THYROID DISEASE CLASSIFICATION USING MACHINE LEARNING ALGORITHMS

DONE BY

K.AGALYA(20326ER039)

A.AMEERA BANU(20326ER040)

P.ANUPRIYA(20326ER041)

A.ARUL AKILA(20326ER042)

CONTENT

- INTRODUCTION
- PROBLEM DEFINITION & DESIGN THINKING
- RESULT
- ADVANTAGES
- APPLICATIONS
- CONCLUSION
- FUTURE SCOPE

ABSTRACT

- **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 body's metabolism.**
- **Machine Learning 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.**

INTRODUCTION

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 body's metabolism.
- 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.

Purpose

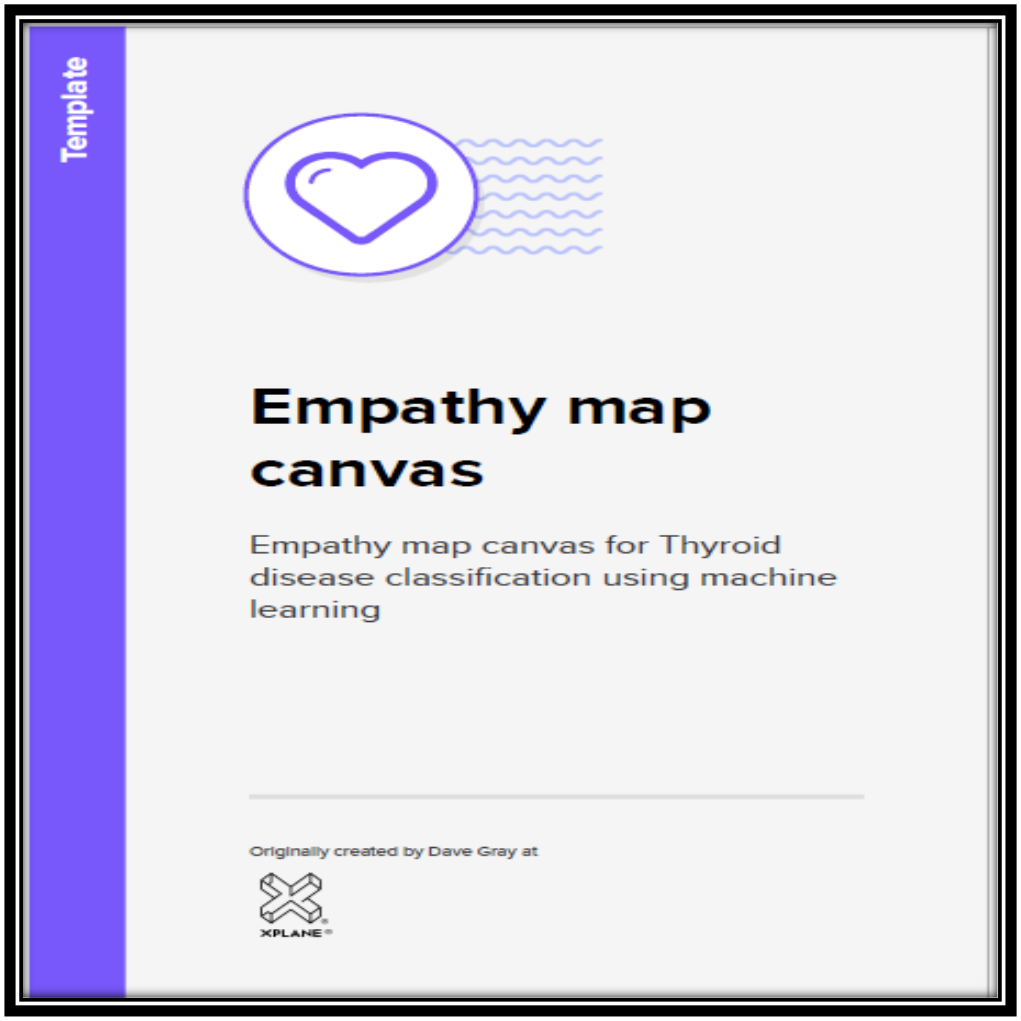
- Machine learning algorithms can help improve the accuracy and efficiency of thyroid disease diagnosis by analyzing large amounts of data from diverse sources, such as medical records, laboratory tests, and imaging studies.
- These algorithms can learn from the data and develop models that can accurately classify different types of thyroid diseases based on their features and characteristics.
- The use of machine learning algorithms for thyroid disease classification can also help reduce the time and cost involved in diagnosis, leading to faster and more efficient patient care.

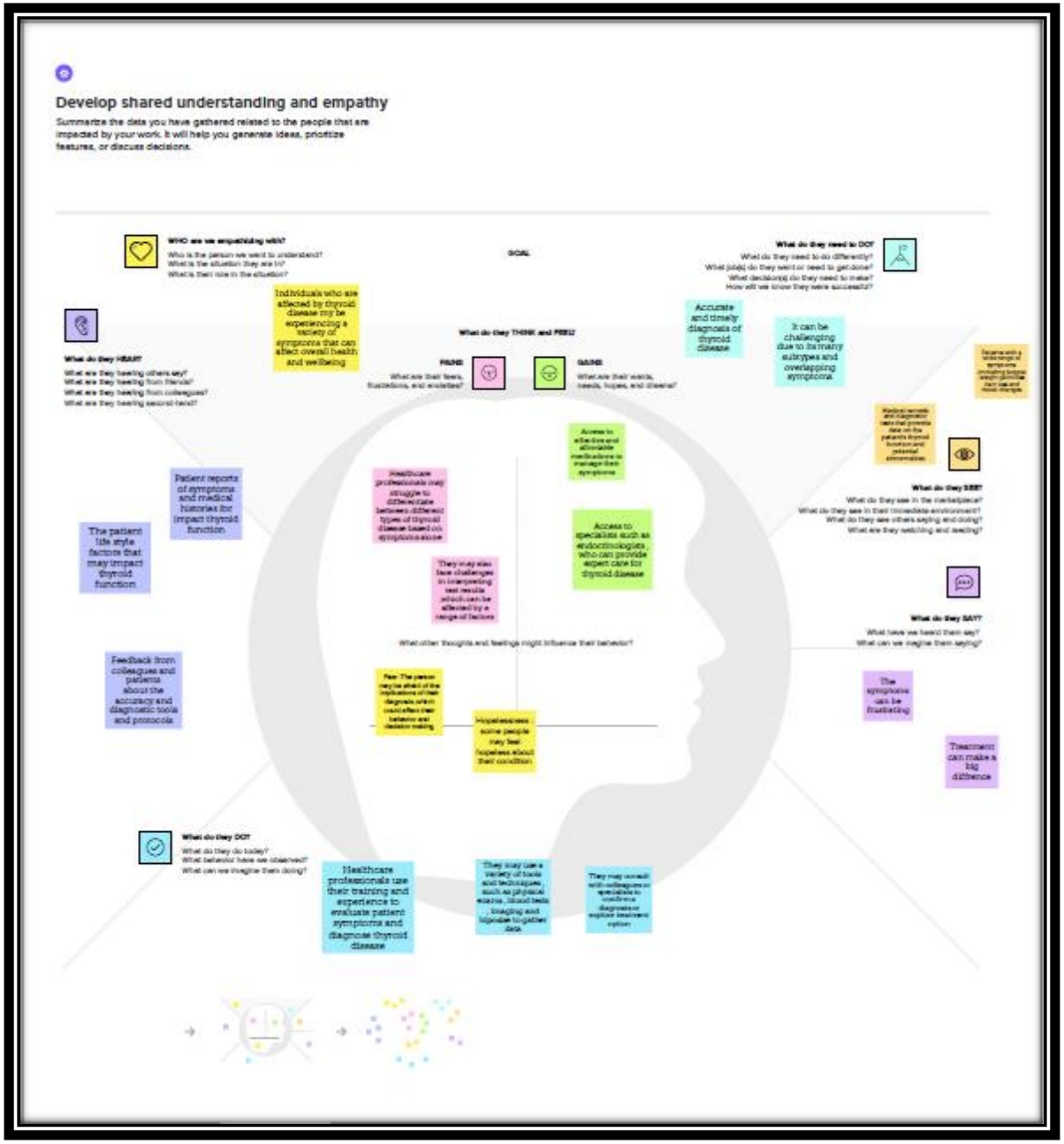
PROBLEM DEFINITION AND DESIGN THINKING

Empathy map

The empathy map is a tool used in design thinking and customer experience design to better understand the needs and experiences of customers or users. It is a simple framework that helps teams develop empathy for their users by visualizing their thoughts, feelings, and behaviors.

Empathy Map


Template
Empathy map canvas
Empathy map canvas for Thyroid disease classification using machine learning
Originally created by Dave Gray at
XPLANE

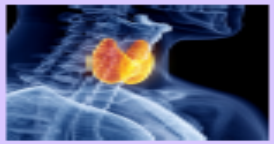

Develop shared understanding and empathy
Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.
WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
GOAL
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
Individuals who are affected by thyroid disease may be experiencing a variety of symptoms that can affect overall health and wellbeing
Accurate and timely diagnosis of thyroid disease
It can be challenging due to its many subtypes and overlapping symptoms
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes, and dreams?
Access to effective and affordable medications to manage their symptoms
Patient reports of symptoms and medical histories for impact thyroid function
Healthcare professionals may struggle to differentiate between different types of thyroid disease based on symptoms alone
Access to specialists such as endocrinologists, who can provide expert care for thyroid disease
The patient life style factors that may impact thyroid function
They may also face challenges in interpreting test results which can be affected by a range of factors
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What other thoughts and feelings might influence their behavior?
What do they SAY?
What have we heard them say?
What can we imagine them saying?
Feedback from colleagues and patients about the accuracy and diagnostic tools and protocols
Fear: The person may be afraid of the implications of their diagnosis which could affect their behavior and decision making
Hopelessness: some people may feel hopeless about their condition
The symptoms can be frustrating
Treatment can make a big difference
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
Healthcare professionals use their training and experience to evaluate patient symptoms and diagnose thyroid disease
They may use a variety of tools and techniques, such as physical exams, blood tests, imaging and biopsies to gather data
They may consult with colleagues or specialists to confirm a diagnosis or explore treatment options

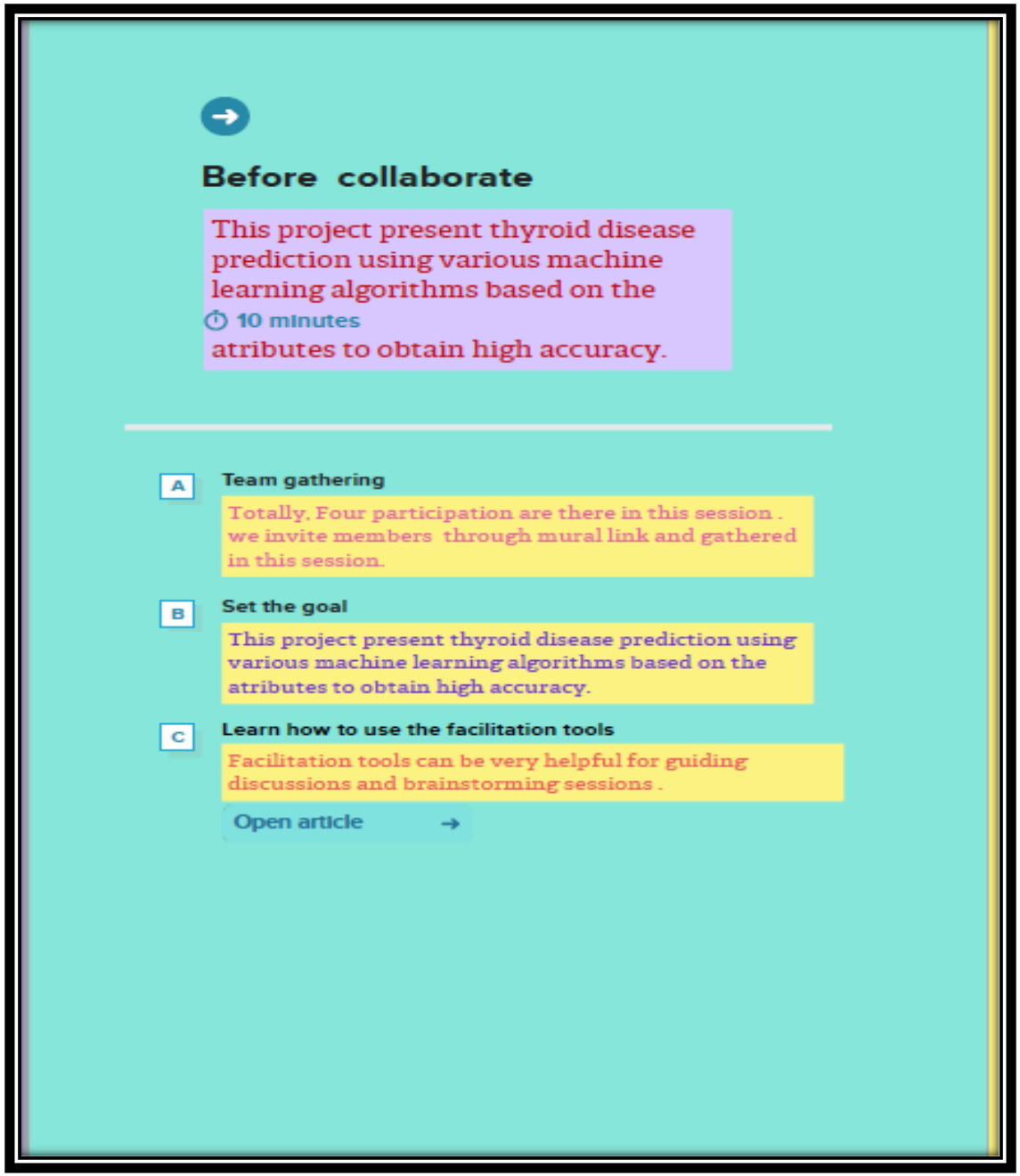
Ideation & Brainstorming Map

- Brainstorming is a process of generating creative and innovative ideas through group discussion and collaboration. It is typically used in the early stages of a project or problem-solving process to generate a wide range of ideas without judgment or criticism.
- Set clear goals and objectives: Before starting the brainstorming session, it's essential to define the problem or challenge you want to address and set clear goals and objectives for the session.
- Encourage participation: Encourage all participants to contribute their ideas and opinions, no matter how unconventional or outlandish they may seem. The goal is to generate as many ideas as possible.
- Avoid criticism and judgment: Criticism and judgment can stifle creativity and prevent participants from sharing their ideas. Encourage participants to build on each other's ideas and avoid negative feedback.

