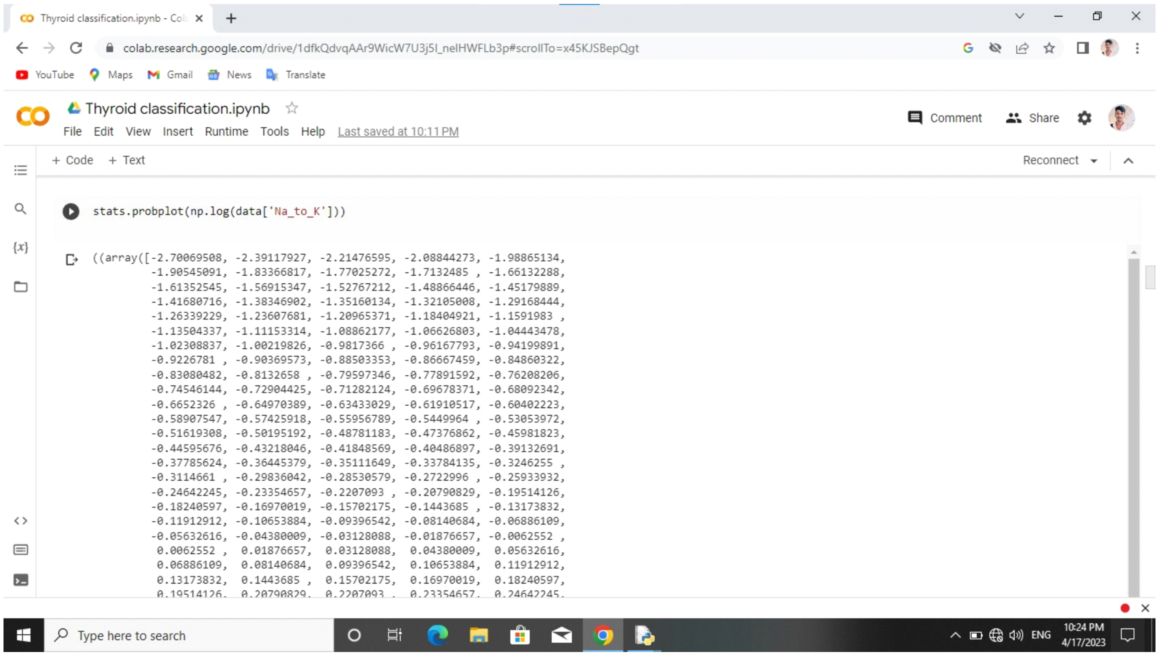


```
#descriptive analysis
data.describe(include='all')
```

```
              Age  Sex    BP  Cholesterol     Na_to_K   Drug
 count 200.000000  200   200          200  200.000000    200
unique        NaN    2     3            2         NaN      5
   top        NaN    M  HIGH         HIGH         NaN  DrugY
  freq        NaN  104    77          103         NaN     91
  mean  44.315000  NaN   NaN          NaN   16.084485    NaN
   std  16.544315  NaN   NaN          NaN    7.223956    NaN
   min  15.000000  NaN   NaN          NaN    6.269000    NaN
   25%  31.000000  NaN   NaN          NaN   10.445500    NaN
   50%  45.000000  NaN   NaN          NaN   13.936500    NaN
   75%  58.000000  NaN   NaN          NaN   19.380000    NaN
   max  74.000000  NaN   NaN          NaN   38.247000    NaN
```

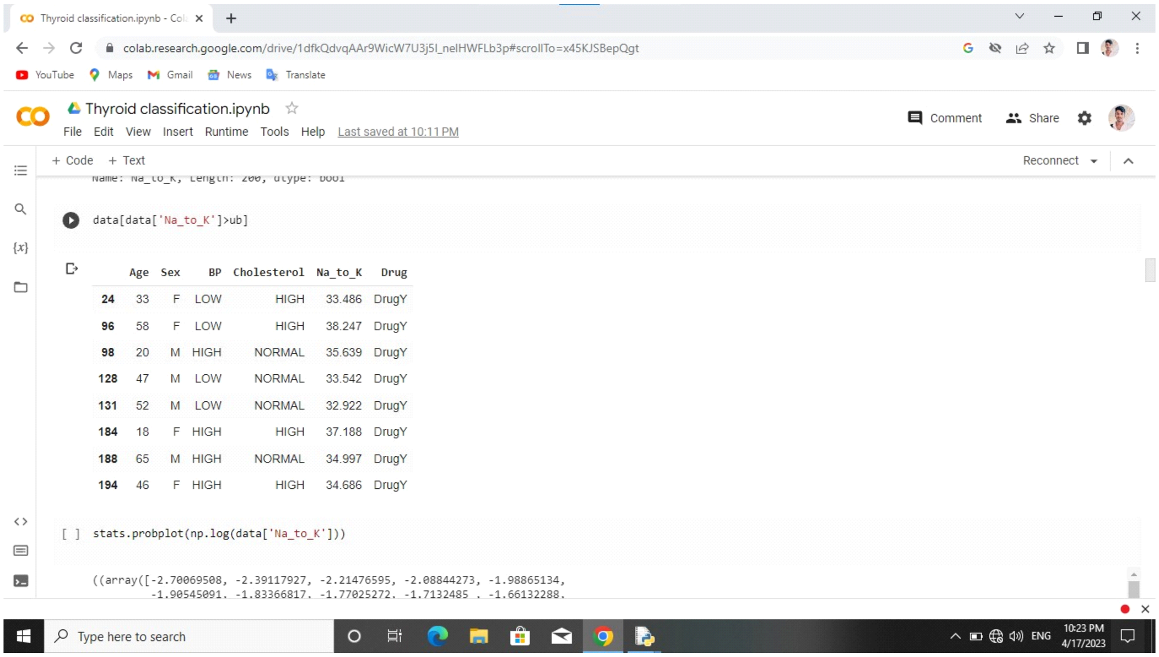


```
[ ] transform(np.log(data['Na_to_K']))
```



```
stats.probplot(np.log(data['Na_to_K']))
```

```
((array([-2.70069508, -2.39117927, -2.21476595, -2.08844273, -1.98865134,
         -1.90545091, -1.83366817, -1.77025272, -1.7132485 , -1.66132288,
         -1.61352545, -1.56915347, -1.52767212, -1.48866446, -1.45179889,
         -1.41680716, -1.38346902, -1.35160134, -1.32105008, -1.29168444,
         -1.26339229, -1.23607681, -1.20965371, -1.18404921, -1.1591983 ,
         -1.13504337, -1.11153314, -1.08862177, -1.06626803, -1.04443478,
         -1.02308837, -1.00219826, -0.9817366 , -0.96167793, -0.94199891,
         -0.9226781 , -0.90369573, -0.88503353, -0.86667459, -0.84860322,
         -0.83080482, -0.8132658 , -0.79597346, -0.77891592, -0.76208206,
         -0.74546144, -0.72904425, -0.71282124, -0.69678371, -0.68092342,
         -0.6652326 , -0.64970389, -0.63433029, -0.61910517, -0.60402223,
         -0.58907547, -0.57425918, -0.55956789, -0.5449964 , -0.53053972,
         -0.51619308, -0.50195192, -0.48781183, -0.47376862, -0.45981823,
         -0.44595676, -0.43218046, -0.41848569, -0.40486897, -0.39132691,
         -0.37785624, -0.36445379, -0.35111649, -0.33784135, -0.3246255 ,
         -0.3114661 , -0.29836042, -0.28530579, -0.2722996 , -0.25933932,
         -0.24642245, -0.23354657, -0.2207093 , -0.20790829, -0.19514126,
         -0.18240597, -0.16970019, -0.15702175, -0.1443685 , -0.13173832,
         -0.11912912, -0.10653884, -0.09396542, -0.08140684, -0.06886109,
         -0.05632616, -0.04380009, -0.03128088, -0.01876657, -0.0062552 ,
          0.0062552 ,  0.01876657,  0.03128088,  0.04380009,  0.05632616,
          0.06886109,  0.08140684,  0.09396542,  0.10653884,  0.11912912,
          0.13173832,  0.1443685 ,  0.15702175,  0.16970019,  0.18240597,
          0.19514126,  0.20790829,  0.2207093 ,  0.23354657,  0.24642245,
```



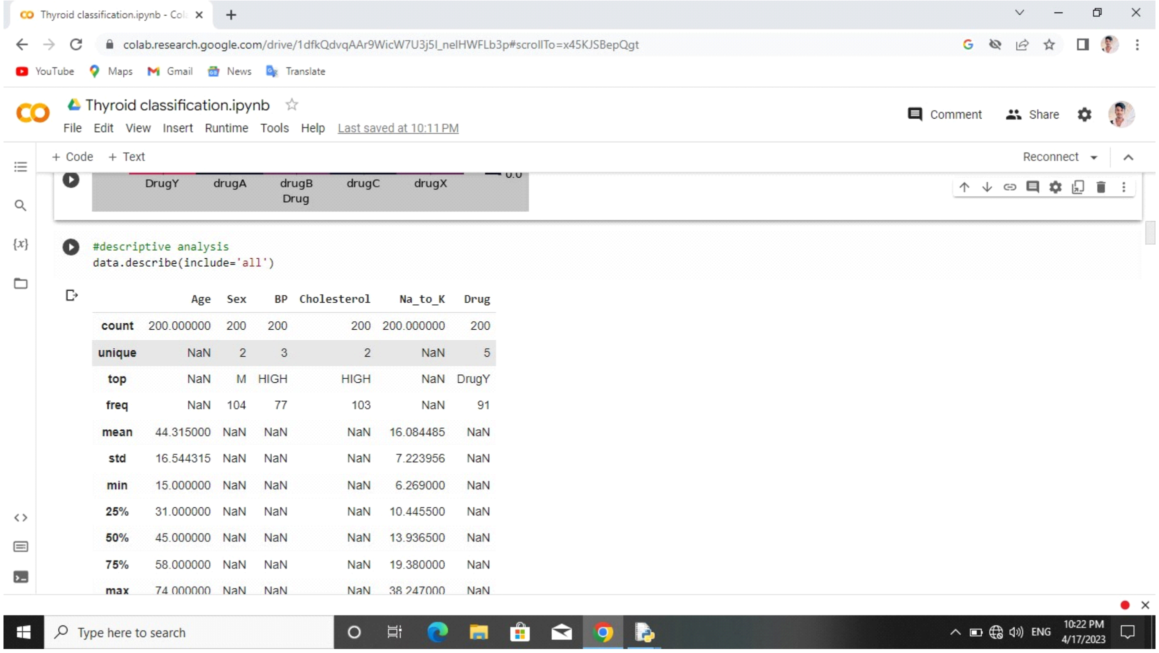
```
Name: Na_to_K, Length: 200, dtype: bool
```

```
data[data['Na_to_K']>ub]
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
24	33	F	LOW	HIGH	33.486	DrugY
96	58	F	LOW	HIGH	38.247	DrugY
98	20	M	HIGH	NORMAL	35.639	DrugY
128	47	M	LOW	NORMAL	33.542	DrugY
131	52	M	LOW	NORMAL	32.922	DrugY
184	18	F	HIGH	HIGH	37.188	DrugY
188	65	M	HIGH	NORMAL	34.997	DrugY
194	46	F	HIGH	HIGH	34.686	DrugY

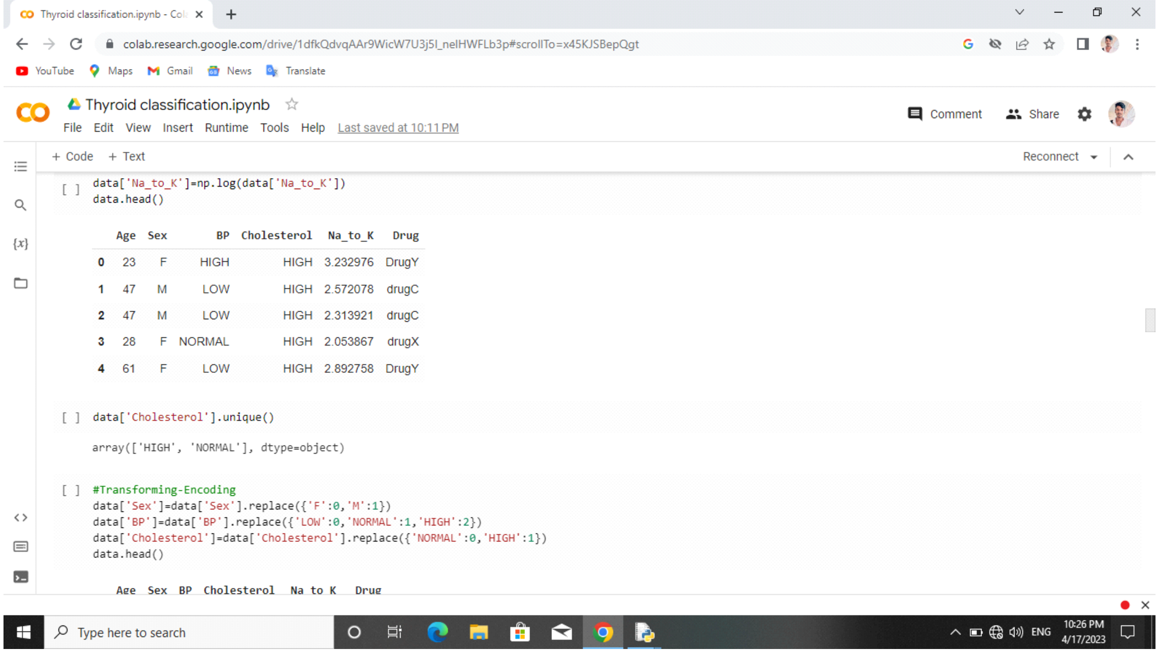
```
[ ] stats.probplot(np.log(data['Na_to_K']))
```