Brainstorm & Idea prioritization

Thyroid Disease Classification Using Machine Learning




Before collaborate
This project present thyroid disease prediction using various machine learning algorithms based on the
10 minutes
atributes to obtain high accuracy.
A
Team gathering
Totally, Four participation are there in this session . we invite members through mural link and gathered in this session.
B
Set the goal
This project present thyroid disease prediction using various machine learning algorithms based on the atributes to obtain high accuracy.
C
Learn how to use the facilitation tools
Facilitation tools can be very helpful for guiding discussions and brainstorming sessions .
Open article

1

Problem statement

1.The Thyroid Disease can be easily identify based on symptoms in the patient's history.

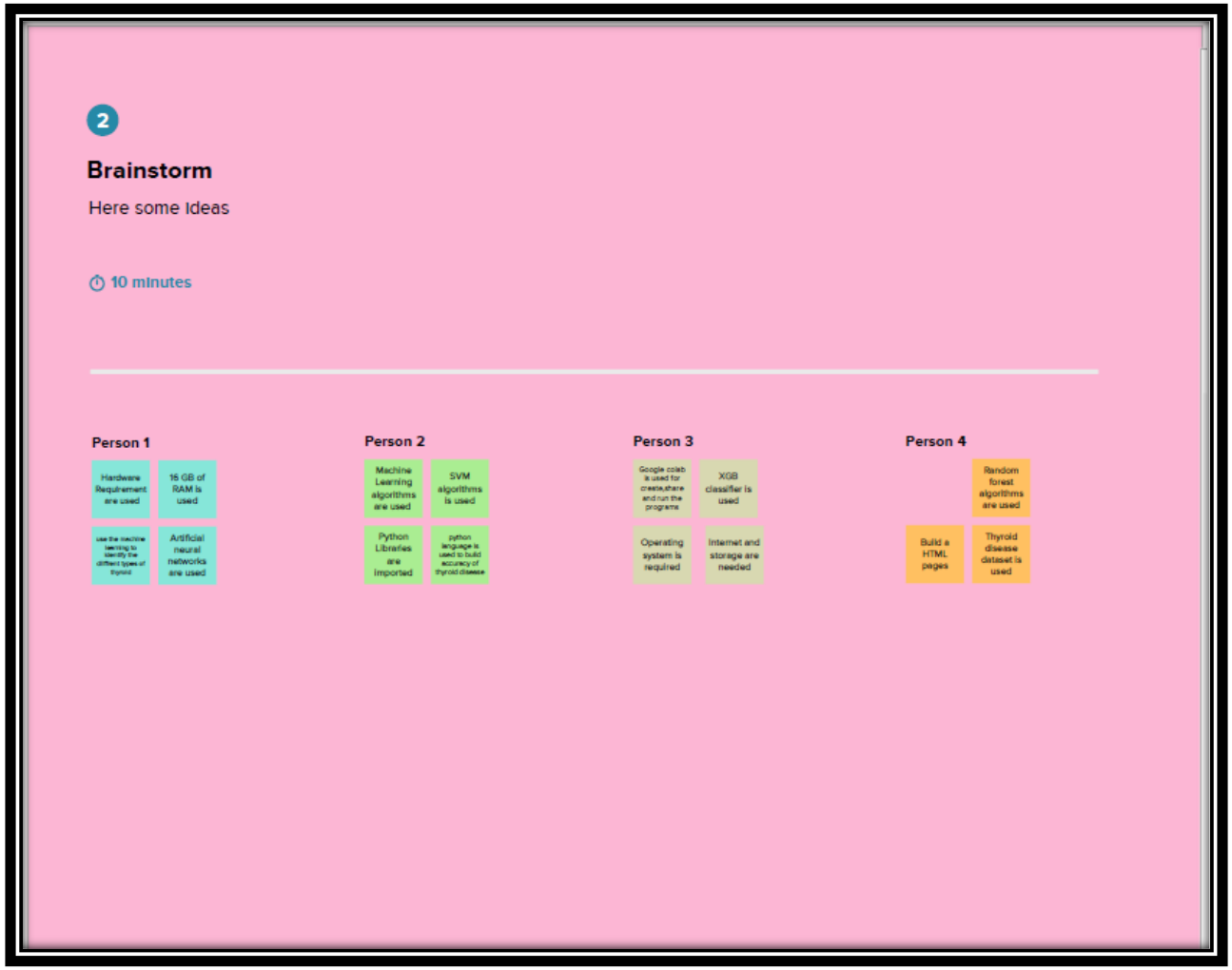
5 minutes

2.Thyroid disease is a medical condition that affects the function of the thyroid gland

3.This project present thyroid disease predction using various machine learning algorithms based on the atributes to obtain high accuracy

4.That dataset create for this project is apply from the kaggle thyroid disease dataset.

5.The important of feature show be estimated to select the option number of features thyroid disease classification.

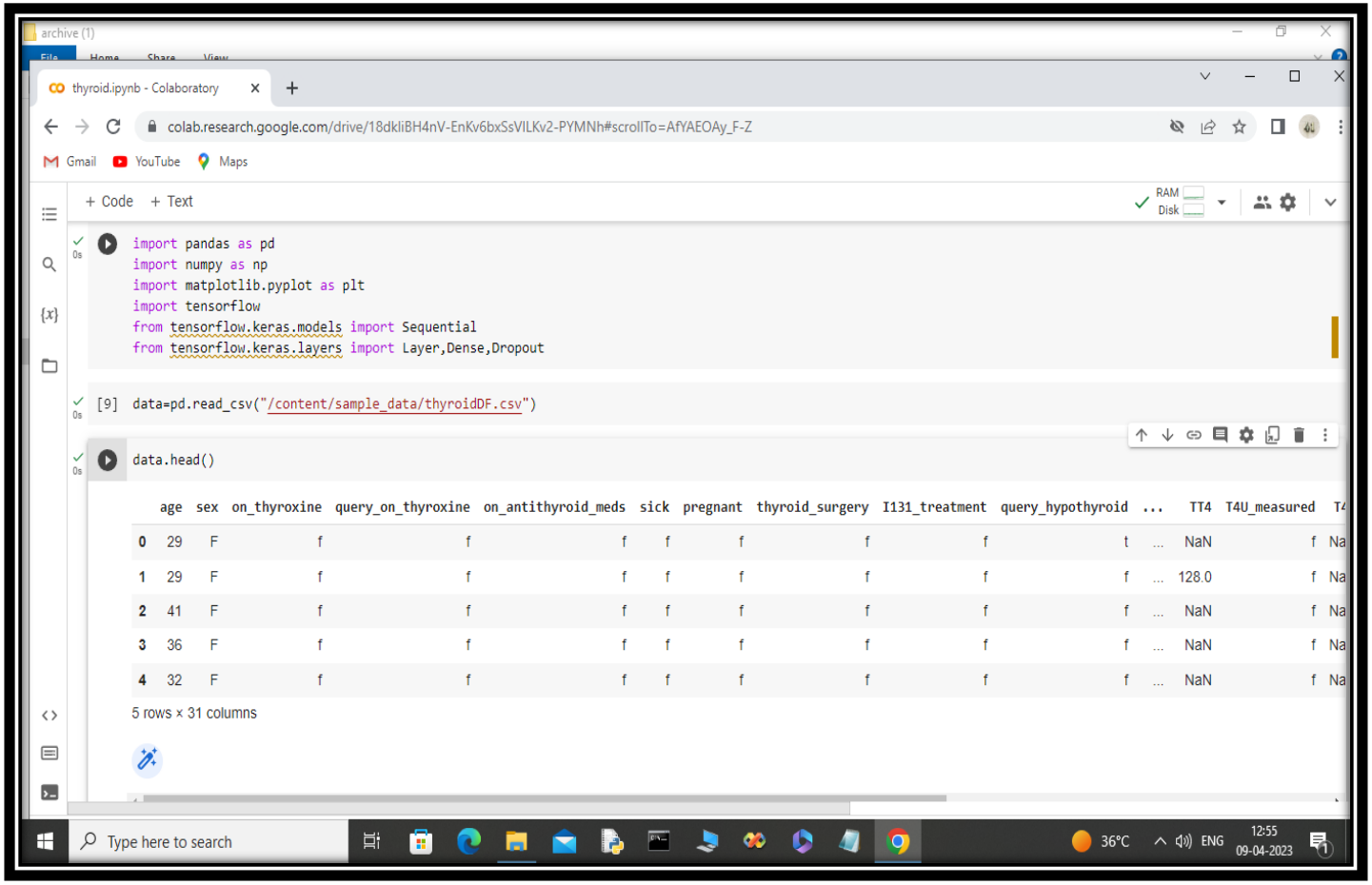

2
Brainstorm
Here some Ideas
10 minutes
Person 1
Hardware Requirement are used
16 GB of RAM is used
use the machine learning to identify the different types of thyroid
Artificial neural networks are used
Person 2
Machine Learning algorithms are used
SVM algorithms is used
Python Libraries are Imported
python language is used to build accuracy of thyroid disease
Person 3
Google colab is used for create,share and run the programs
XGB classifier is used
Operating system is required
Internet and storage are needed
Person 4
Random forest algorithms are used
Build a HTML pages
Thyroid disease dataset is used

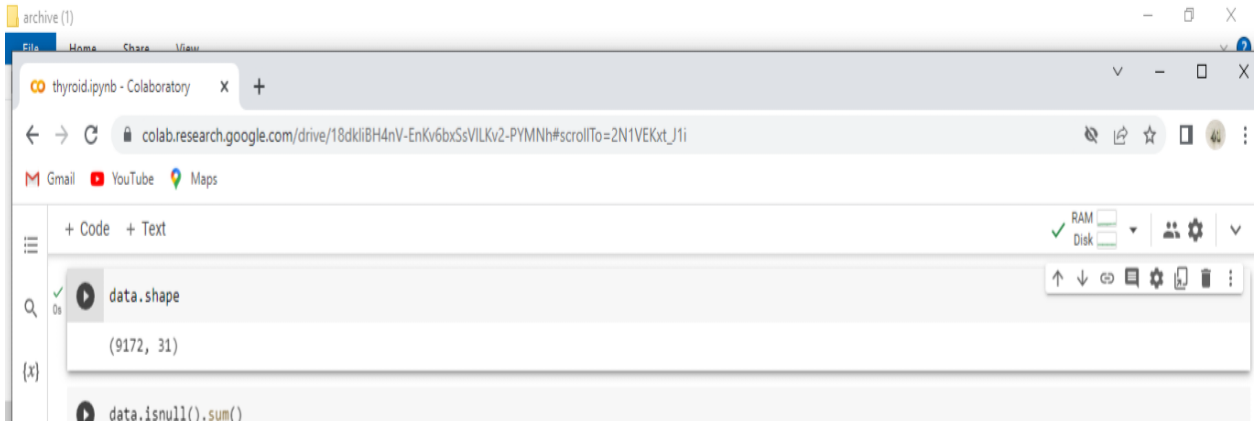

Prioritize
Prioritize the ideas
20 minutes
Feasibility

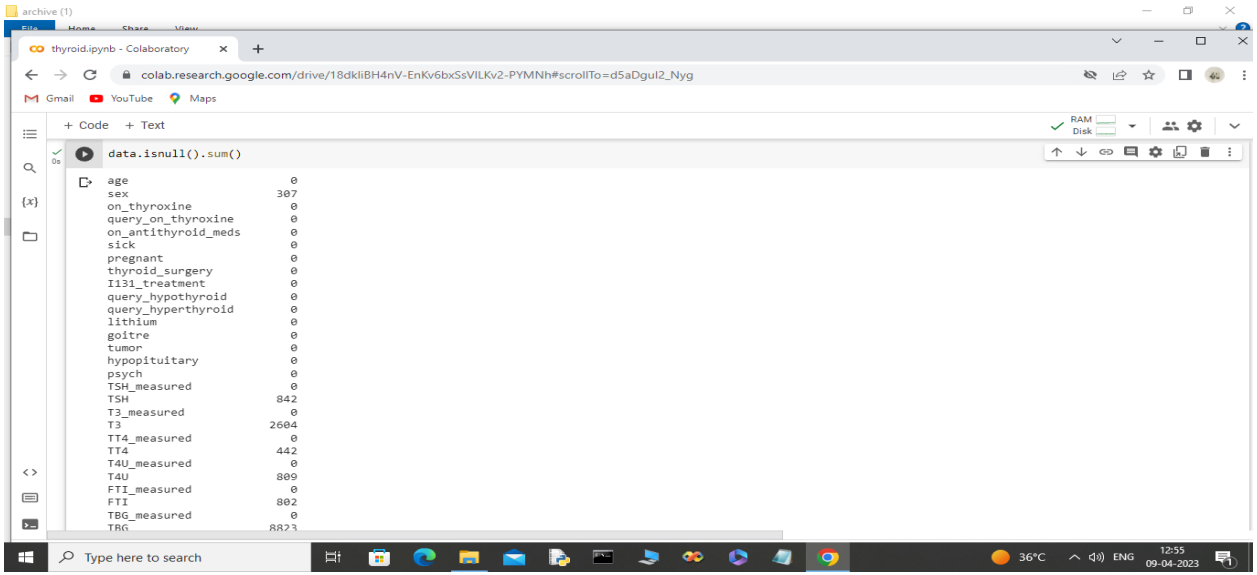
After collaborate

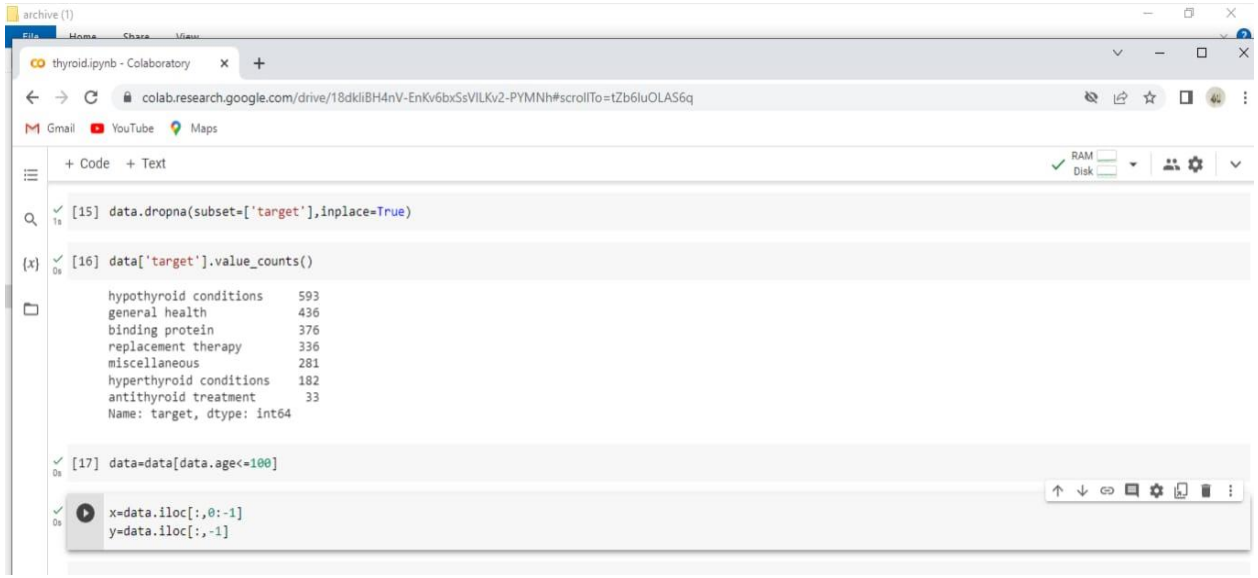
we can export the mural as pdf to share . It is helpful to getting information.

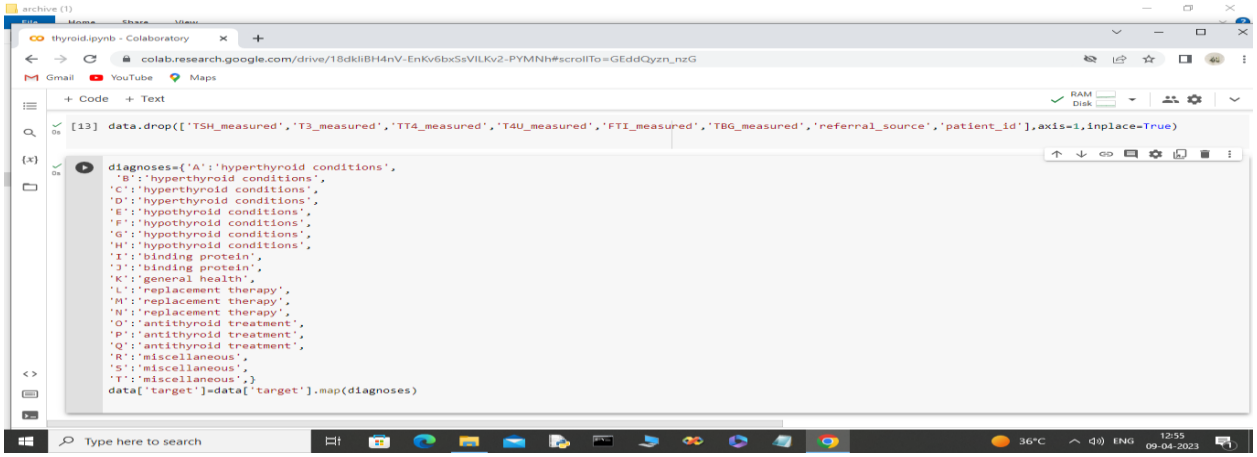
Importing the libraries and data preprocessing



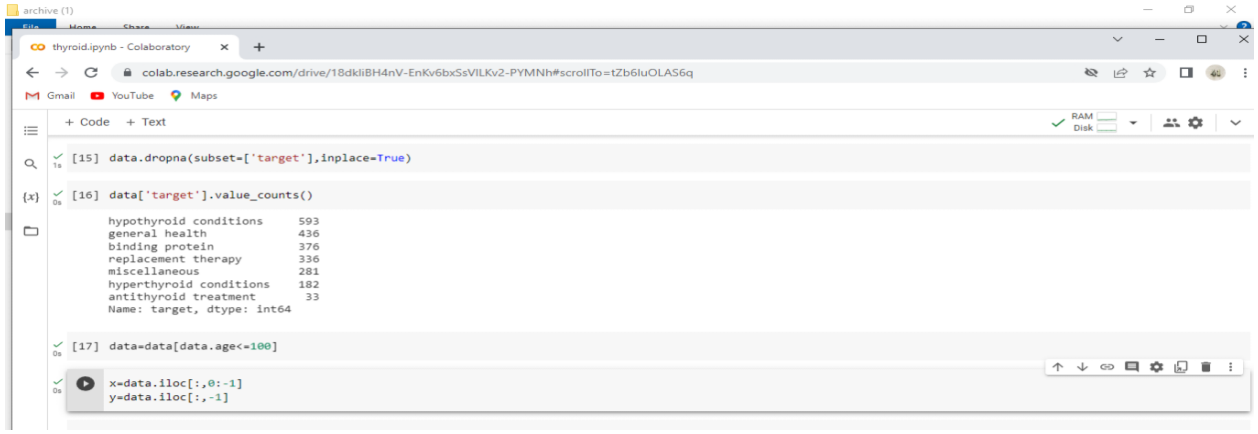

archive (1)
thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=2N1VEKxt_J1i
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
data.shape
(9172, 31)
data.isnull().sum()

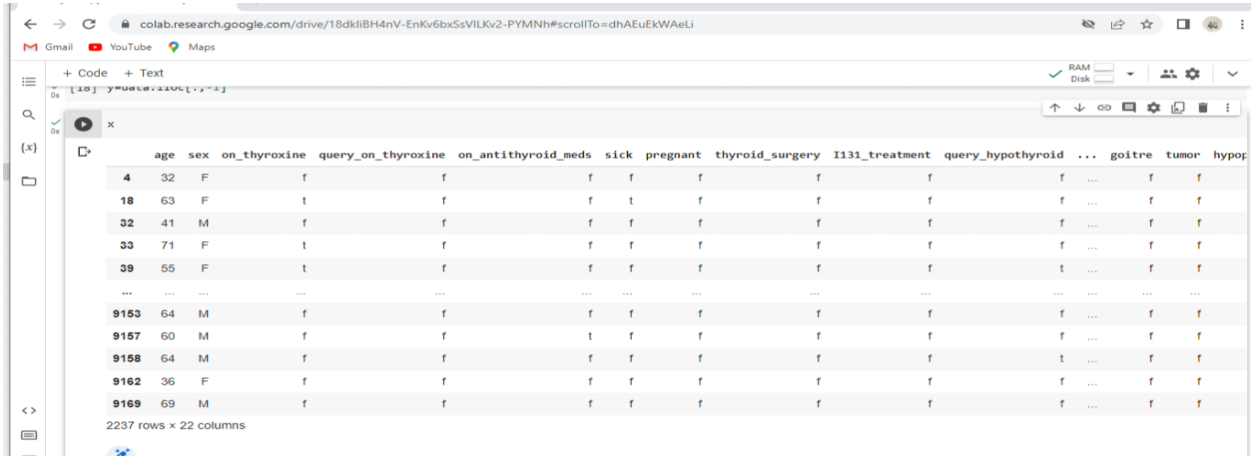

archive (1)
thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=d5aDgul2_Nyg
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
data.isnull().sum()
age 0
sex 307
on_thyroxine 0
query_on_thyroxine 0
on_antithyroid_meds 0
sick 0
pregnant 0
thyroid_surgery 0
I131_treatment 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
Type here to search
36°C
ENG
12:55
09-04-2023


archive (1)
thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=tZb6luOLAS6q
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[15] data.dropna(subset=['target'],inplace=True)
[16] 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
[17] data=data[data.age<=100]
x=data.iloc[:,0:-1]
y=data.iloc[:,-1]

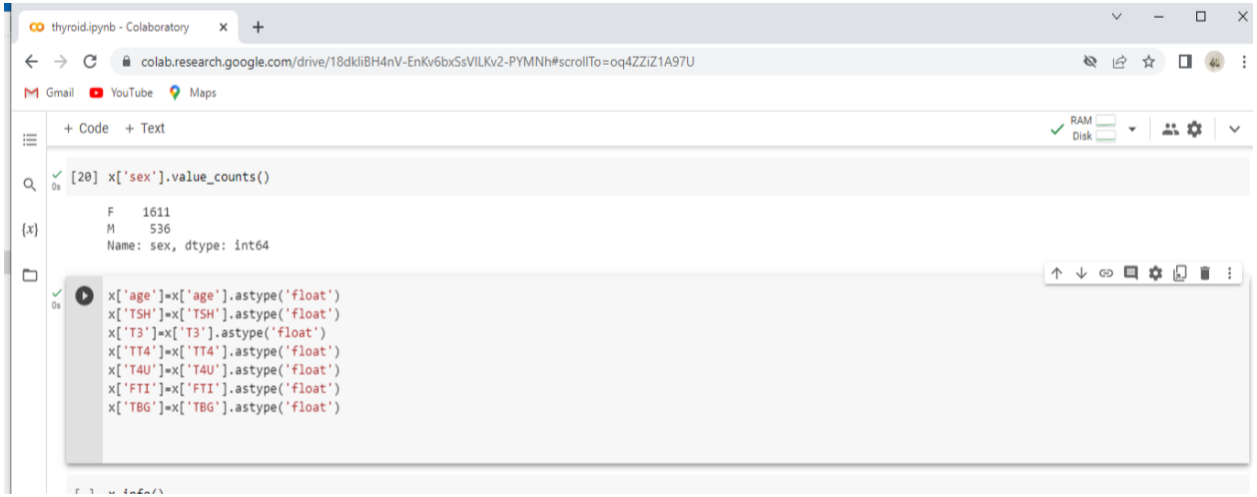

archive (1)
thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=GEddQyzn_nzG
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[13] data.drop(['TSH_measured','T3_measured','TT4_measured','T4U_measured','FTI_measured','TBG_measured','referral_source','patient_id'],axis=1,inplace=True)
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)
Type here to search
36°C
ENG
12:55
09-04-2023