```
((array([-2.70069508, -2.39117927, -2.21476595, -2.08844273, -1.98865134,
         -1.90545091, -1.83366817, -1.77025272, -1.7132485 , -1.66132288,
```



```
#descriptive analysis
data.describe(include='all')
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
count	200.000000	200	200	200	200.000000	200
unique	NaN	2	3	2	NaN	5
top	NaN	M	HIGH	HIGH	NaN	DrugY
freq	NaN	104	77	103	NaN	91
mean	44.315000	NaN	NaN	NaN	16.084485	NaN
std	16.544315	NaN	NaN	NaN	7.223956	NaN
min	15.000000	NaN	NaN	NaN	6.269000	NaN
25%	31.000000	NaN	NaN	NaN	10.445500	NaN
50%	45.000000	NaN	NaN	NaN	13.936500	NaN
75%	58.000000	NaN	NaN	NaN	19.380000	NaN
max	74.000000	NaN	NaN	NaN	38.247000	NaN



```
data['Na_to_K']=np.log(data['Na_to_K'])
data.head()
```

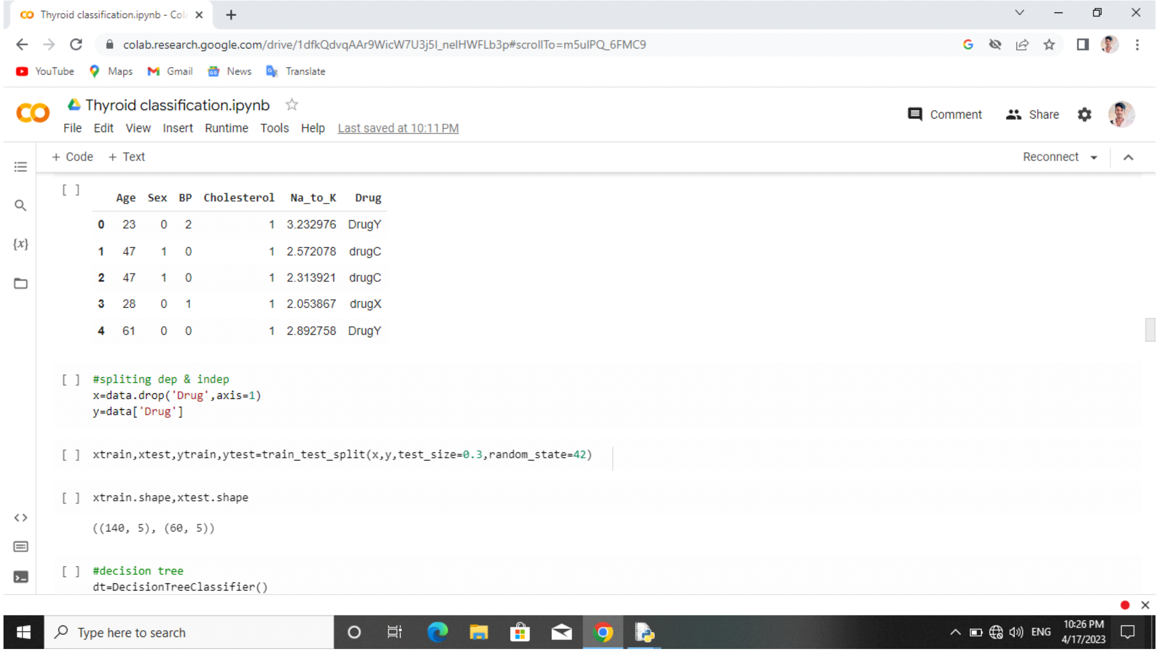
	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	3.232976	DrugY
1	47	M	LOW	HIGH	2.572078	drugC
2	47	M	LOW	HIGH	2.313921	drugC
3	28	F	NORMAL	HIGH	2.053867	drugX
4	61	F	LOW	HIGH	2.892758	DrugY

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

```
array(['HIGH', 'NORMAL'], dtype=object)
```

```
#Transforming-Encoding
data['Sex']=data['Sex'].replace({'F':0,'M':1})
data['BP']=data['BP'].replace({'LOW':0,'NORMAL':1,'HIGH':2})
data['Cholesterol']=data['Cholesterol'].replace({'NORMAL':0,'HIGH':1})
data.head()
```

Age	Sex	BP	Cholesterol	Na_to_K	Drug


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=m5uIPQ_6FMC9
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment
Share
+ Code + Text
Reconnect
Type here to search
ENG
10:26 PM
4/17/2023

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	2	1	3.232976	DrugY
1	47	1	0	1	2.572078	drugC
2	47	1	0	1	2.313921	drugC
3	28	0	1	1	2.053867	drugX
4	61	0	0	1	2.892758	DrugY

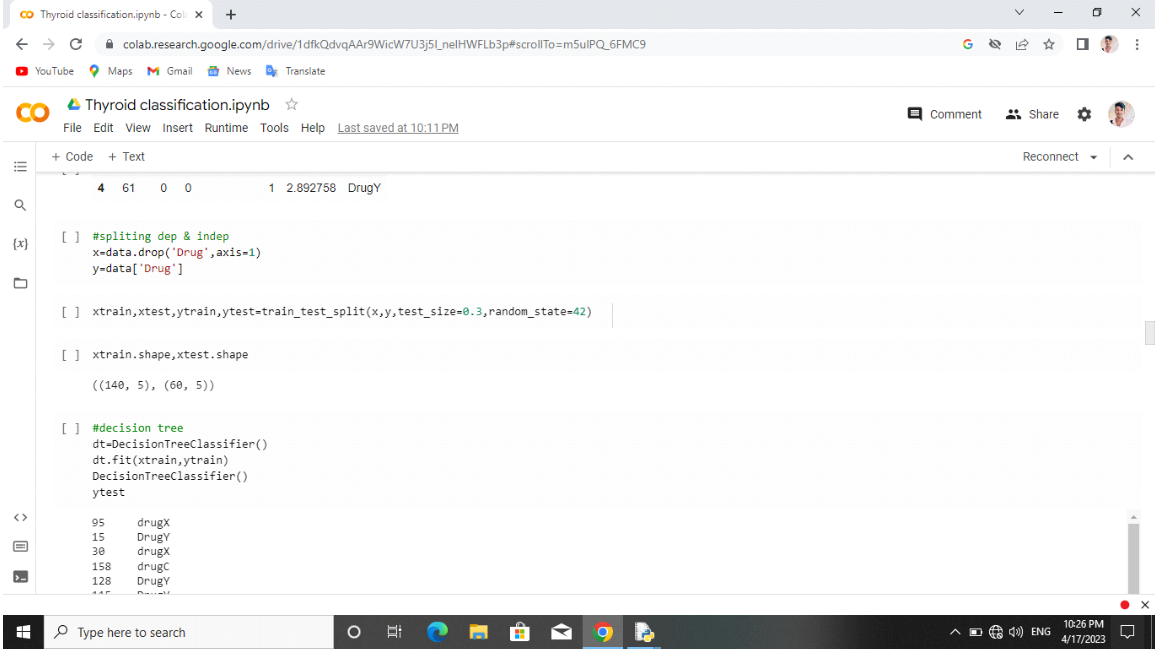
```
#spliting dep & indep
x=data.drop('Drug',axis=1)
y=data['Drug']
```

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=42)
```

```
xtrain.shape,xtest.shape
```

```
((140, 5), (60, 5))
```

```
#decision tree
dt=DecisionTreeClassifier()
```


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=m5uIPQ_6FMC9
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment
Share
+ Code + Text
Reconnect
Type here to search
ENG
10:26 PM
4/17/2023

4	61	0	0	1	2.892758	DrugY

```
#spliting dep & indep
x=data.drop('Drug',axis=1)
y=data['Drug']
```

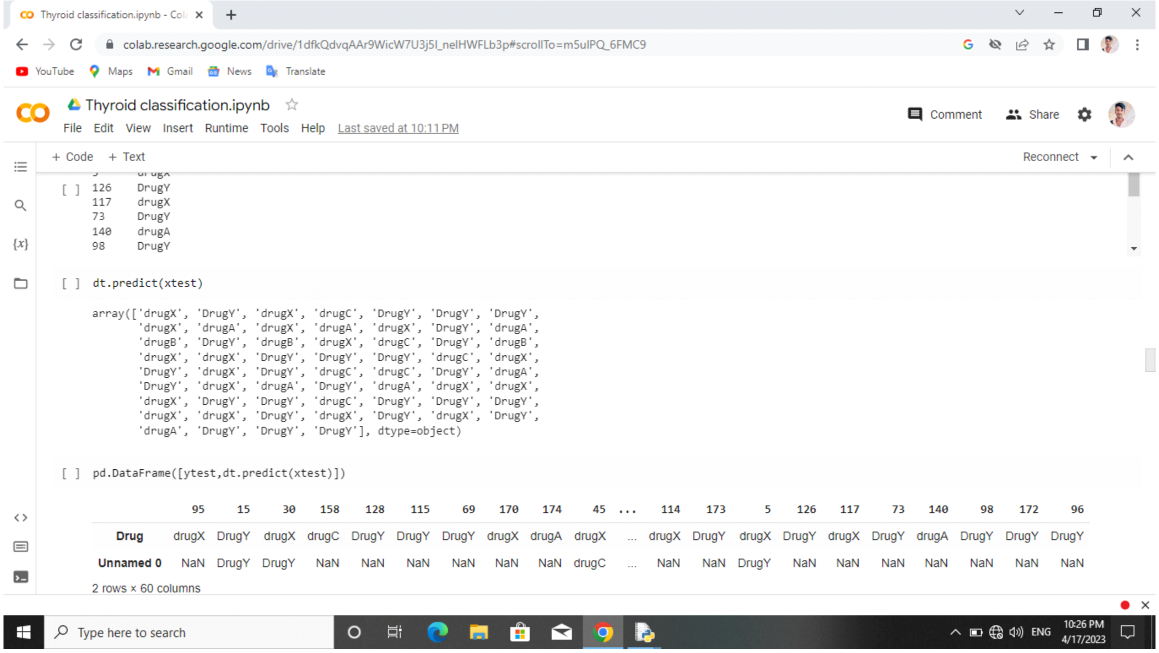
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=42)
```

```
xtrain.shape,xtest.shape
```

```
((140, 5), (60, 5))
```

```
#decision tree
dt=DecisionTreeClassifier()
dt.fit(xtrain,ytrain)
DecisionTreeClassifier()
ytest
```

```
95     drugX
15     DrugY
30     drugX
158    drugC
128    DrugY
```



```
[ ] 126     DrugY
    117     drugX
    73      DrugY
    140     drugA
    98      DrugY
```