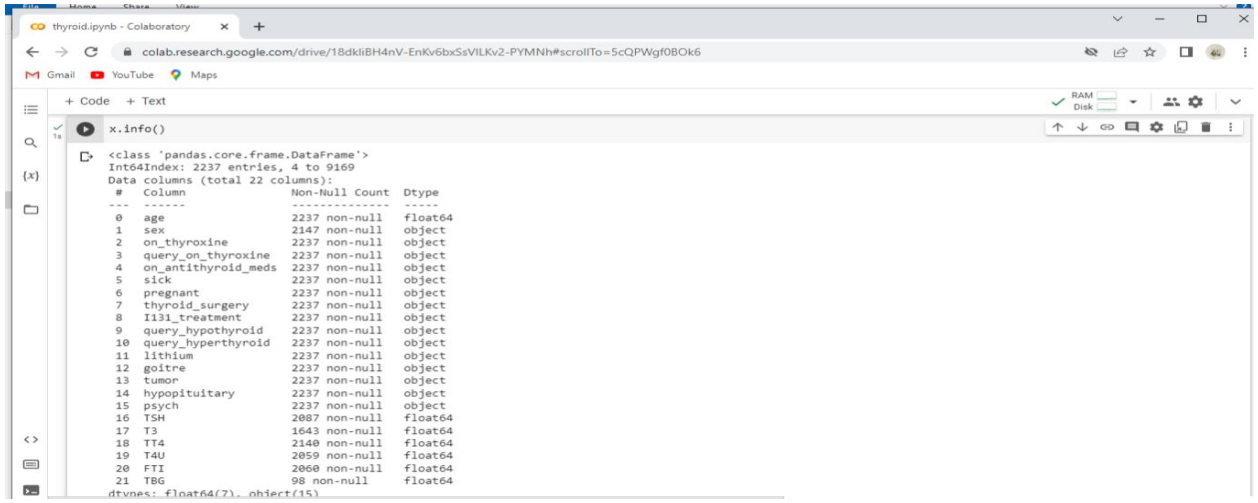
Splitting the data x and y



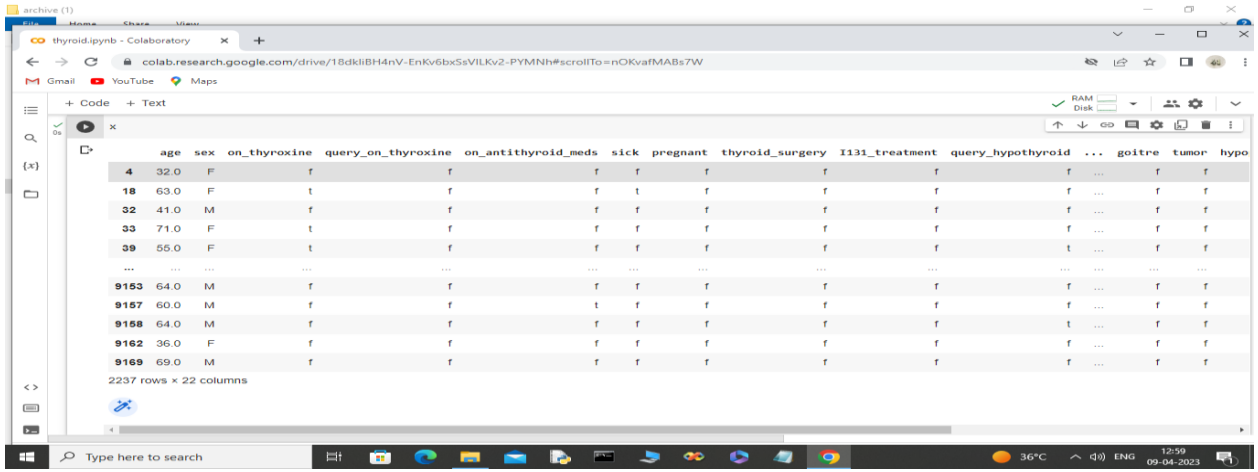


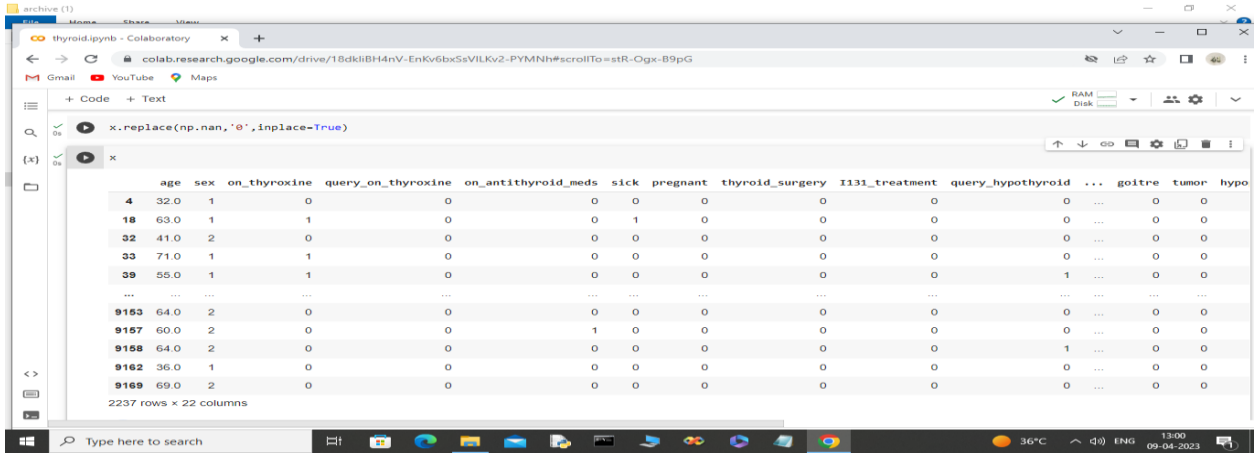
Converting the data types

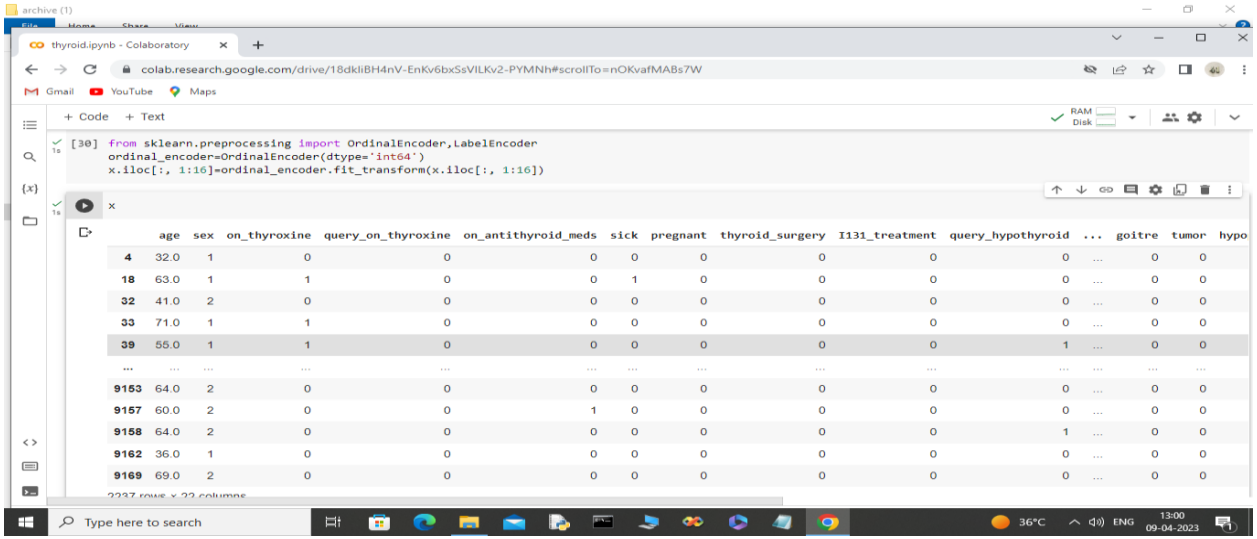


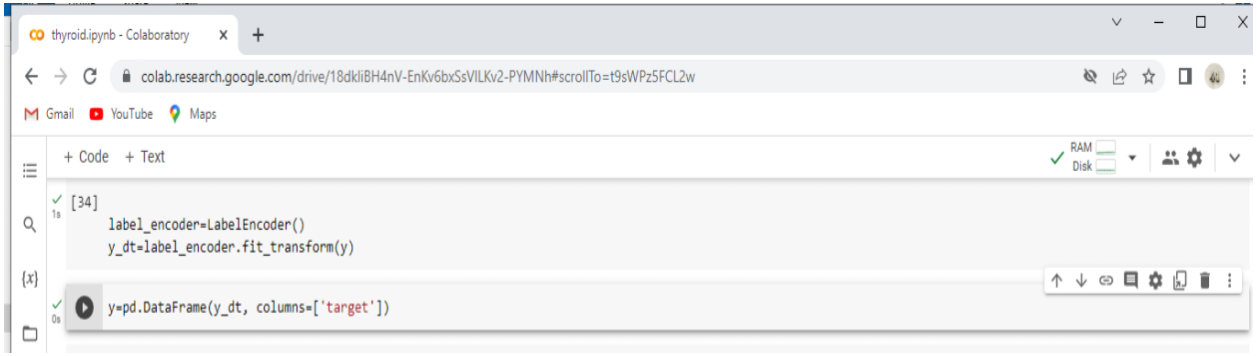


Handling categorical values

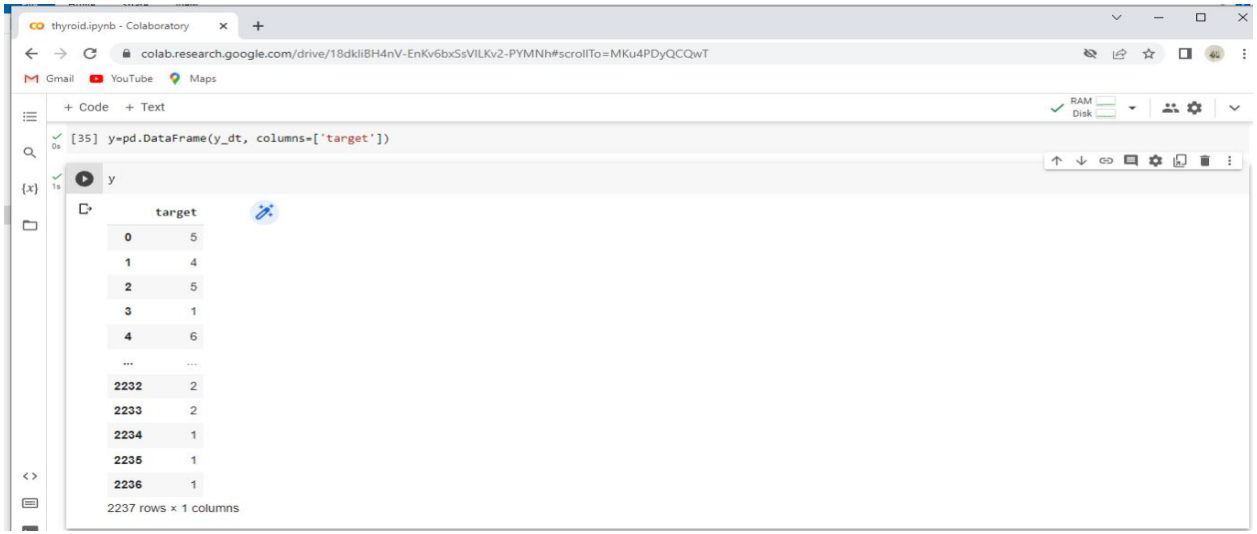


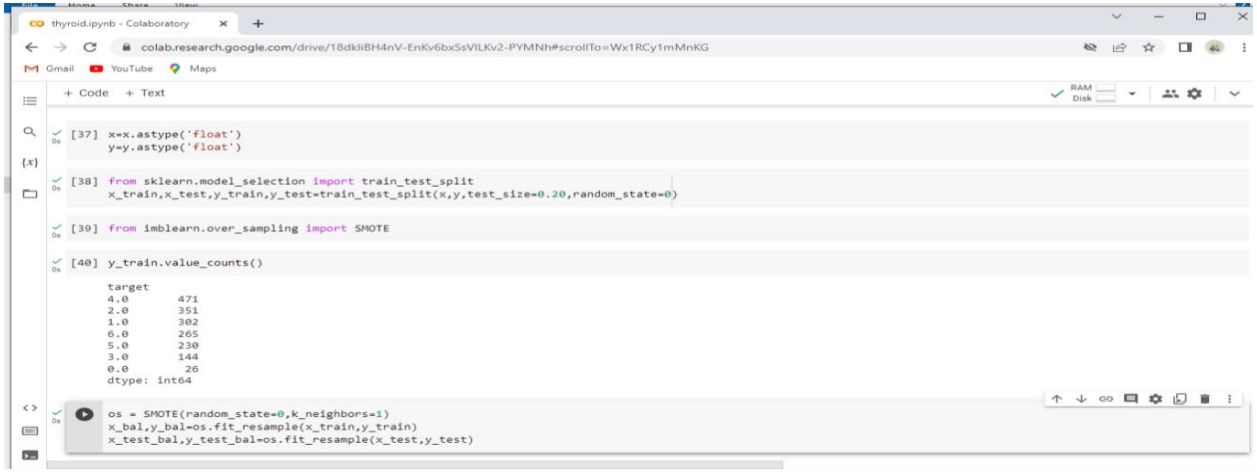



thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=nOKvafMABs7W
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[30] from sklearn.preprocessing import OrdinalEncoder,LabelEncoder
ordinal_encoder=OrdinalEncoder(dtype='int64')
x.iloc[:, 1:16]=ordinal_encoder.fit_transform(x.iloc[:, 1:16])
age sex on_thyroxine query_on_thyroxine on_antithyroid_meds sick pregnant thyroid_surgery I131_treatment query_hypothyroid ... goitre tumor hypo
4 32.0 1 0 0 0 0 0 0 0 0 ... 0 0
18 63.0 1 1 0 0 1 0 0 0 0 ... 0 0
32 41.0 2 0 0 0 0 0 0 0 0 ... 0 0
33 71.0 1 1 0 0 0 0 0 0 0 ... 0 0
39 55.0 1 1 0 0 0 0 0 0 1 ... 0 0
9153 64.0 2 0 0 0 0 0 0 0 0 ... 0 0
9157 60.0 2 0 0 1 0 0 0 0 0 ... 0 0
9158 64.0 2 0 0 0 0 0 0 0 1 ... 0 0
9162 36.0 1 0 0 0 0 0 0 0 0 ... 0 0
9169 69.0 2 0 0 0 0 0 0 0 0 ... 0 0
2237 rows x 22 columns
Type here to search
36°C
ENG
13:00
09-04-2023