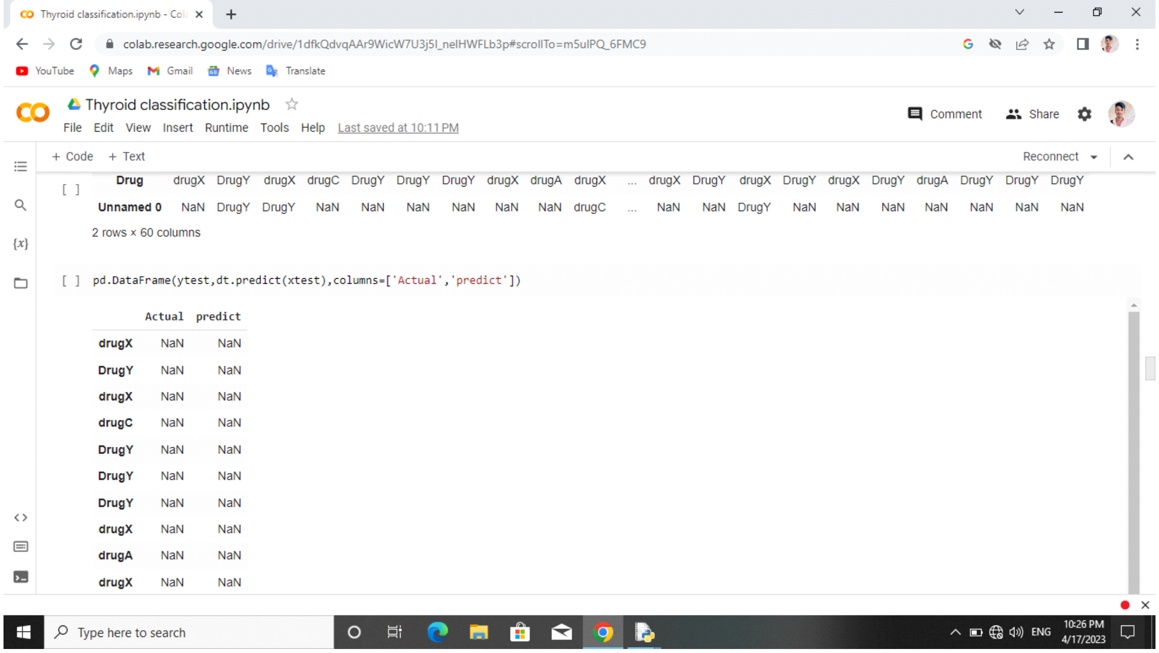
```
[ ] dt.predict(xtest)
```

```
array(['drugX', 'DrugY', 'drugX', 'drugC', 'DrugY', 'DrugY', 'DrugY',
       'drugX', 'drugA', 'drugX', 'drugA', 'drugX', 'DrugY', 'drugA',
       'drugB', 'DrugY', 'drugB', 'drugX', 'drugC', 'DrugY', 'drugB',
       'drugX', 'drugX', 'DrugY', 'DrugY', 'DrugY', 'drugC', 'drugX',
       'DrugY', 'drugX', 'DrugY', 'drugC', 'drugC', 'DrugY', 'drugA',
       'DrugY', 'drugX', 'drugA', 'DrugY', 'drugA', 'drugX', 'drugX',
       'drugX', 'DrugY', 'DrugY', 'drugC', 'DrugY', 'DrugY', 'DrugY',
       'drugX', 'drugX', 'DrugY', 'drugX', 'DrugY', 'drugX', 'DrugY',
       'drugA', 'DrugY', 'DrugY', 'DrugY'], dtype=object)
```

```
[ ] pd.DataFrame([ytest,dt.predict(xtest)])
```

	95	15	30	158	128	115	69	170	174	45	...	114	173	5	126	117	73	140	98	172	96
Drug	drugX	DrugY	drugX	drugC	DrugY	DrugY	DrugY	drugX	drugA	drugX	...	drugX	DrugY	drugX	DrugY	drugX	DrugY	drugA	DrugY	DrugY	DrugY
Unnamed 0	NaN	DrugY	DrugY	NaN	NaN	NaN	NaN	NaN	NaN	drugC	...	NaN	NaN	DrugY	NaN	NaN	NaN	NaN	NaN	NaN	NaN

2 rows × 60 columns

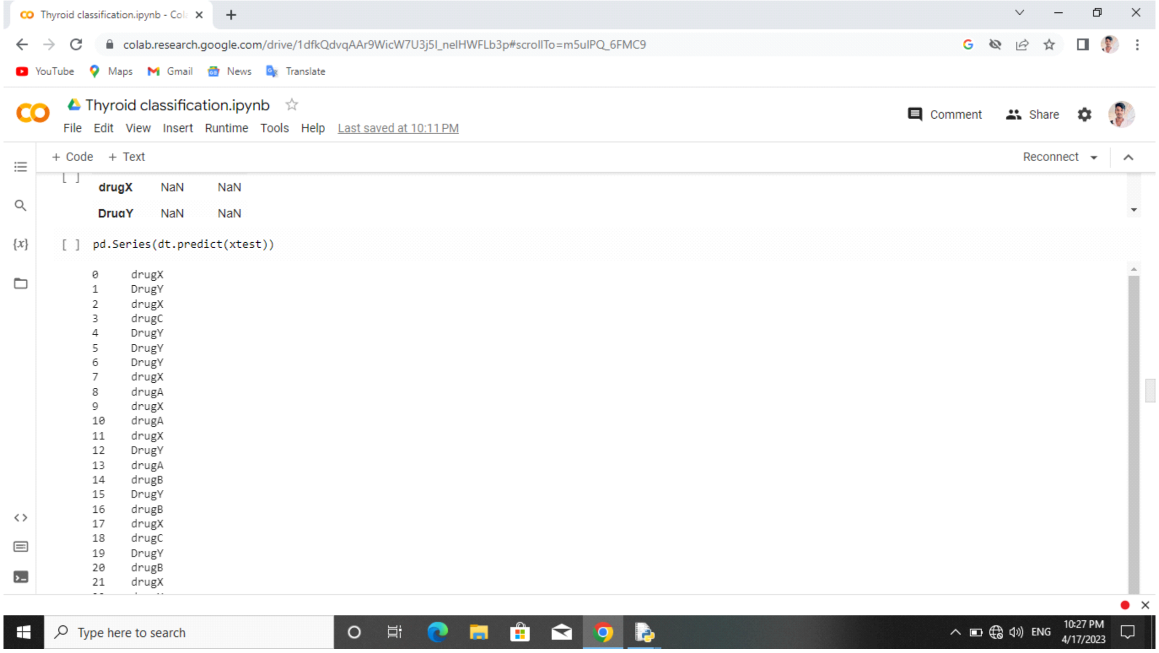


Drug	drugX	DrugY	drugX	drugC	DrugY	DrugY	DrugY	drugX	drugA	drugX	...	drugX	DrugY	drugX	DrugY	drugX	DrugY	drugA	DrugY	DrugY	DrugY
Unnamed 0	NaN	DrugY	DrugY	NaN	NaN	NaN	NaN	NaN	NaN	drugC	...	NaN	NaN	DrugY	NaN	NaN	NaN	NaN	NaN	NaN	NaN

2 rows × 60 columns

```
[ ] pd.DataFrame(ytest,dt.predict(xtest),columns=['Actual','predict'])
```

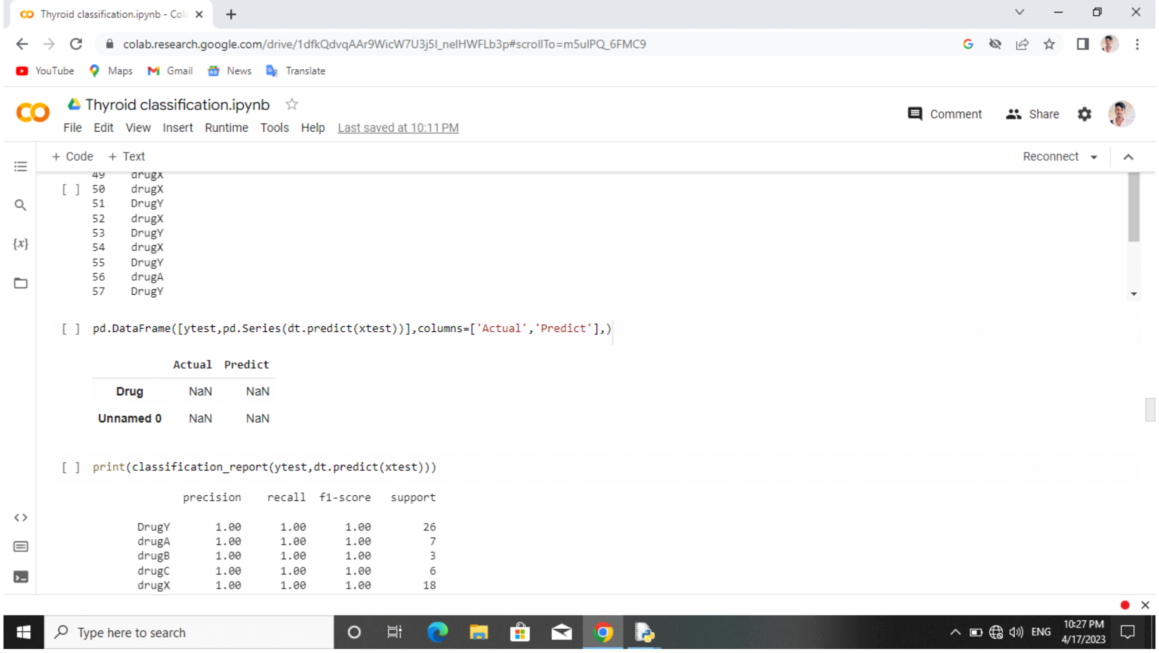
	Actual	predict
drugX	NaN	NaN
DrugY	NaN	NaN
drugX	NaN	NaN
drugC	NaN	NaN
DrugY	NaN	NaN
DrugY	NaN	NaN
DrugY	NaN	NaN
drugX	NaN	NaN
drugA	NaN	NaN
drugX	NaN	NaN



drugX	NaN	NaN
DrugY	NaN	NaN

```
pd.Series(dt.predict(xtest))
```

```
0     drugX
1     DrugY
2     drugX
3     drugC
4     DrugY
5     DrugY
6     DrugY
7     drugX
8     drugA
9     drugX
10    drugA
11    drugX
12    DrugY
13    drugA
14    drugB
15    DrugY
16    drugB
17    drugX
18    drugC
19    DrugY
20    drugB
21    drugX
```



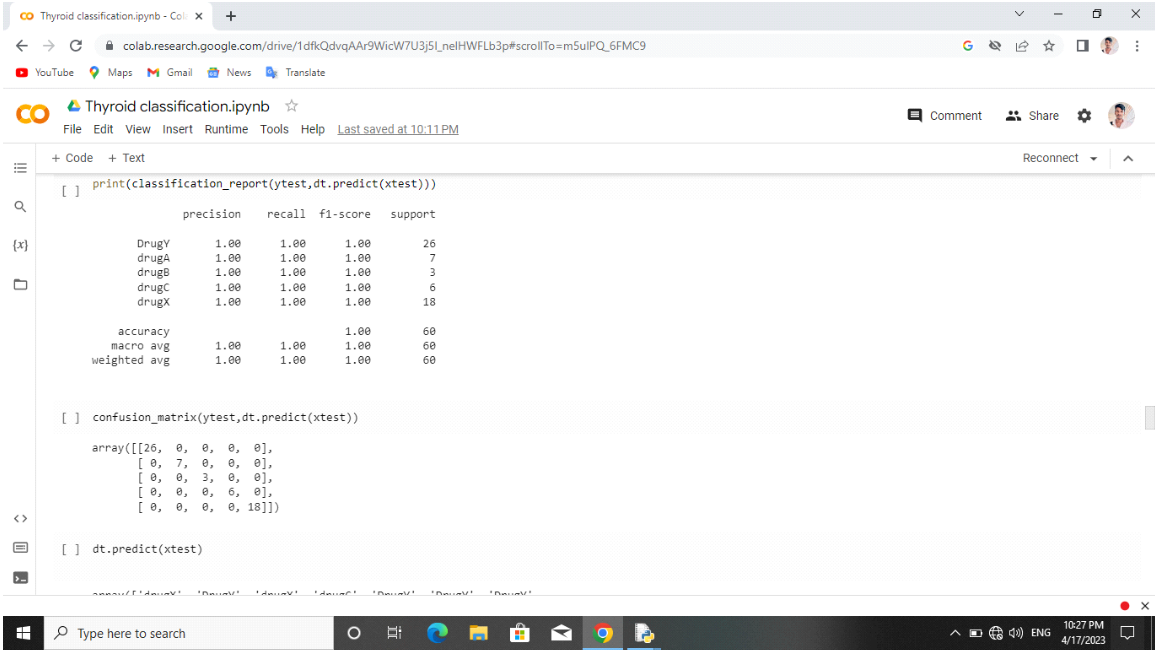
```
49    drugX
50    drugX
51    DrugY
52    drugX
53    DrugY
54    drugX
55    DrugY
56    drugA
57    DrugY
```

```
pd.DataFrame([ytest,pd.Series(dt.predict(xtest))],columns=['Actual','Predict'],)
```

	Actual	Predict
Drug	NaN	NaN
Unnamed 0	NaN	NaN

```
print(classification_report(ytest,dt.predict(xtest)))
```

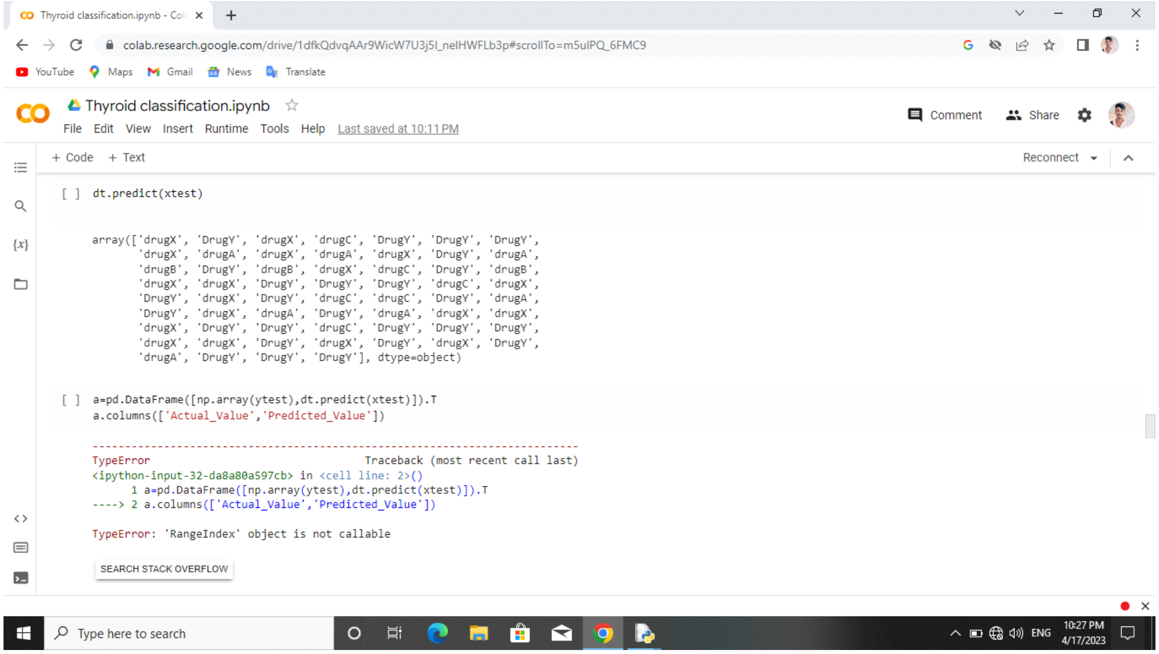
```
              precision    recall  f1-score   support

       DrugY       1.00      1.00      1.00        26
       drugA       1.00      1.00      1.00         7
       drugB       1.00      1.00      1.00         3
       drugC       1.00      1.00      1.00         6
       drugX       1.00      1.00      1.00        18
```


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=m5uIPQ_6FMC9
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment
Share
+ Code + Text
Reconnect
Type here to search
ENG
10:27 PM
4/17/2023

```
[ ] print(classification_report(ytest,dt.predict(xtest)))

                  precision    recall  f1-score   support

           DrugY       1.00      1.00      1.00        26
           drugA       1.00      1.00      1.00         7
           drugB       1.00      1.00      1.00         3
           drugC       1.00      1.00      1.00         6
           drugX       1.00      1.00      1.00        18

        accuracy                           1.00        60
       macro avg       1.00      1.00      1.00        60
    weighted avg       1.00      1.00      1.00        60
```

```
[ ] confusion_matrix(ytest,dt.predict(xtest))

    array([[26,  0,  0,  0,  0],
           [ 0,  7,  0,  0,  0],
           [ 0,  0,  3,  0,  0],
           [ 0,  0,  0,  6,  0],
           [ 0,  0,  0,  0, 18]])
```

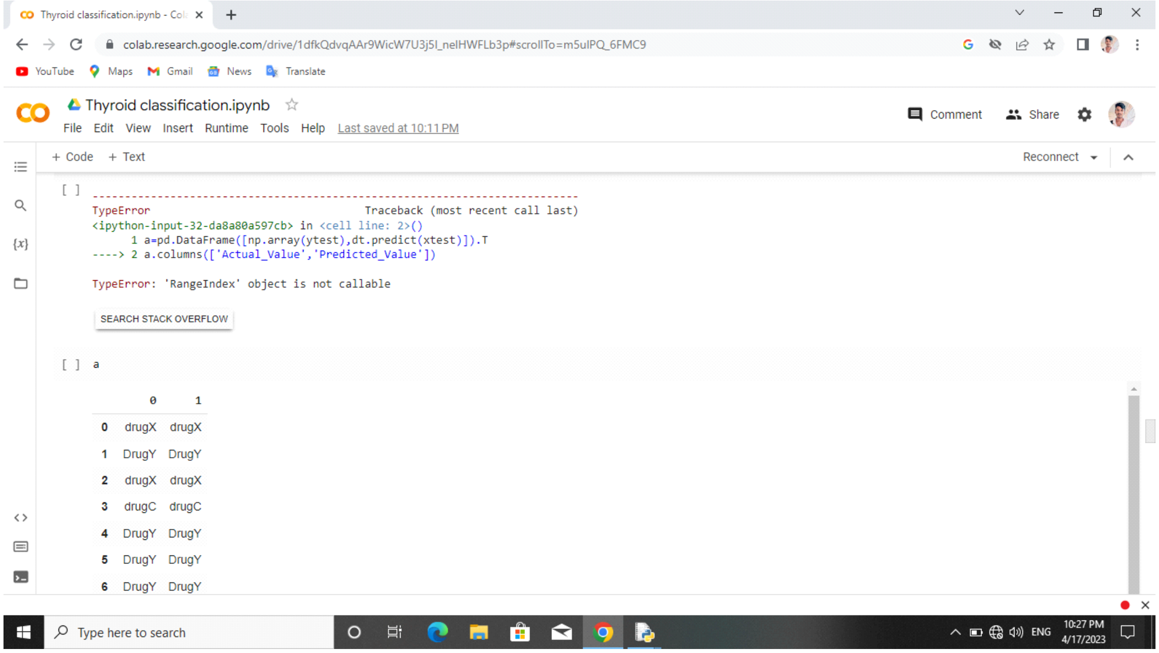
```
[ ] dt.predict(xtest)

    array(['drugX', 'DrugY', 'drugX', 'drugC', 'DrugY', 'DrugY', 'DrugY',
```


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=m5uIPQ_6FMC9
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment
Share
+ Code + Text
Reconnect
Type here to search
ENG
10:27 PM
4/17/2023

```
[ ] dt.predict(xtest)


    array(['drugX', 'DrugY', 'drugX', 'drugC', 'DrugY', 'DrugY', 'DrugY',
           'drugX', 'drugA', 'drugX', 'drugA', 'drugX', 'DrugY', 'drugA',
           'drugB', 'DrugY', 'drugB', 'drugX', 'drugC', 'DrugY', 'drugB',
           'drugX', 'drugX', 'DrugY', 'DrugY', 'DrugY', 'drugC', 'drugX',
           'DrugY', 'drugX', 'DrugY', 'drugC', 'drugC', 'DrugY', 'drugA',
           'DrugY', 'drugX', 'drugA', 'DrugY', 'drugA', 'drugX', 'drugX',
           'drugX', 'DrugY', 'DrugY', 'drugC', 'DrugY', 'DrugY', 'DrugY',
           'drugX', 'drugX', 'DrugY', 'drugX', 'DrugY', 'drugX', 'DrugY',
           'drugA', 'DrugY', 'DrugY', 'DrugY'], dtype=object)
```

```
[ ] a=pd.DataFrame([np.array(ytest),dt.predict(xtest)]).T
    a.columns(['Actual_Value','Predicted_Value'])

    ---------------------------------------------------------------------------
    TypeError                                 Traceback (most recent call last)
    <ipython-input-32-da8a80a597cb> in <cell line: 2>()
          1 a=pd.DataFrame([np.array(ytest),dt.predict(xtest)]).T
    ----> 2 a.columns(['Actual_Value','Predicted_Value'])

    TypeError: 'RangeIndex' object is not callable
```

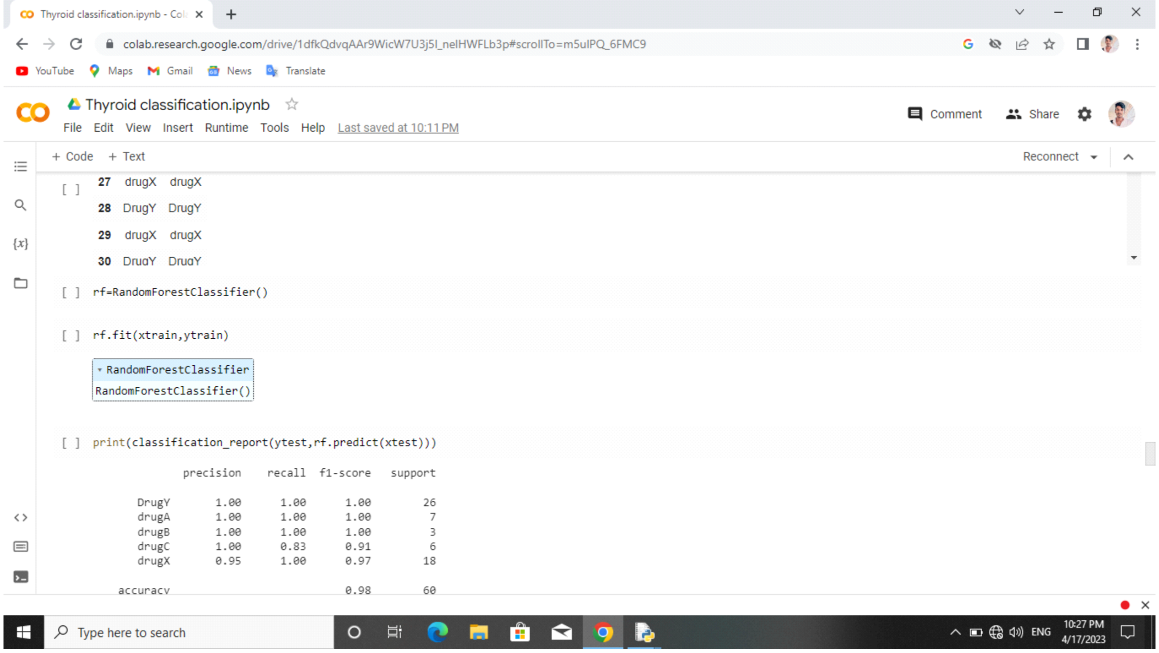
SEARCH STACK OVERFLOW



```
[ ] ---------------------------------------------------------------------------
    TypeError                                 Traceback (most recent call last)
    <ipython-input-32-da8a80a597cb> in <cell line: 2>()
          1 a=pd.DataFrame([np.array(ytest),dt.predict(xtest)]).T
    ----> 2 a.columns(['Actual_Value','Predicted_Value'])

    TypeError: 'RangeIndex' object is not callable
```

```
[ ] a
```

```
         0      1
0    drugX  drugX
1    DrugY  DrugY
2    drugX  drugX
3    drugC  drugC
4    DrugY  DrugY
5    DrugY  DrugY
6    DrugY  DrugY
```



```
27   drugX  drugX
28   DrugY  DrugY
29   drugX  drugX
30   DrugY  DrugY
```