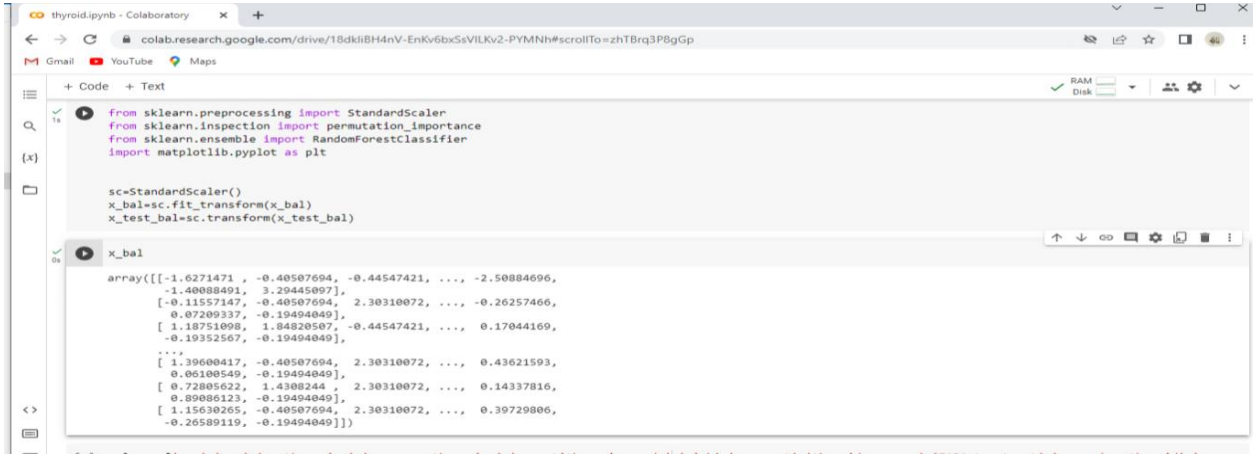

thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=t9sWPz5FCL2w
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[34] label_encoder=LabelEncoder()
y_dt=label_encoder.fit_transform(y)
y=pd.DataFrame(y_dt, columns=['target'])

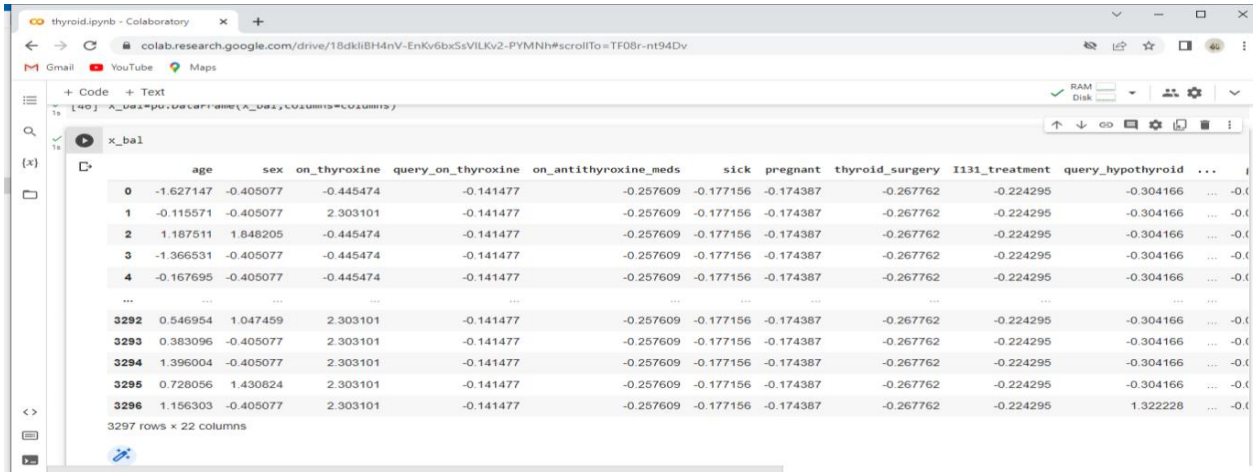
Splitting data into train and test





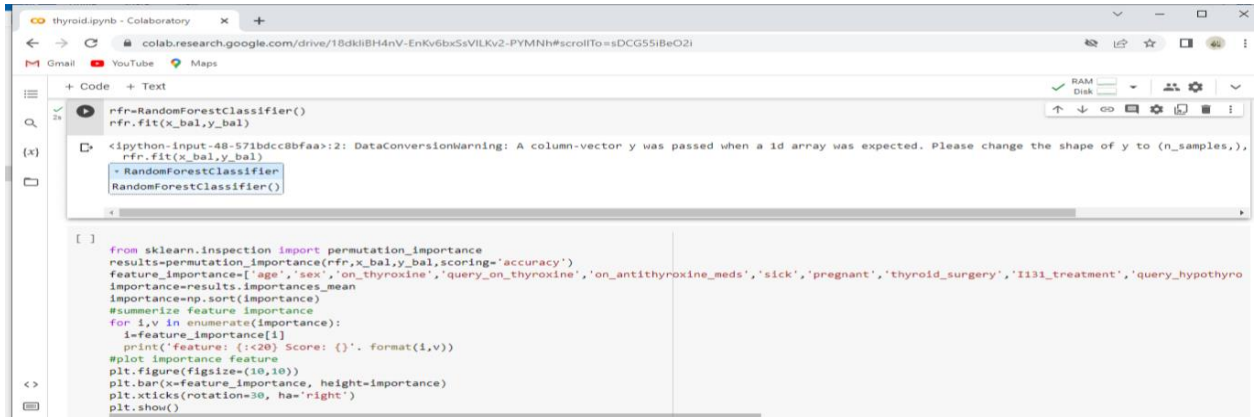
Handling imbalanced data

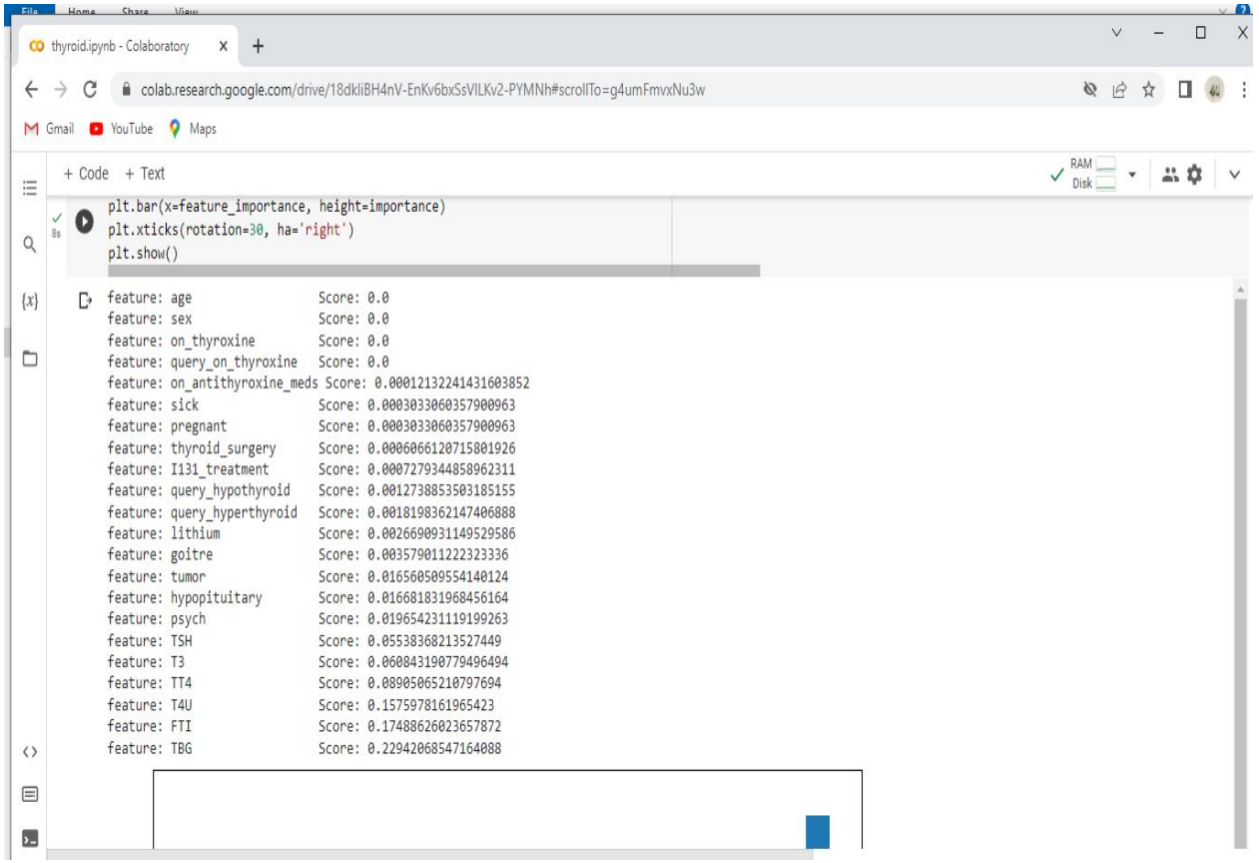


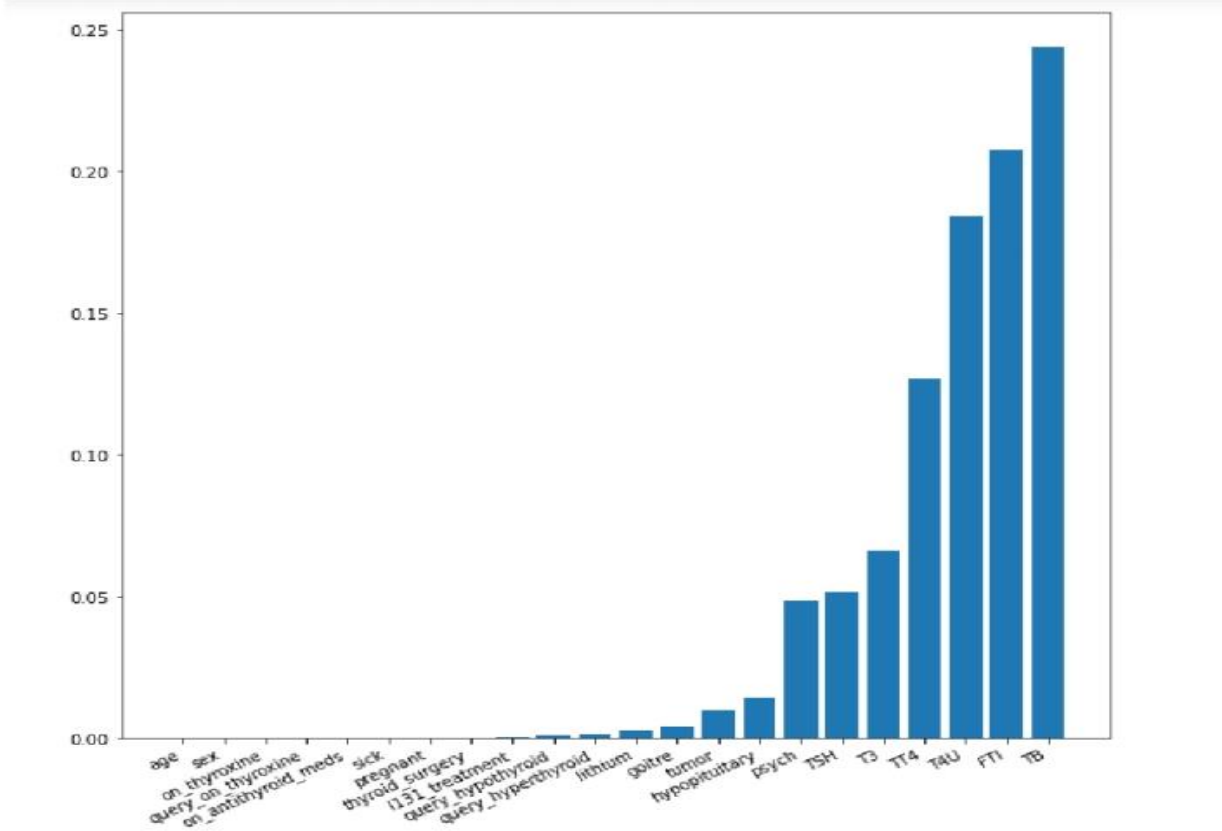


	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroxine_meds	sick	pregnant	thyroid_surgery	I131_treatment	query_hypothyroid	...	
0	-1.627147	-0.405077	-0.445474	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
1	-0.115571	-0.405077	2.303101	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
2	1.187511	1.848205	-0.445474	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
3	-1.366531	-0.405077	-0.445474	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
4	-0.167695	-0.405077	-0.445474	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
...	...	...	...	...	...	...	...	...	...	...	...	
3292	0.546954	1.047459	2.303101	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
3293	0.383096	-0.405077	2.303101	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
3294	1.396004	-0.405077	2.303101	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
3295	0.728056	1.430824	2.303101	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	-0.304166	...	-0.0
3296	1.156303	-0.405077	2.303101	-0.141477	-0.257609	-0.177156	-0.174387	-0.267762	-0.224295	1.322228	...	-0.0

3297 rows × 22 columns




thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=g4umFmvxNu3w
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
plt.bar(x=feature_importance, height=importance)
plt.xticks(rotation=30, ha='right')
plt.show()
feature: age Score: 0.0
feature: sex Score: 0.0
feature: on_thyroxine Score: 0.0
feature: query_on_thyroxine Score: 0.0
feature: on_antithyroxine_meds Score: 0.0001213224143160385 2
feature: sick Score: 0.0003033060357900963
feature: pregnant Score: 0.0003033060357900963
feature: thyroid_surgery Score: 0.0006066120715801926
feature: I131_treatment Score: 0.0007279344858962311
feature: query_hypothyroid Score: 0.0012738853503185155
feature: query_hyperthyroid Score: 0.0018198362147406888
feature: lithium Score: 0.0026690931149529586
feature: goitre Score: 0.003579011222323336
feature: tumor Score: 0.016560509554140124
feature: hypopituitary Score: 0.016681831968456164
feature: psych Score: 0.019654231119199263
feature: TSH Score: 0.05538368213527449
feature: T3 Score: 0.060843190779496494
feature: TT4 Score: 0.08905065210797694
feature: T4U Score: 0.1575978161965423
feature: FTI Score: 0.17488626023657872
feature: TBG Score: 0.22942068547164088