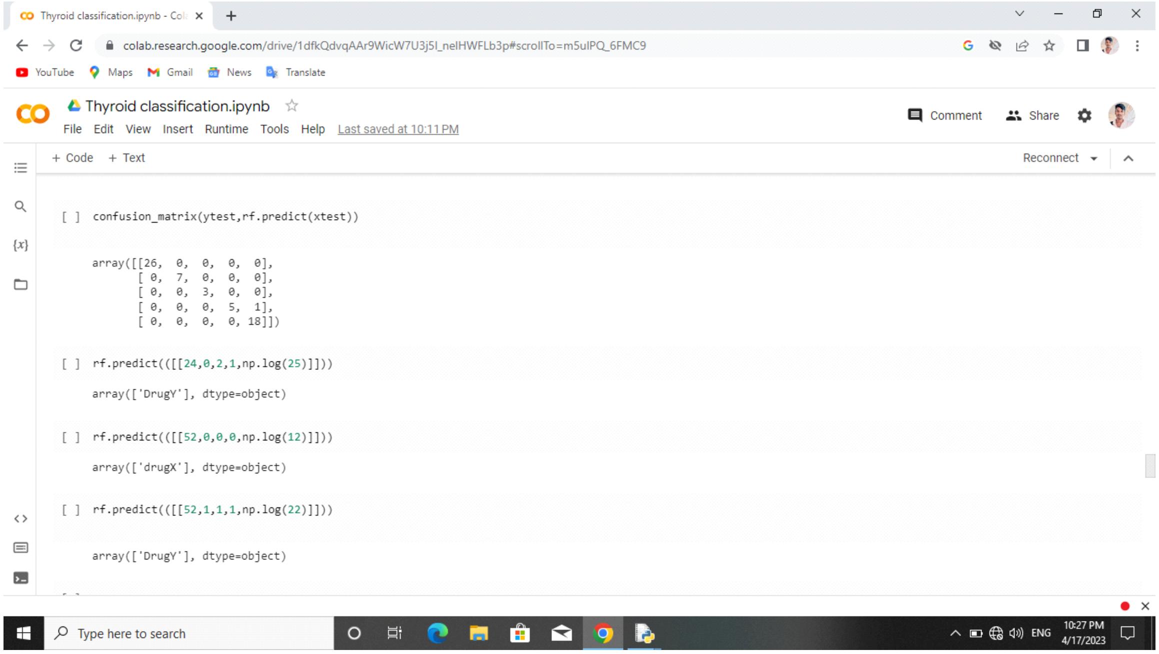
```
[ ] rf=RandomForestClassifier()
```

```
[ ] rf.fit(xtrain,ytrain)
```

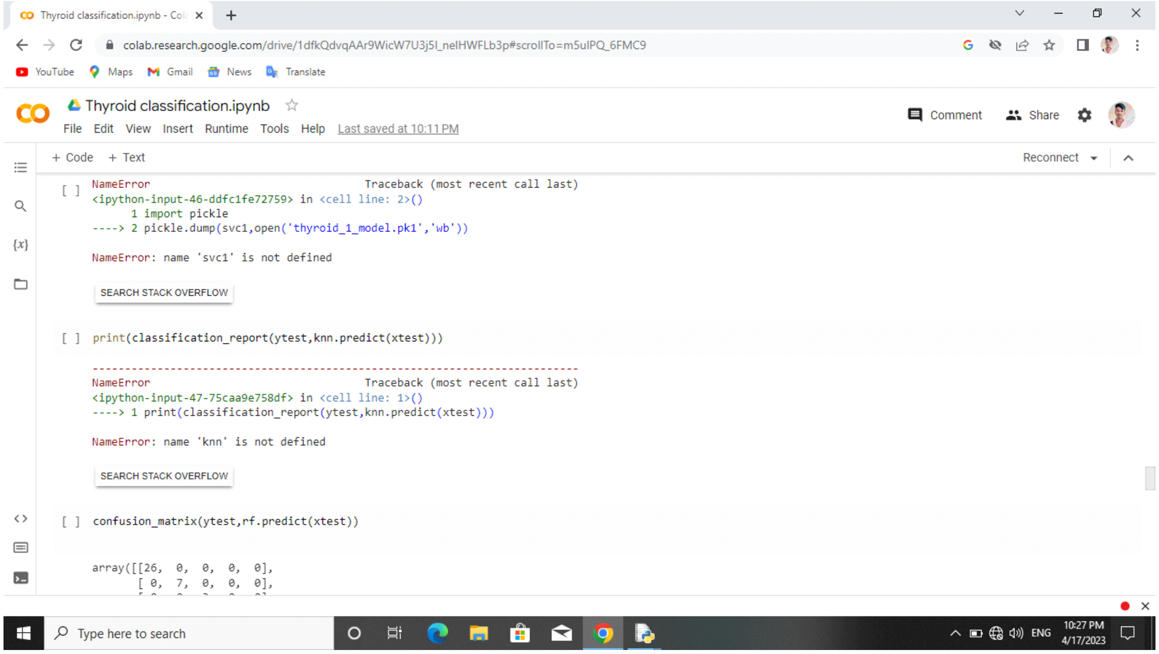
```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
[ ] print(classification_report(ytest,rf.predict(xtest)))
```

```
              precision    recall  f1-score   support

       DrugY       1.00      1.00      1.00        26
       drugA       1.00      1.00      1.00         7
       drugB       1.00      1.00      1.00         3
       drugC       1.00      0.83      0.91         6
       drugX       0.95      1.00      0.97        18

    accuracy                           0.98        60
```

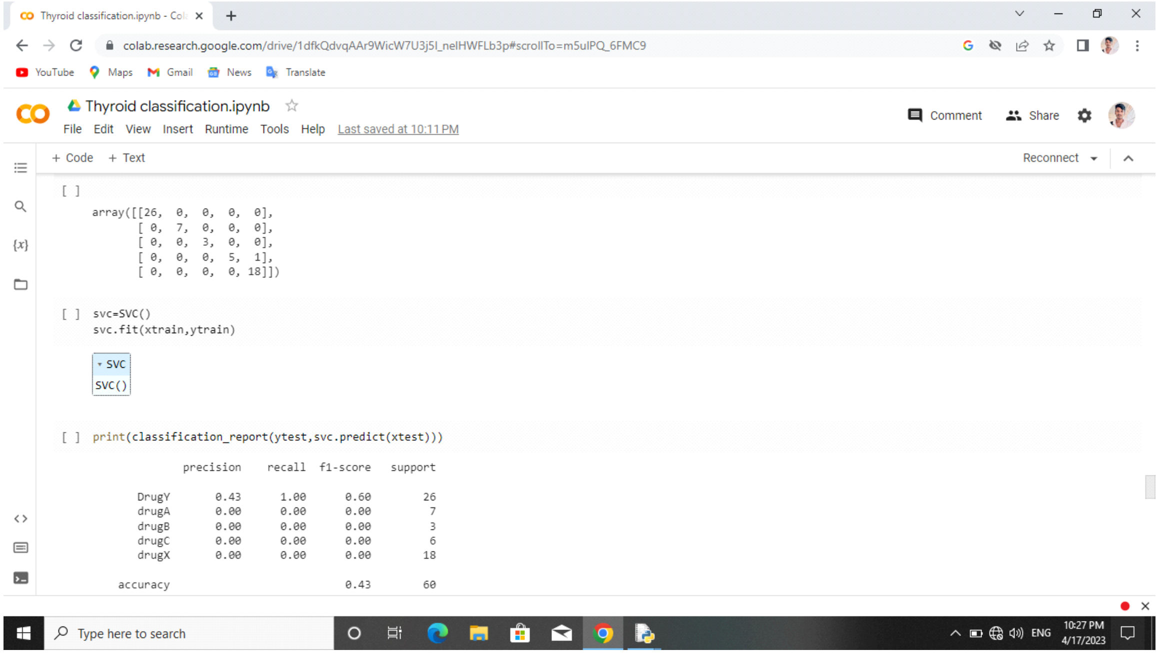


```
[ ] confusion_matrix(ytest,rf.predict(xtest))

    array([[26,  0,  0,  0,  0],
           [ 0,  7,  0,  0,  0],
           [ 0,  0,  3,  0,  0],
           [ 0,  0,  0,  5,  1],
           [ 0,  0,  0,  0, 18]])

[ ] rf.predict(([[24,0,2,1,np.log(25)]]))

    array(['DrugY'], dtype=object)

[ ] rf.predict(([[52,0,0,0,np.log(12)]]))

    array(['drugX'], dtype=object)

[ ] rf.predict(([[52,1,1,1,np.log(22)]]))

    array(['DrugY'], dtype=object)
```



```
[ ] NameError                                 Traceback (most recent call last)
    <ipython-input-46-ddfc1fe72759> in <cell line: 2>()
          1 import pickle
    ----> 2 pickle.dump(svc1,open('thyroid_1_model.pk1','wb'))

    NameError: name 'svc1' is not defined

[ ] print(classification_report(ytest,knn.predict(xtest)))

    ---------------------------------------------------------------------------
    NameError                                 Traceback (most recent call last)
    <ipython-input-47-75caa9e758df> in <cell line: 1>()
    ----> 1 print(classification_report(ytest,knn.predict(xtest)))

    NameError: name 'knn' is not defined

[ ] confusion_matrix(ytest,rf.predict(xtest))

    array([[26,  0,  0,  0,  0],
           [ 0,  7,  0,  0,  0],
```

SVC method ;