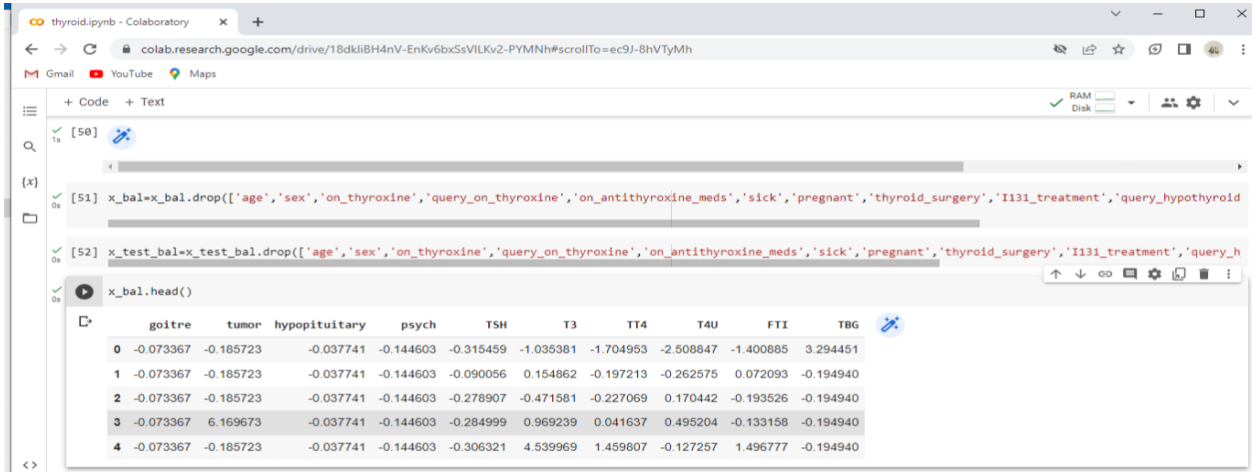

0.25
0.20
0.15
0.10
0.05
0.00
age
sex
on_thyroxine
query_on_thyroxine
on_antithyroid_meds
sick
pregnant
thyroid_surgery
i131_treatment
query_hypothyroid
query_hyperthyroid
lithium
goitre
tumor
hypopituitary
psych
TSH
T3
TT4
T4U
FTI
TB

Selecting Output Columns

```
x.head()
```

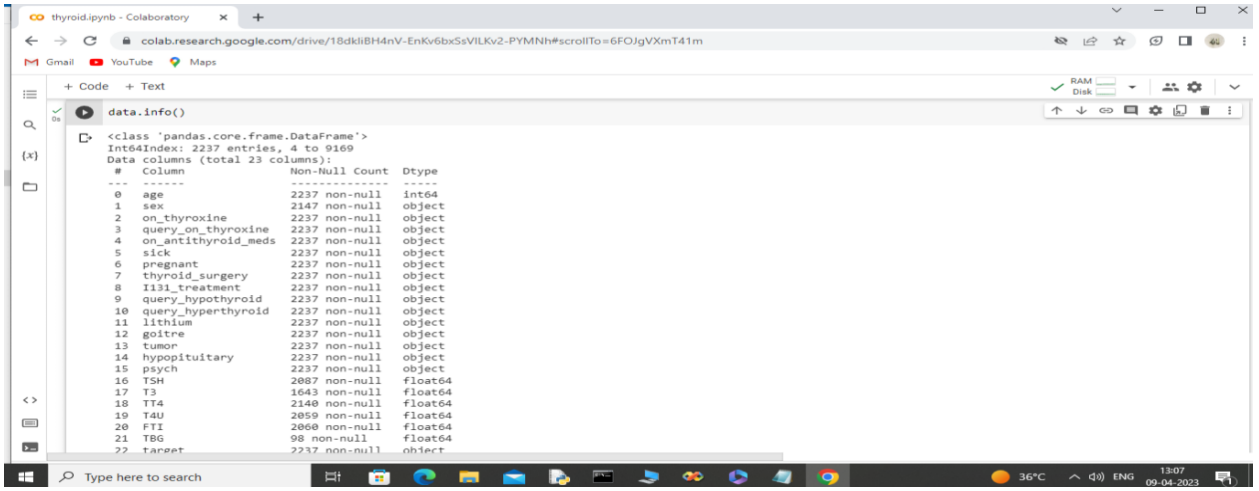
	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	query_hypothyroid	...	goitre	tumor	hypopi
4	32.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
18	63.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
32	41.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
33	71.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
39	55.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	

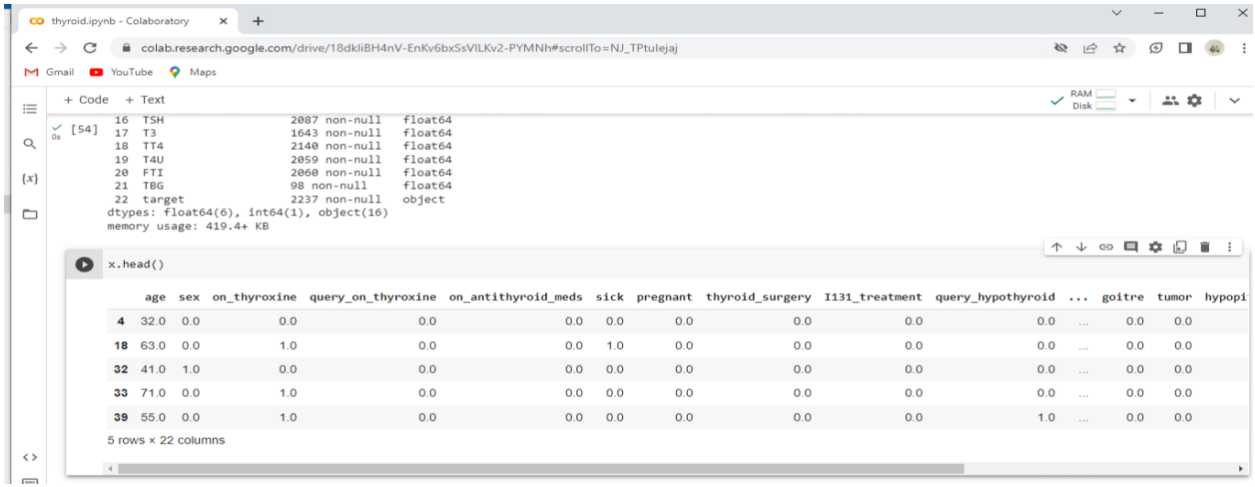
5 rows x 22 columns



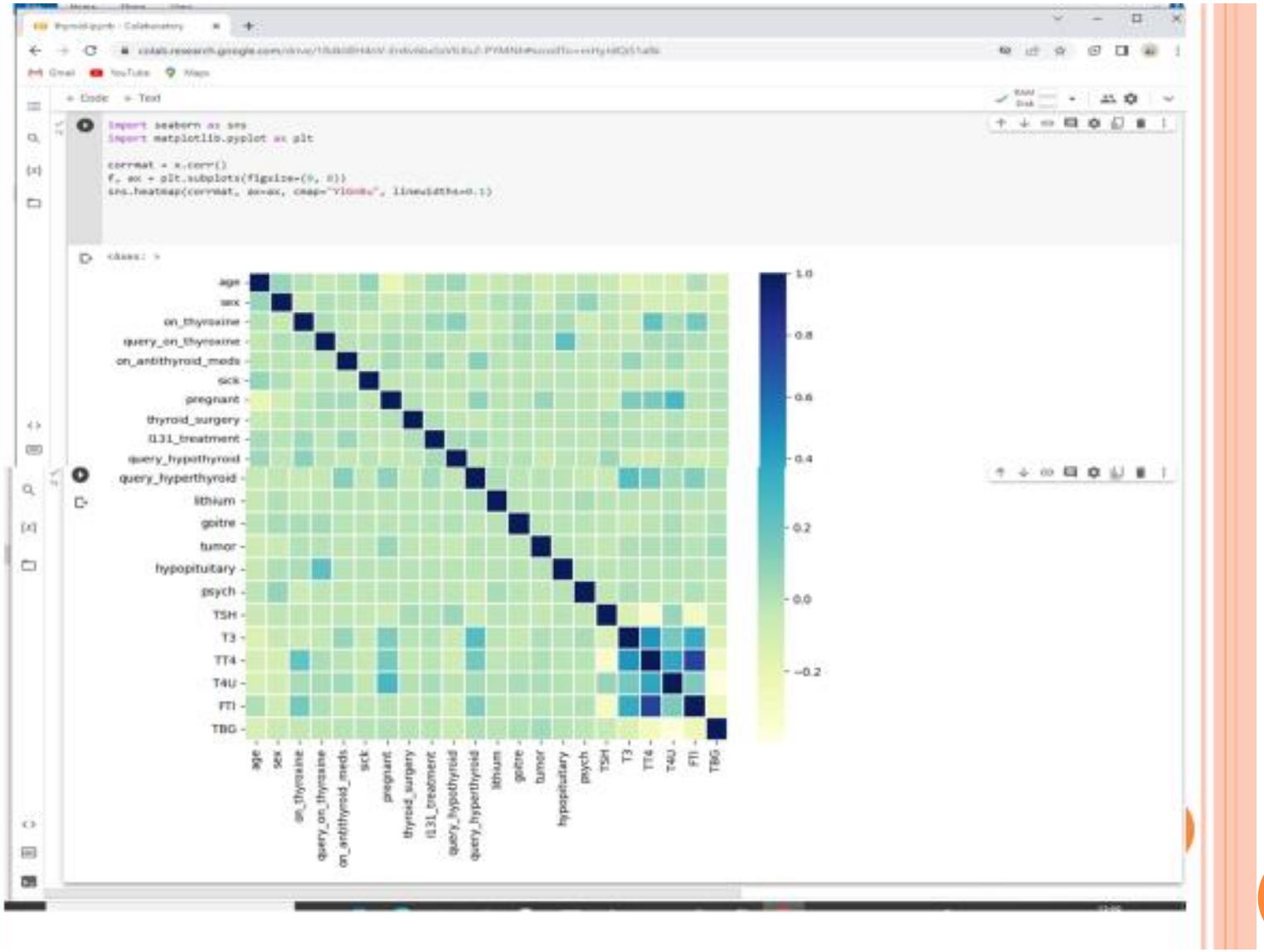
	goitre	tumor	hypopituitary	psych	TSH	T3	TT4	T4U	FTI	TBG
0	-0.073367	-0.185723	-0.037741	-0.144603	-0.315459	-1.035381	-1.704953	-2.508847	-1.400885	3.294451
1	-0.073367	-0.185723	-0.037741	-0.144603	-0.090056	0.154862	-0.197213	-0.262575	0.072093	-0.194940
2	-0.073367	-0.185723	-0.037741	-0.144603	-0.278907	-0.471581	-0.227069	0.170442	-0.193526	-0.194940
3	-0.073367	6.169673	-0.037741	-0.144603	-0.284999	0.969239	0.041637	0.495204	-0.133158	-0.194940
4	-0.073367	-0.185723	-0.037741	-0.144603	-0.306321	4.539969	1.459807	-0.127257	1.496777	-0.194940

Exploratory Data Analysis

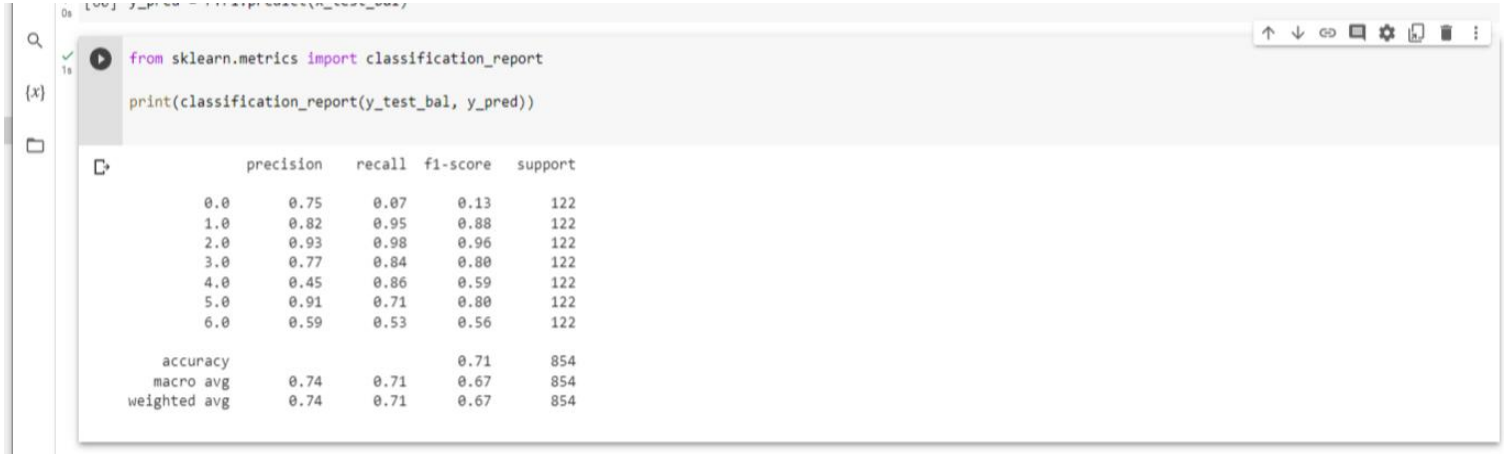




Visual Analysis



Model building



```
from sklearn.metrics import classification_report

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

	precision	recall	f1-score	support
0.0	0.75	0.07	0.13	122
1.0	0.82	0.95	0.88	122
2.0	0.93	0.98	0.96	122
3.0	0.77	0.84	0.80	122
4.0	0.45	0.86	0.59	122
5.0	0.91	0.71	0.80	122
6.0	0.59	0.53	0.56	122
accuracy			0.71	854
macro avg	0.74	0.71	0.67	854
weighted avg	0.74	0.71	0.67	854

XGB Classifier model