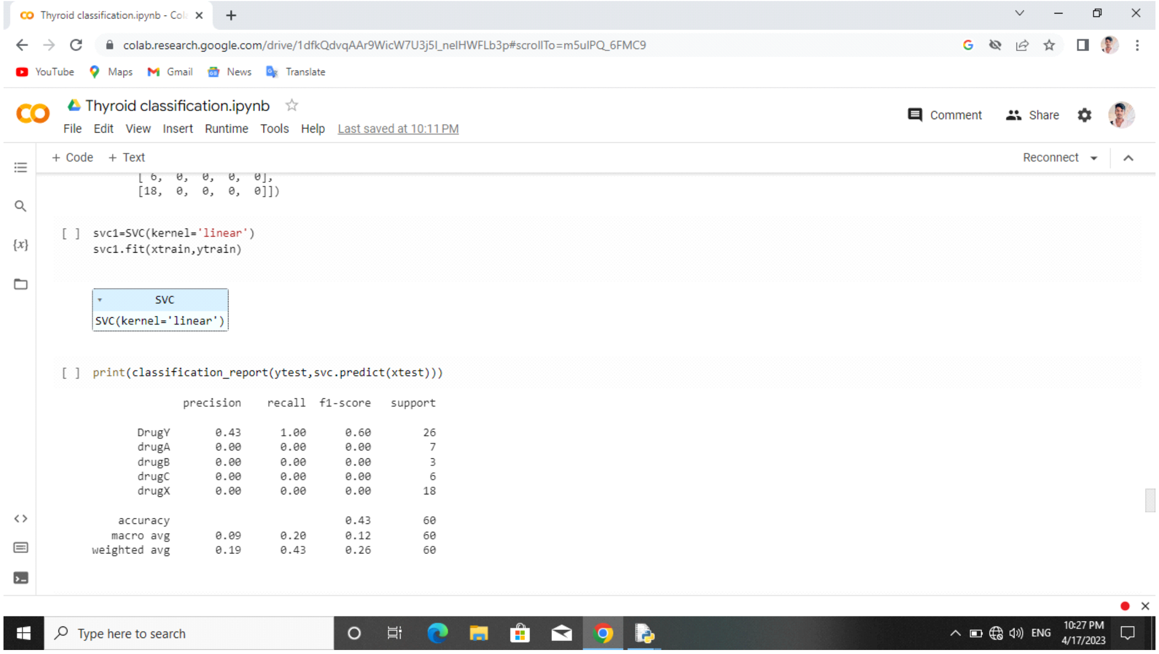
```
array([[26,  0,  0,  0,  0],
       [ 0,  7,  0,  0,  0],
       [ 0,  0,  3,  0,  0],
       [ 0,  0,  0,  5,  1],
       [ 0,  0,  0,  0, 18]])
```

```
svc=SVC()
svc.fit(xtrain,ytrain)
```

```
SVC()
```

```
print(classification_report(ytest,svc.predict(xtest)))
```

```
              precision    recall  f1-score   support

       DrugY       0.43      1.00      0.60        26
       drugA       0.00      0.00      0.00         7
       drugB       0.00      0.00      0.00         3
       drugC       0.00      0.00      0.00         6
       drugX       0.00      0.00      0.00        18

    accuracy                           0.43        60
```



```
       [ 6,  0,  0,  0,  0],
       [18,  0,  0,  0,  0]])
```

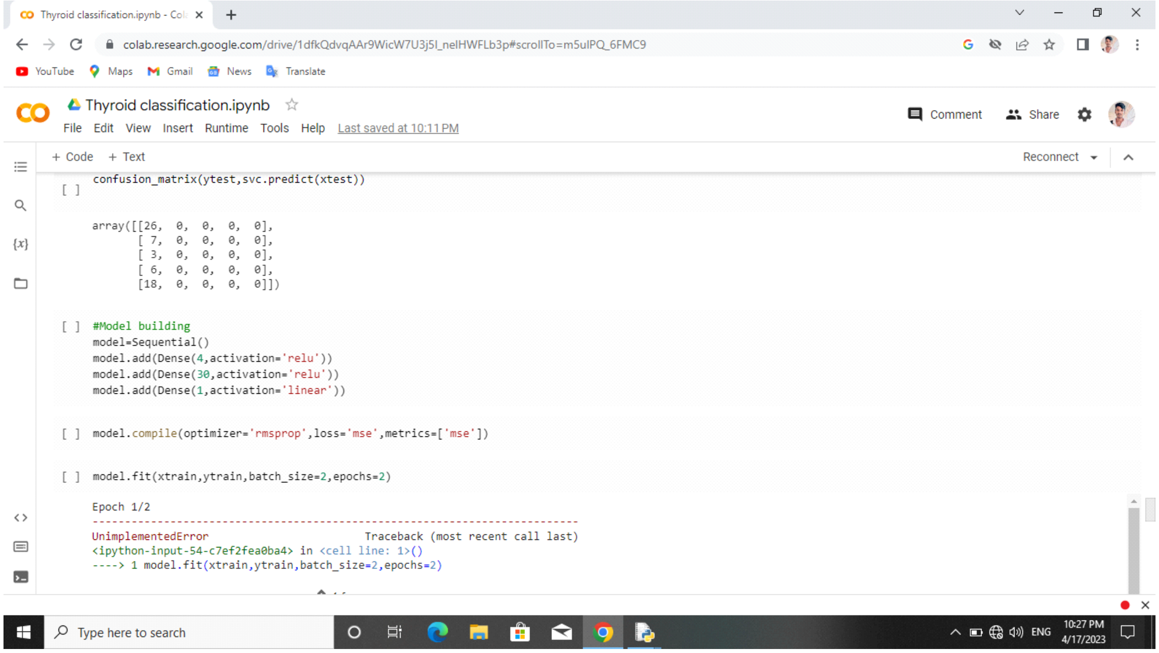
```
svc1=SVC(kernel='linear')
svc1.fit(xtrain,ytrain)
```

```
SVC(kernel='linear')
```

```
print(classification_report(ytest,svc.predict(xtest)))
```

```
              precision    recall  f1-score   support

       DrugY       0.43      1.00      0.60        26
       drugA       0.00      0.00      0.00         7
       drugB       0.00      0.00      0.00         3
       drugC       0.00      0.00      0.00         6
       drugX       0.00      0.00      0.00        18

    accuracy                           0.43        60
   macro avg       0.09      0.20      0.12        60
weighted avg       0.19      0.43      0.26        60
```

Model Building:

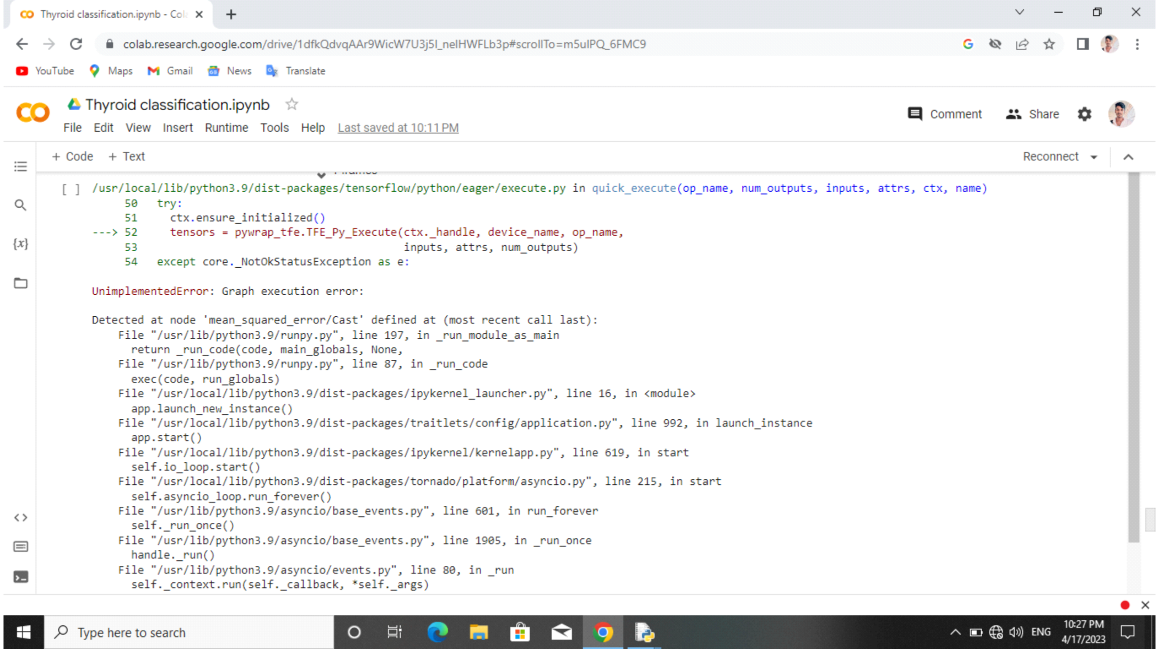

Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=m5uIPQ_6FMC9
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help
Last saved at 10:11 PM
Comment
Share
+ Code
+ Text
Reconnect
Type here to search
ENG
10:27 PM
4/17/2023

```
[ ] confusion_matrix(ytest,svc.predict(xtest))

    array([[26,  0,  0,  0,  0],
           [ 7,  0,  0,  0,  0],
           [ 3,  0,  0,  0,  0],
           [ 6,  0,  0,  0,  0],
           [18,  0,  0,  0,  0]])
```

```
[ ] #Model building
    model=Sequential()
    model.add(Dense(4,activation='relu'))
    model.add(Dense(30,activation='relu'))
    model.add(Dense(1,activation='linear'))
```

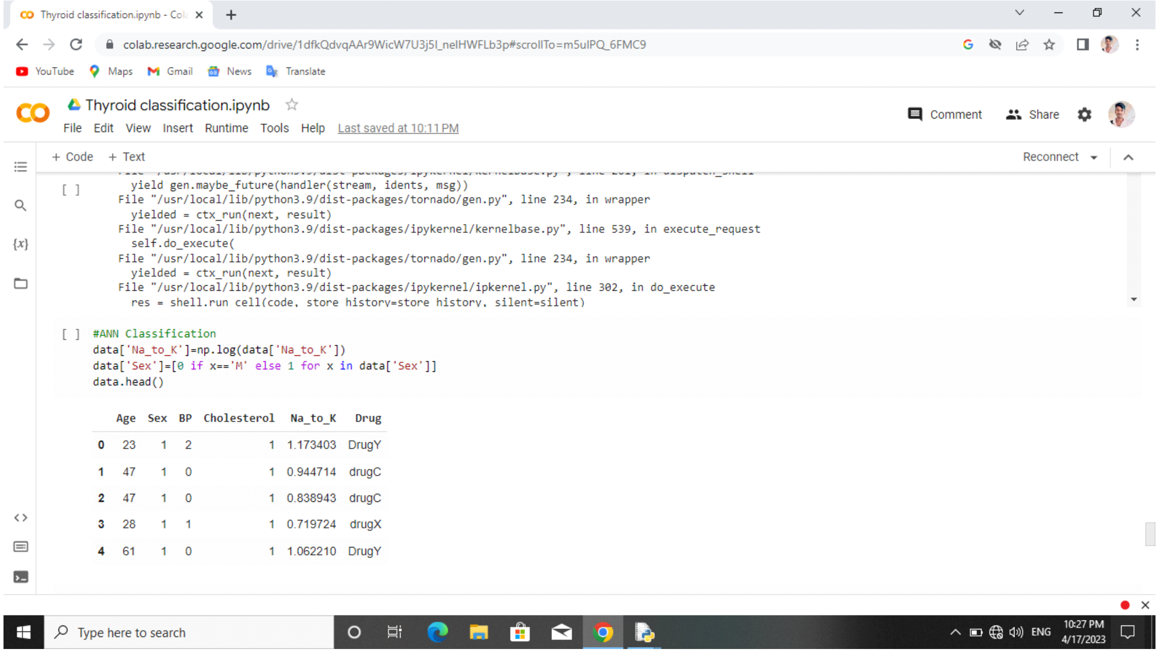
```
[ ] model.compile(optimizer='rmsprop',loss='mse',metrics=['mse'])
```

```
[ ] model.fit(xtrain,ytrain,batch_size=2,epochs=2)

    Epoch 1/2
    ---------------------------------------------------------------------------
    UnimplementedError                        Traceback (most recent call last)
    <ipython-input-54-c7ef2fea0ba4> in <cell line: 1>()
    ----> 1 model.fit(xtrain,ytrain,batch_size=2,epochs=2)
```


Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=m5uIPQ_6FMC9
YouTube
Maps
Gmail
News
Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help
Last saved at 10:11 PM
Comment
Share
+ Code
+ Text
Reconnect
Type here to search
ENG
10:27 PM
4/17/2023

```
[ ] /usr/local/lib/python3.9/dist-packages/tensorflow/python/eager/execute.py in quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
         50   try:
         51     ctx.ensure_initialized()
    ---> 52     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
         53                                         inputs, attrs, num_outputs)
         54   except core._NotOkStatusException as e:

    UnimplementedError: Graph execution error:

    Detected at node 'mean_squared_error/Cast' defined at (most recent call last):
        File "/usr/lib/python3.9/runpy.py", line 197, in _run_module_as_main
          return _run_code(code, main_globals, None,
        File "/usr/lib/python3.9/runpy.py", line 87, in _run_code
          exec(code, run_globals)
        File "/usr/local/lib/python3.9/dist-packages/ipykernel_launcher.py", line 16, in <module>
          app.launch_new_instance()
        File "/usr/local/lib/python3.9/dist-packages/traitlets/config/application.py", line 992, in launch_instance
          app.start()
        File "/usr/local/lib/python3.9/dist-packages/ipykernel/kernelapp.py", line 619, in start
          self.io_loop.start()
        File "/usr/local/lib/python3.9/dist-packages/tornado/platform/asyncio.py", line 215, in start
          self.asyncio_loop.run_forever()
        File "/usr/lib/python3.9/asyncio/base_events.py", line 601, in run_forever
          self._run_once()
        File "/usr/lib/python3.9/asyncio/base_events.py", line 1905, in _run_once
          handle._run()
        File "/usr/lib/python3.9/asyncio/events.py", line 80, in _run
          self._context.run(self._callback, *self._args)
```

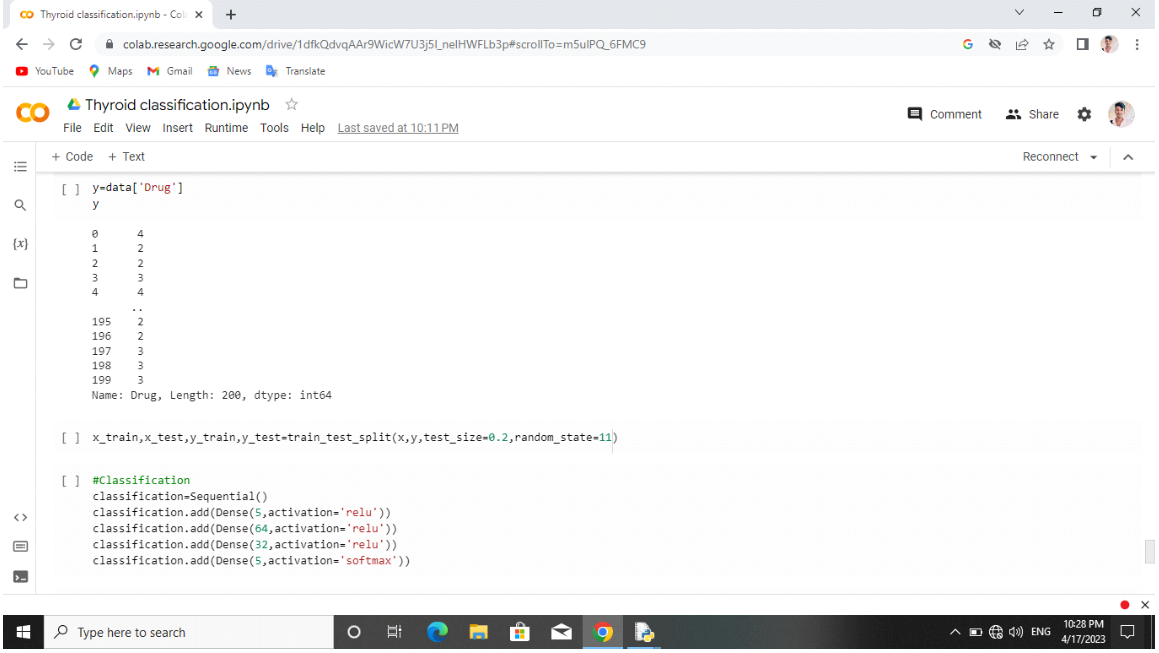
KNN.classification:



```
    yield gen.maybe_future(handler(stream, idents, msg))
  File "/usr/local/lib/python3.9/dist-packages/tornado/gen.py", line 234, in wrapper
    yielded = ctx_run(next, result)
  File "/usr/local/lib/python3.9/dist-packages/ipykernel/kernelbase.py", line 539, in execute_request
    self.do_execute(
  File "/usr/local/lib/python3.9/dist-packages/tornado/gen.py", line 234, in wrapper
    yielded = ctx_run(next, result)
  File "/usr/local/lib/python3.9/dist-packages/ipykernel/ipkernel.py", line 302, in do_execute
    res = shell.run cell(code, store history=store history, silent=silent)
```

```
#ANN Classification
data['Na_to_K']=np.log(data['Na_to_K'])
data['Sex']=[0 if x=='M' else 1 for x in data['Sex']]
data.head()
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	1	2	1	1.173403	DrugY
1	47	1	0	1	0.944714	drugC
2	47	1	0	1	0.838943	drugC
3	28	1	1	1	0.719724	drugX
4	61	1	0	1	1.062210	DrugY

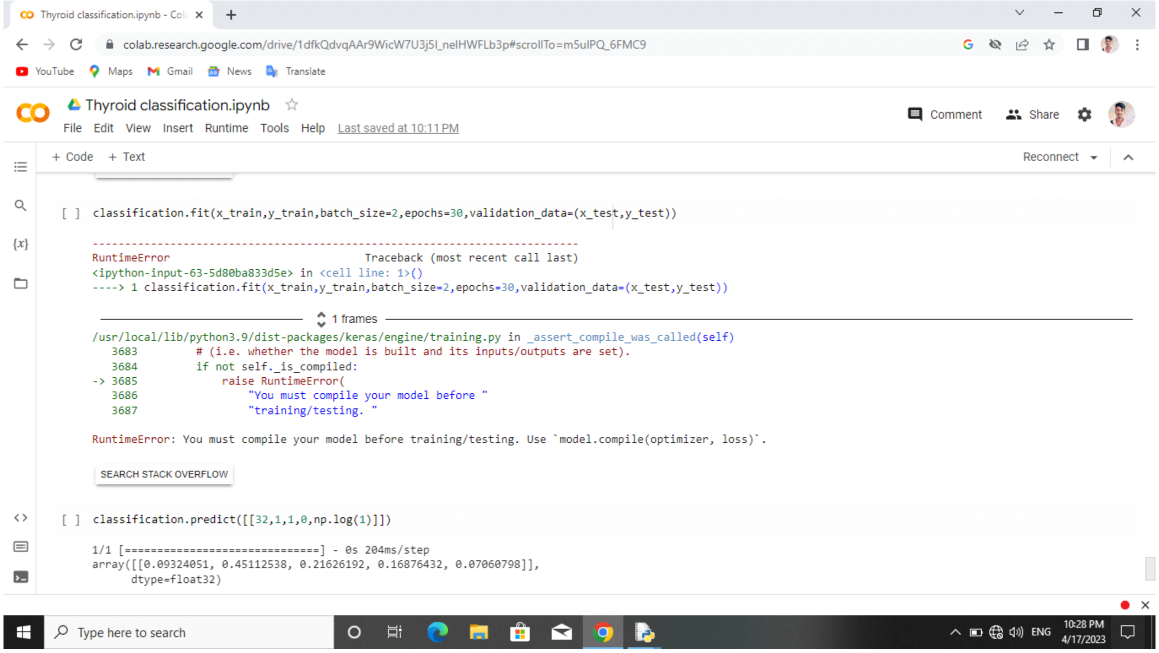


```
y=data['Drug']
y
```

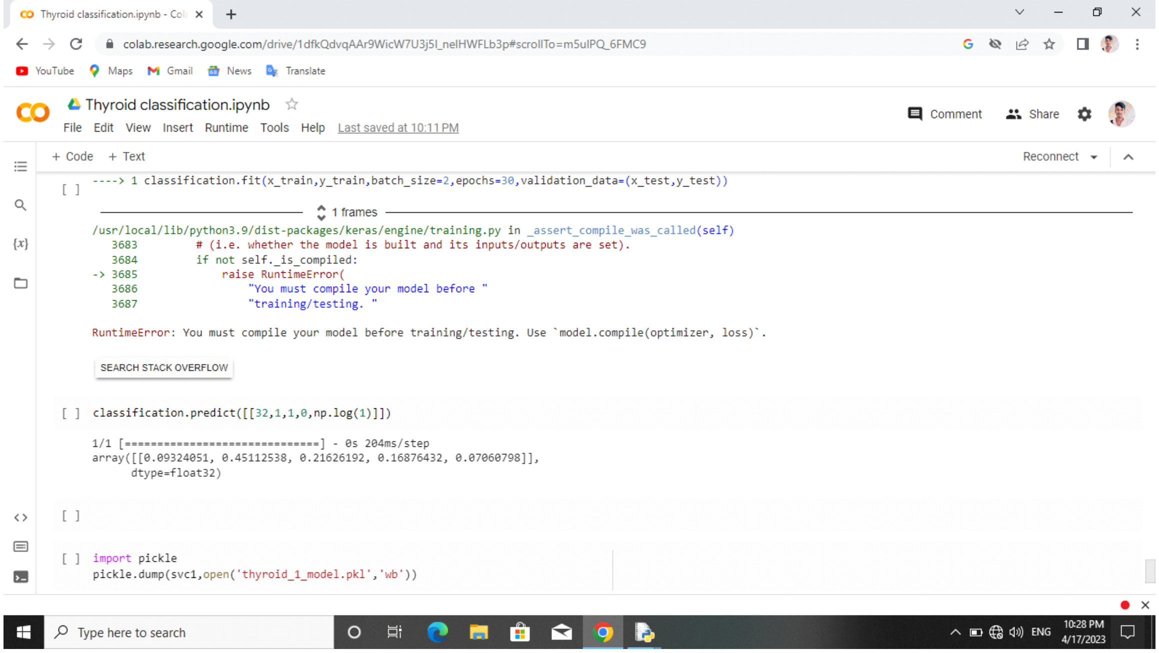
```
0      4
1      2
2      2
3      3
4      4
      ..
195    2
196    2
197    3
198    3
199    3
Name: Drug, Length: 200, dtype: int64
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=11)
```

```
#Classification
classification=Sequential()
classification.add(Dense(5,activation='relu'))
classification.add(Dense(64,activation='relu'))
classification.add(Dense(32,activation='relu'))
classification.add(Dense(5,activation='softmax'))
```

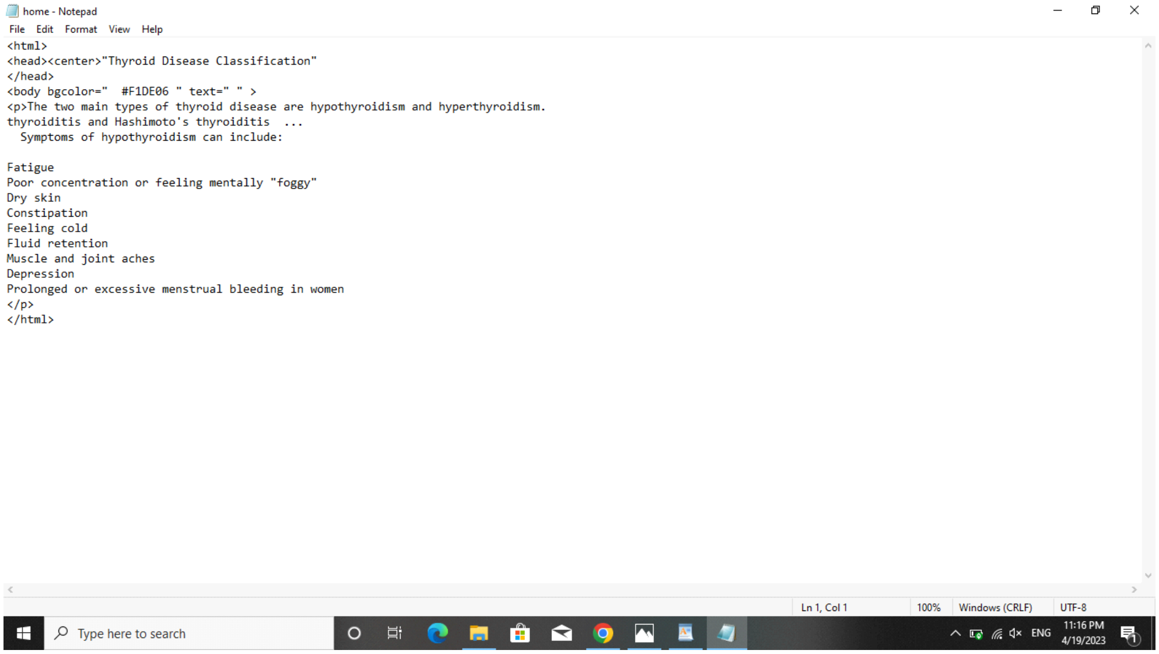

Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=m5uIPQ_6FMC9
YouTube Maps Gmail News Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment Share
+ Code + Text
Reconnect
Type here to search
ENG
10:28 PM
4/17/2023

```
[ ] classification.fit(x_train,y_train,batch_size=2,epochs=30,validation_data=(x_test,y_test))

    ---------------------------------------------------------------------------
    RuntimeError                              Traceback (most recent call last)
    <ipython-input-63-5d80ba833d5e> in <cell line: 1>()
    ----> 1 classification.fit(x_train,y_train,batch_size=2,epochs=30,validation_data=(x_test,y_test))

    ─────────────────────── 1 frames ───────────────────────
    /usr/local/lib/python3.9/dist-packages/keras/engine/training.py in _assert_compile_was_called(self)
       3683         # (i.e. whether the model is built and its inputs/outputs are set).
       3684         if not self._is_compiled:
    -> 3685             raise RuntimeError(
       3686                 "You must compile your model before "
       3687                 "training/testing. "

    RuntimeError: You must compile your model before training/testing. Use `model.compile(optimizer, loss)`.