```
[62] from sklearn.metrics import accuracy_score
     train_score = accuracy_score(y_bal, rfr1.predict(x_bal))
     train_score

     1.0
```

```
from xgboost import XGBClassifier
xgb1 = XGBClassifier()
xgb1.fit(x_bal,y_bal)
```

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              objective='multi:softprob', predictor=None, ...)
```

thyroid.ipynb - Colaboratory

colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=4Hzv4cCSXy5B

Gmail YouTube Maps

+ Code + Text

RAM Disk

```
[64] y_pred = xgb1.predict(x_test_bal)
```

```
[65] print(classification_report(y_test_bal,y_pred))
```

```
              precision    recall  f1-score   support

         0.0       0.80      0.29      0.42       122
         1.0       0.82      0.93      0.87       122
         2.0       0.96      1.00      0.98       122
         3.0       0.77      0.84      0.80       122
         4.0       0.52      0.84      0.64       122
         5.0       0.84      0.72      0.78       122
         6.0       0.60      0.52      0.56       122

    accuracy                           0.74       854
   macro avg       0.76      0.74      0.72       854
weighted avg       0.76      0.74      0.72       854
```

```
accuracy_score(y_test_bal,y_pred)
```

```
0.7353629976580797
```

SVC Model

```
[67] from sklearn.svm import SVC
     from sklearn.metrics import accuracy_score, classification_report

     svm = SVC()
     svm.fit(x_bal, y_bal)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
  y = column_or_1d(y, warn=True)
```

SVC()

```
y_pred = svm.predict(x_test_bal)
```

thyroid.ipynb - Colaboratory

colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=GZXm9Gdqxk6s

Gmail YouTube Maps

+ Code + Text

RAM Disk

```
[69] print(classification_report(y_test_bal, y_pred))
```

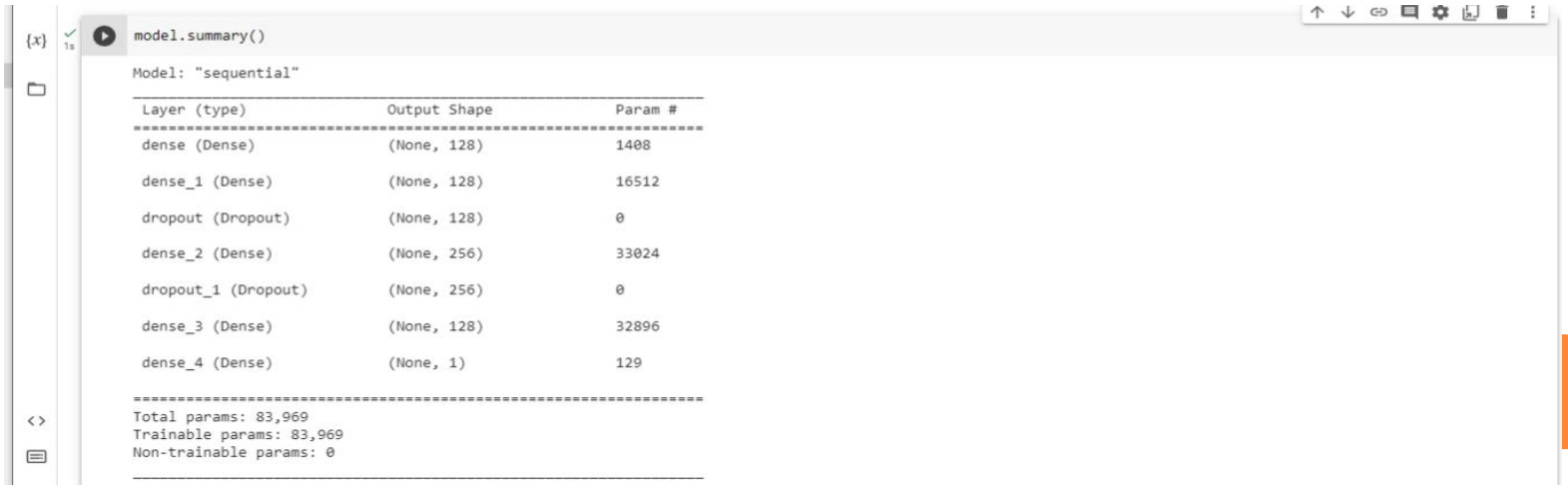
	precision	recall	f1-score	support
0.0	0.70	0.85	0.77	122
1.0	0.78	0.81	0.80	122
2.0	0.88	0.92	0.90	122
3.0	0.73	0.70	0.72	122
4.0	0.73	0.63	0.68	122
5.0	0.83	0.53	0.65	122
6.0	0.51	0.62	0.56	122
accuracy			0.72	854
macro avg	0.74	0.72	0.72	854
weighted avg	0.74	0.72	0.72	854

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

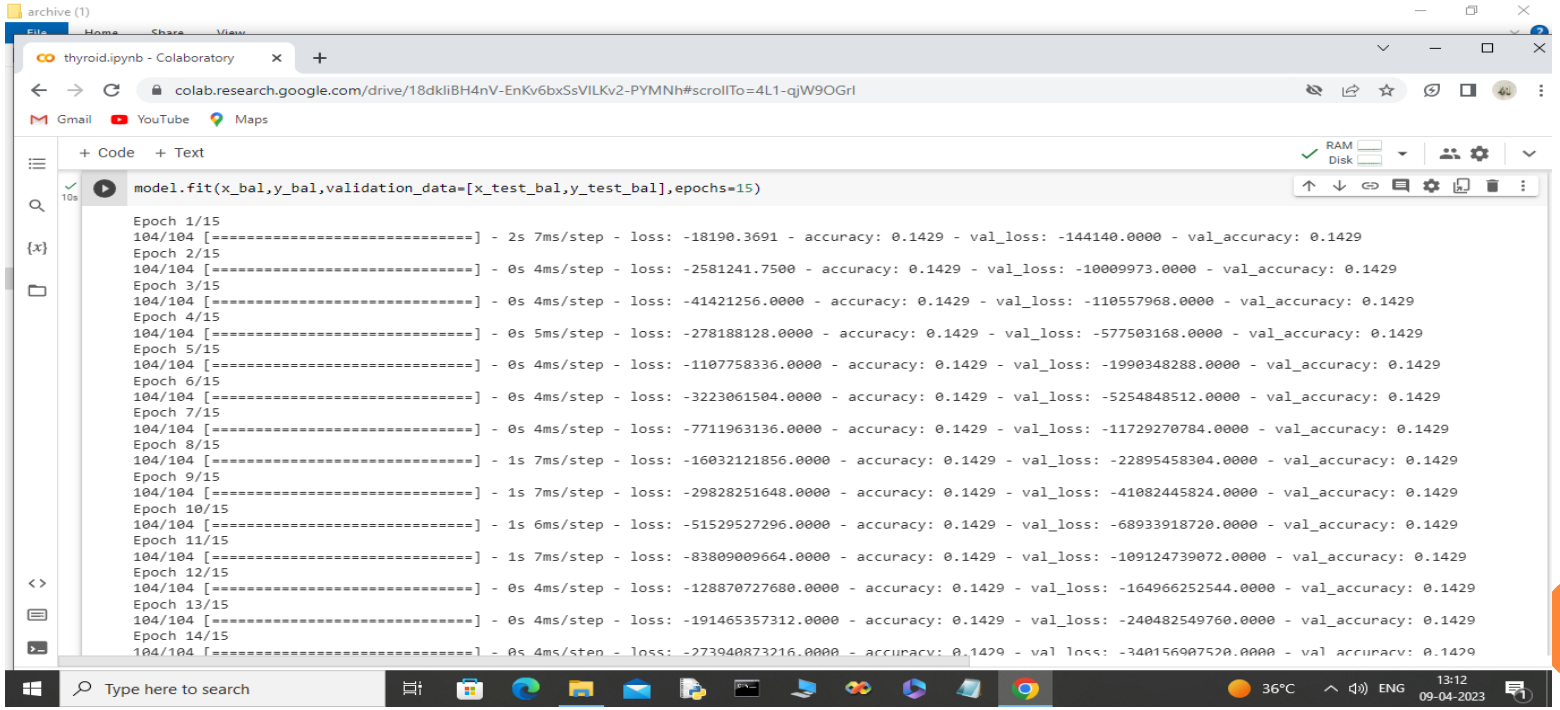
```
0.7209584470730968
```

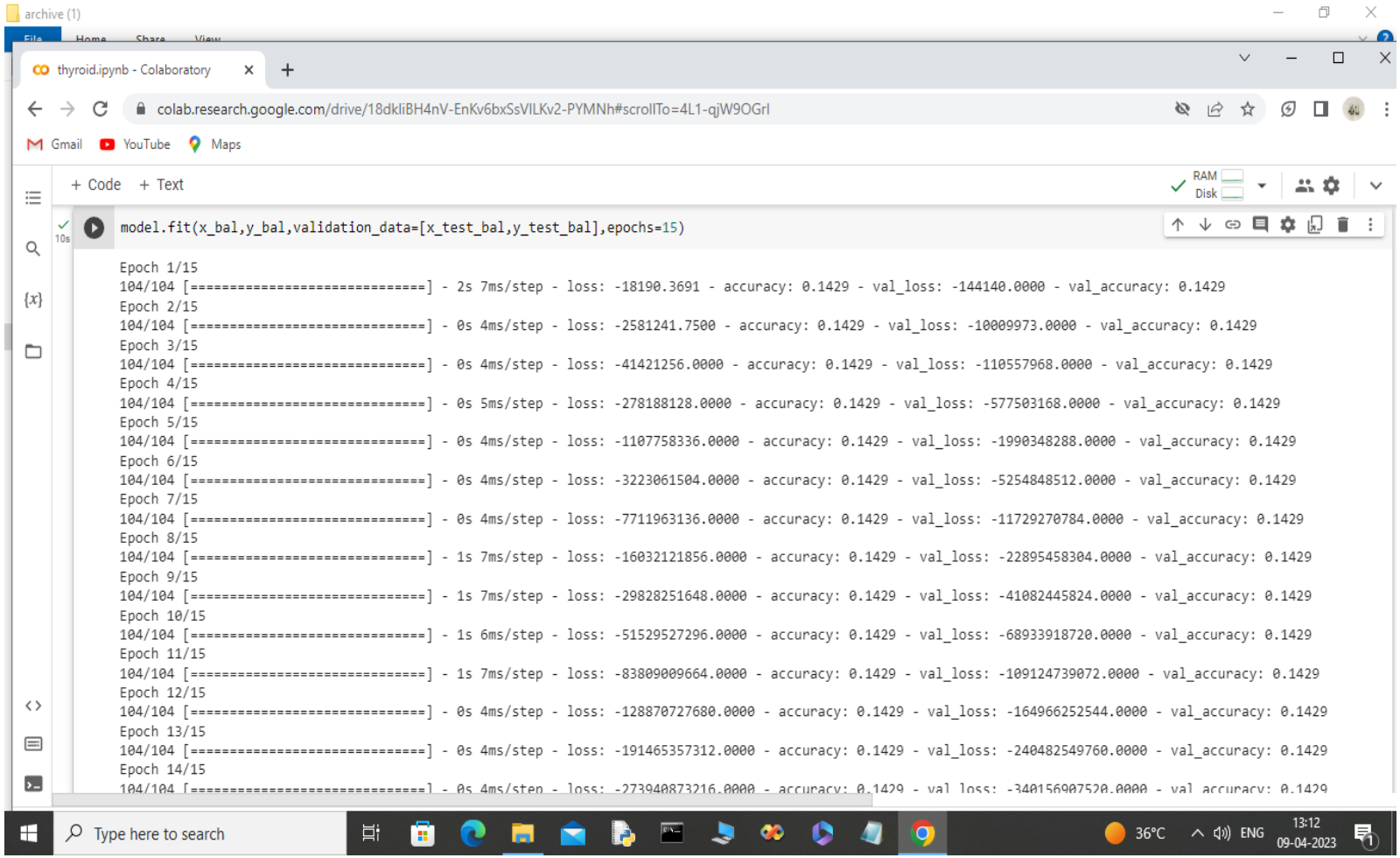
ANN Model


thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=ivZncMnMUKMr
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[71] model= Sequential()
[72] model.add(Dense(units=128,activation='relu',input_shape=(10,)))
[73] 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'))