    SEARCH STACK OVERFLOW

[ ] classification.predict([[32,1,1,0,np.log(1)]])

    1/1 [==============================] - 0s 204ms/step
    array([[0.09324051, 0.45112538, 0.21626192, 0.16876432, 0.07060798]],
          dtype=float32)
```

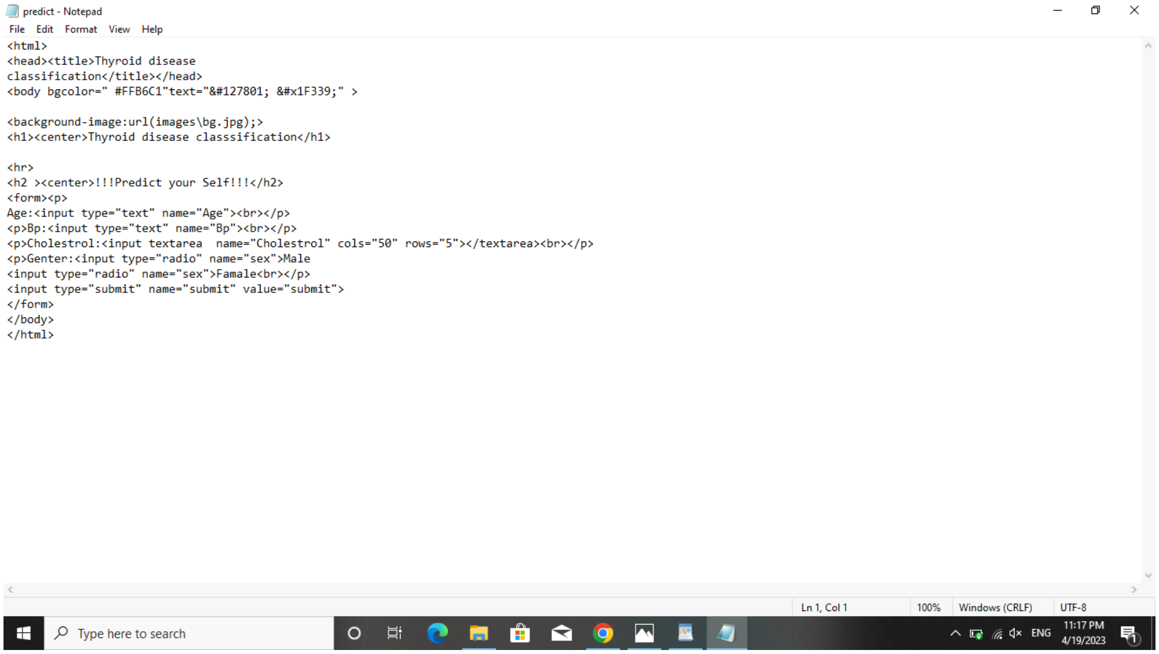

Thyroid classification.ipynb - Col
colab.research.google.com/drive/1dfkQdvqAAr9WicW7U3j5I_nelHWFLb3p#scrollTo=m5uIPQ_6FMC9
YouTube Maps Gmail News Translate
Thyroid classification.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:11 PM
Comment Share
+ Code + Text
Reconnect
Type here to search
ENG
10:28 PM
4/17/2023

```
    ----> 1 classification.fit(x_train,y_train,batch_size=2,epochs=30,validation_data=(x_test,y_test))
[ ]
    ─────────────────────── 1 frames ───────────────────────
    /usr/local/lib/python3.9/dist-packages/keras/engine/training.py in _assert_compile_was_called(self)
       3683         # (i.e. whether the model is built and its inputs/outputs are set).
       3684         if not self._is_compiled:
    -> 3685             raise RuntimeError(
       3686                 "You must compile your model before "
       3687                 "training/testing. "

    RuntimeError: You must compile your model before training/testing. Use `model.compile(optimizer, loss)`.

    SEARCH STACK OVERFLOW

[ ] classification.predict([[32,1,1,0,np.log(1)]])

    1/1 [==============================] - 0s 204ms/step
    array([[0.09324051, 0.45112538, 0.21626192, 0.16876432, 0.07060798]],
          dtype=float32)

[ ]

[ ] import pickle
    pickle.dump(svc1,open('thyroid_1_model.pkl','wb'))
```

Html page:

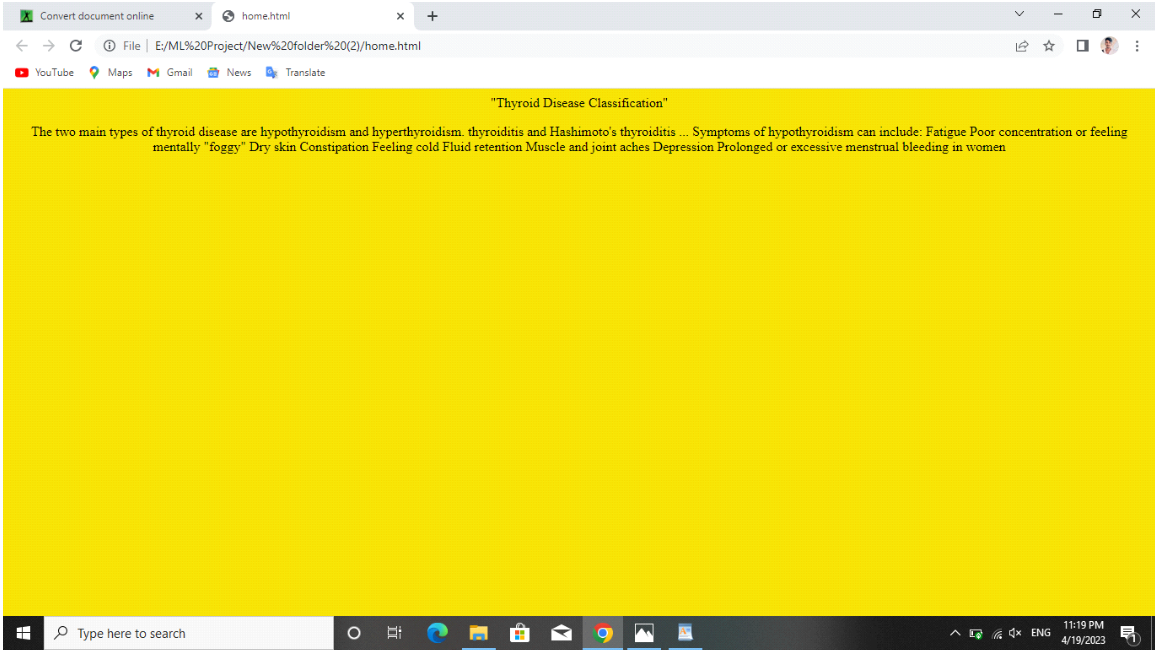


```
<html>
<head><center>"Thyroid Disease Classification"
</head>
<body bgcolor="  #F1DE06 " text=" " >
<p>The two main types of thyroid disease are hypothyroidism and hyperthyroidism.
thyroiditis and Hashimoto's thyroiditis  ...
  Symptoms of hypothyroidism can include:

Fatigue
Poor concentration or feeling mentally "foggy"
Dry skin
Constipation
Feeling cold
Fluid retention
Muscle and joint aches
Depression
Prolonged or excessive menstrual bleeding in women
</p>
</html>
```



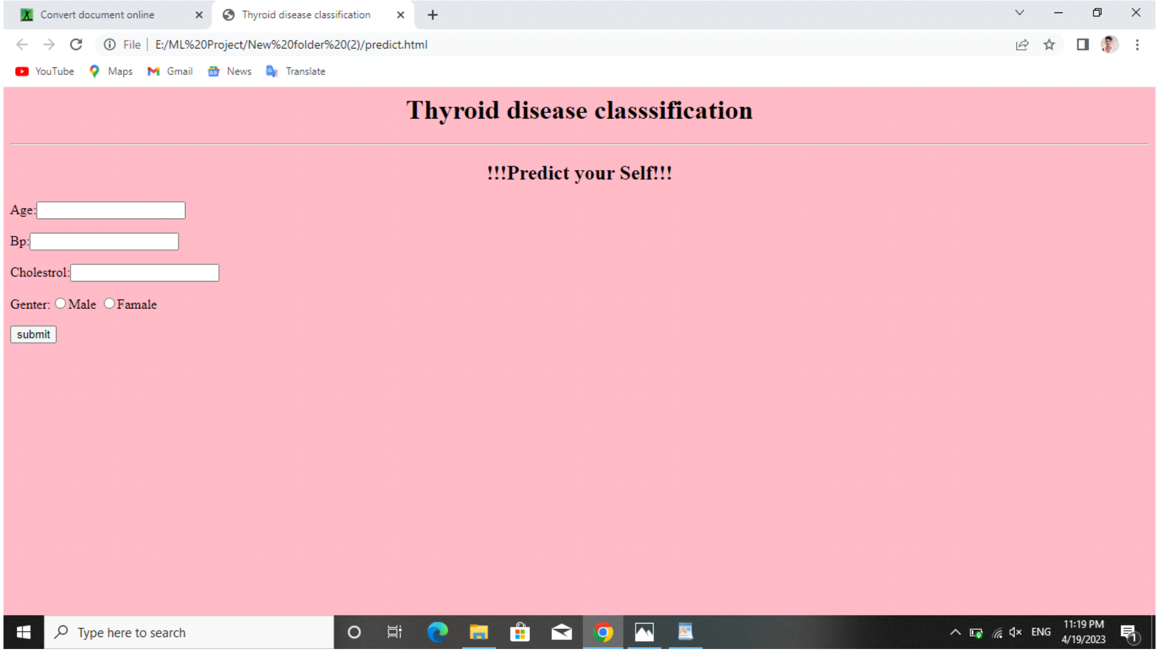
```
<html>
<head><title>Thyroid disease
classification</title></head>
<body bgcolor=" #FFB6C1"text="&#127801; &#x1F339;" >

<background-image:url(images\bg.jpg);>
<h1><center>Thyroid disease classsification</h1>

<hr>
<h2 ><center>!!!Predict your Self!!!</h2>
<form><p>
Age:<input type="text" name="Age"><br></p>
<p>Bp:<input type="text" name="Bp"><br></p>
<p>Cholestrol:<input textarea  name="Cholestrol" cols="50" rows="5"></textarea><br></p>
<p>Genter:<input type="radio" name="sex">Male
<input type="radio" name="sex">Famale<br></p>
<input type="submit" name="submit" value="submit">
</form>
</body>
</html>
```

Output:



"Thyroid Disease Classification"

The two main types of thyroid disease are hypothyroidism and hyperthyroidism. thyroiditis and Hashimoto's thyroiditis ... Symptoms of hypothyroidism can include: Fatigue Poor concentration or feeling mentally "foggy" Dry skin Constipation Feeling cold Fluid retention Muscle and joint aches Depression Prolonged or excessive menstrual bleeding in women



Thyroid disease classsification

!!!Predict your Self!!!

Age:

Bp:

Cholestrol:

Genter: ○ Male ○ Famale

submit

THANK YOU !!!