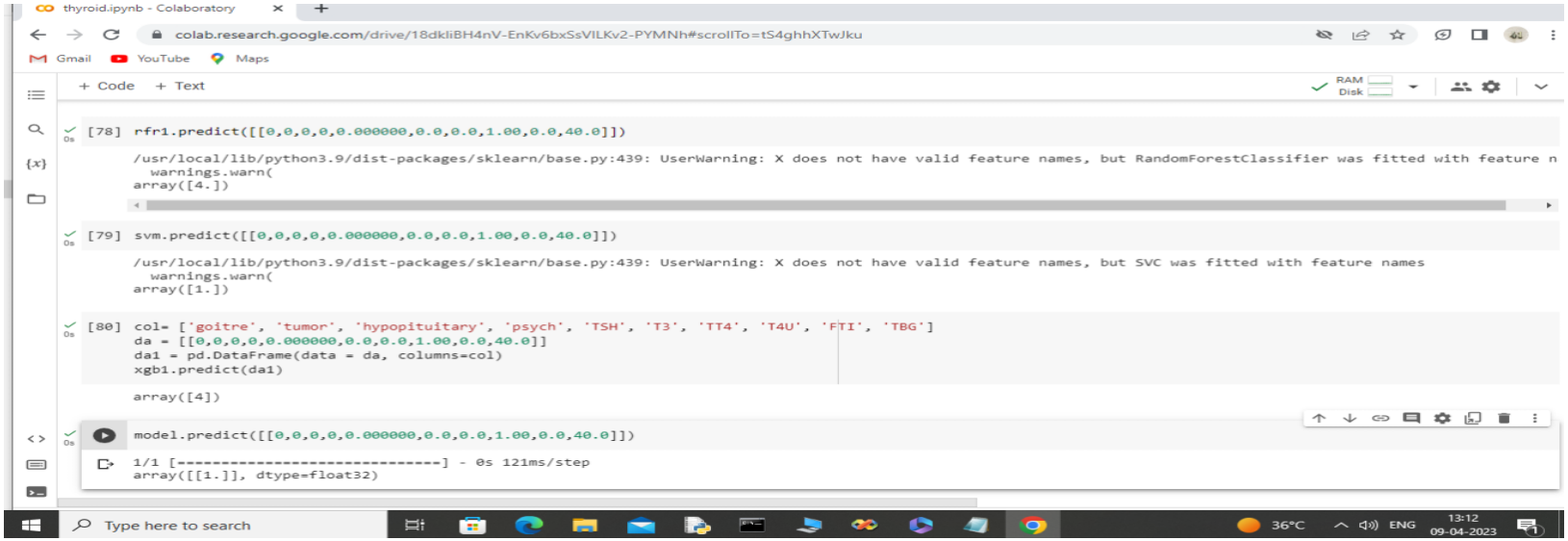

model.summary()
Model: "sequential"
Layer (type) Output Shape Param #
dense (Dense) (None, 128) 1408
dense_1 (Dense) (None, 128) 16512
dropout (Dropout) (None, 128) 0
dense_2 (Dense) (None, 256) 33024
dropout_1 (Dropout) (None, 256) 0
dense_3 (Dense) (None, 128) 32896
dense_4 (Dense) (None, 1) 129
Total params: 83,969
Trainable params: 83,969
Non-trainable params: 0

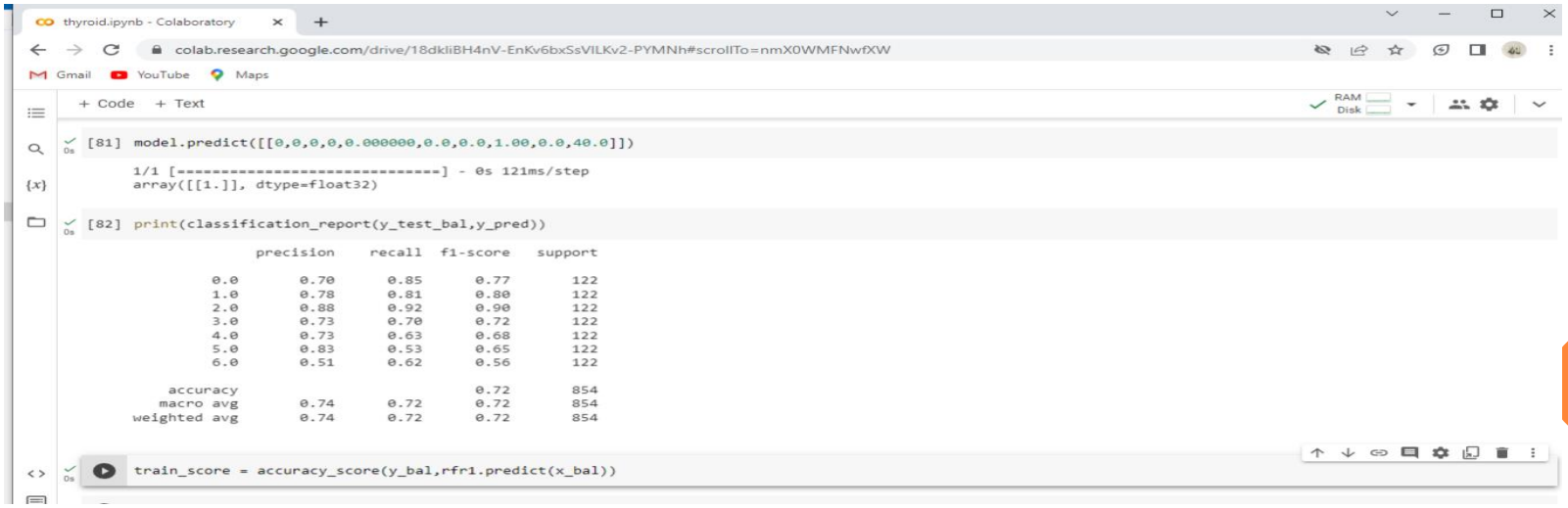

thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=4L1-qjW9OGrI
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[76] model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])


archive (1)
thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=4L1-qjW9OGrI
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
model.fit(x_bal,y_bal,validation_data=[x_test_bal,y_test_bal],epochs=15)
Epoch 1/15
104/104 [==============================] - 2s 7ms/step - loss: -18190.3691 - accuracy: 0.1429 - val_loss: -144140.0000 - val_accuracy: 0.1429
Epoch 2/15
104/104 [==============================] - 0s 4ms/step - loss: -2581241.7500 - accuracy: 0.1429 - val_loss: -10009973.0000 - val_accuracy: 0.1429
Epoch 3/15
104/104 [==============================] - 0s 4ms/step - loss: -41421256.0000 - accuracy: 0.1429 - val_loss: -110557968.0000 - val_accuracy: 0.1429
Epoch 4/15
104/104 [==============================] - 0s 5ms/step - loss: -278188128.0000 - accuracy: 0.1429 - val_loss: -577503168.0000 - val_accuracy: 0.1429
Epoch 5/15
104/104 [==============================] - 0s 4ms/step - loss: -1107758336.0000 - accuracy: 0.1429 - val_loss: -1990348288.0000 - val_accuracy: 0.1429
Epoch 6/15
104/104 [==============================] - 0s 4ms/step - loss: -3223061504.0000 - accuracy: 0.1429 - val_loss: -5254848512.0000 - val_accuracy: 0.1429
Epoch 7/15
104/104 [==============================] - 0s 4ms/step - loss: -7711963136.0000 - accuracy: 0.1429 - val_loss: -11729270784.0000 - val_accuracy: 0.1429
Epoch 8/15
104/104 [==============================] - 1s 7ms/step - loss: -16032121856.0000 - accuracy: 0.1429 - val_loss: -22895458304.0000 - val_accuracy: 0.1429
Epoch 9/15
104/104 [==============================] - 1s 7ms/step - loss: -29828251648.0000 - accuracy: 0.1429 - val_loss: -41082445824.0000 - val_accuracy: 0.1429
Epoch 10/15
104/104 [==============================] - 1s 6ms/step - loss: -51529527296.0000 - accuracy: 0.1429 - val_loss: -68933918720.0000 - val_accuracy: 0.1429
Epoch 11/15
104/104 [==============================] - 1s 7ms/step - loss: -83809009664.0000 - accuracy: 0.1429 - val_loss: -109124739072.0000 - val_accuracy: 0.1429
Epoch 12/15
104/104 [==============================] - 0s 4ms/step - loss: -128870727680.0000 - accuracy: 0.1429 - val_loss: -164966252544.0000 - val_accuracy: 0.1429
Epoch 13/15
104/104 [==============================] - 0s 4ms/step - loss: -191465357312.0000 - accuracy: 0.1429 - val_loss: -240482549760.0000 - val_accuracy: 0.1429
Epoch 14/15
Type here to search
36°C
ENG
13:12
09-04-2023

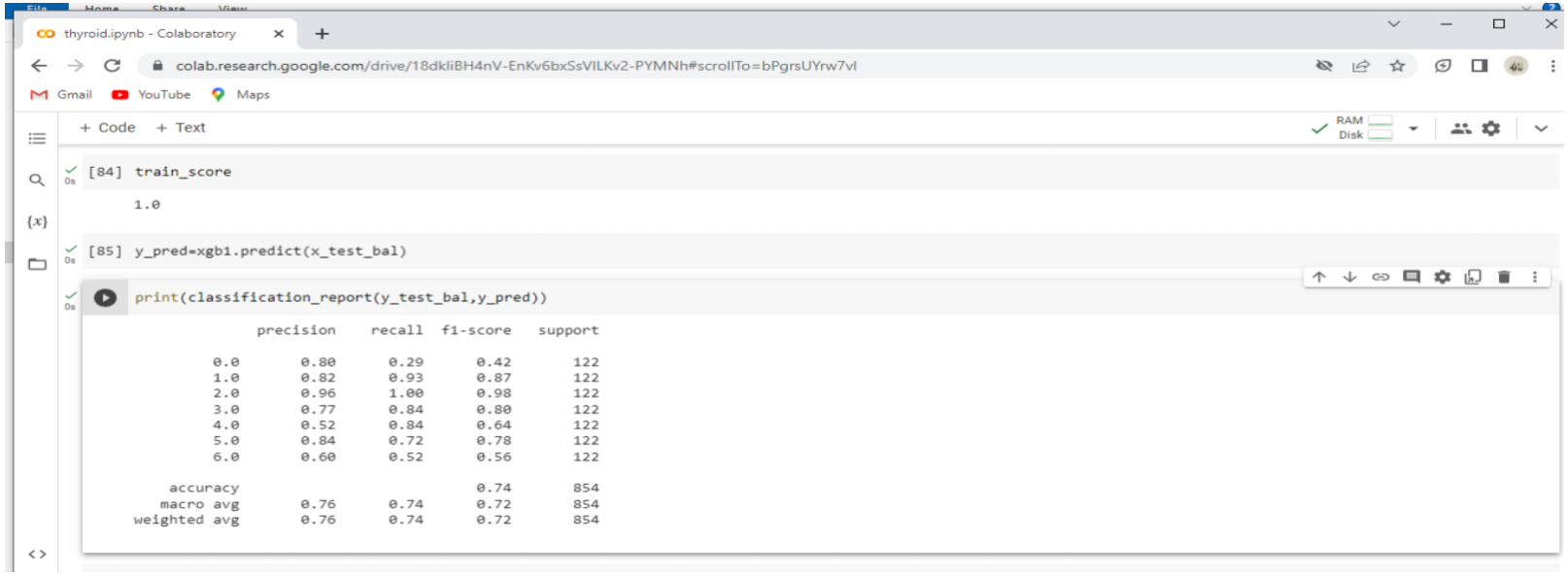

archive (1)
thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=4L1-qjW9OGrl
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
model.fit(x_bal,y_bal,validation_data=[x_test_bal,y_test_bal],epochs=15)
Epoch 1/15
104/104 [==============================] - 2s 7ms/step - loss: -18190.3691 - accuracy: 0.1429 - val_loss: -144140.0000 - val_accuracy: 0.1429
Epoch 2/15
104/104 [==============================] - 0s 4ms/step - loss: -2581241.7500 - accuracy: 0.1429 - val_loss: -10009973.0000 - val_accuracy: 0.1429
Epoch 3/15
104/104 [==============================] - 0s 4ms/step - loss: -41421256.0000 - accuracy: 0.1429 - val_loss: -110557968.0000 - val_accuracy: 0.1429
Epoch 4/15
104/104 [==============================] - 0s 5ms/step - loss: -278188128.0000 - accuracy: 0.1429 - val_loss: -577503168.0000 - val_accuracy: 0.1429
Epoch 5/15
104/104 [==============================] - 0s 4ms/step - loss: -1107758336.0000 - accuracy: 0.1429 - val_loss: -1990348288.0000 - val_accuracy: 0.1429
Epoch 6/15
104/104 [==============================] - 0s 4ms/step - loss: -3223061504.0000 - accuracy: 0.1429 - val_loss: -5254848512.0000 - val_accuracy: 0.1429
Epoch 7/15
104/104 [==============================] - 0s 4ms/step - loss: -7711963136.0000 - accuracy: 0.1429 - val_loss: -11729270784.0000 - val_accuracy: 0.1429
Epoch 8/15
104/104 [==============================] - 1s 7ms/step - loss: -16032121856.0000 - accuracy: 0.1429 - val_loss: -22895458304.0000 - val_accuracy: 0.1429
Epoch 9/15
104/104 [==============================] - 1s 7ms/step - loss: -29828251648.0000 - accuracy: 0.1429 - val_loss: -41082445824.0000 - val_accuracy: 0.1429
Epoch 10/15
104/104 [==============================] - 1s 6ms/step - loss: -51529527296.0000 - accuracy: 0.1429 - val_loss: -68933918720.0000 - val_accuracy: 0.1429
Epoch 11/15
104/104 [==============================] - 1s 7ms/step - loss: -83809009664.0000 - accuracy: 0.1429 - val_loss: -109124739072.0000 - val_accuracy: 0.1429
Epoch 12/15
104/104 [==============================] - 0s 4ms/step - loss: -128870727680.0000 - accuracy: 0.1429 - val_loss: -164966252544.0000 - val_accuracy: 0.1429
Epoch 13/15
104/104 [==============================] - 0s 4ms/step - loss: -191465357312.0000 - accuracy: 0.1429 - val_loss: -240482549760.0000 - val_accuracy: 0.1429
Epoch 14/15
Type here to search
36°C
ENG
13:12
09-04-2023

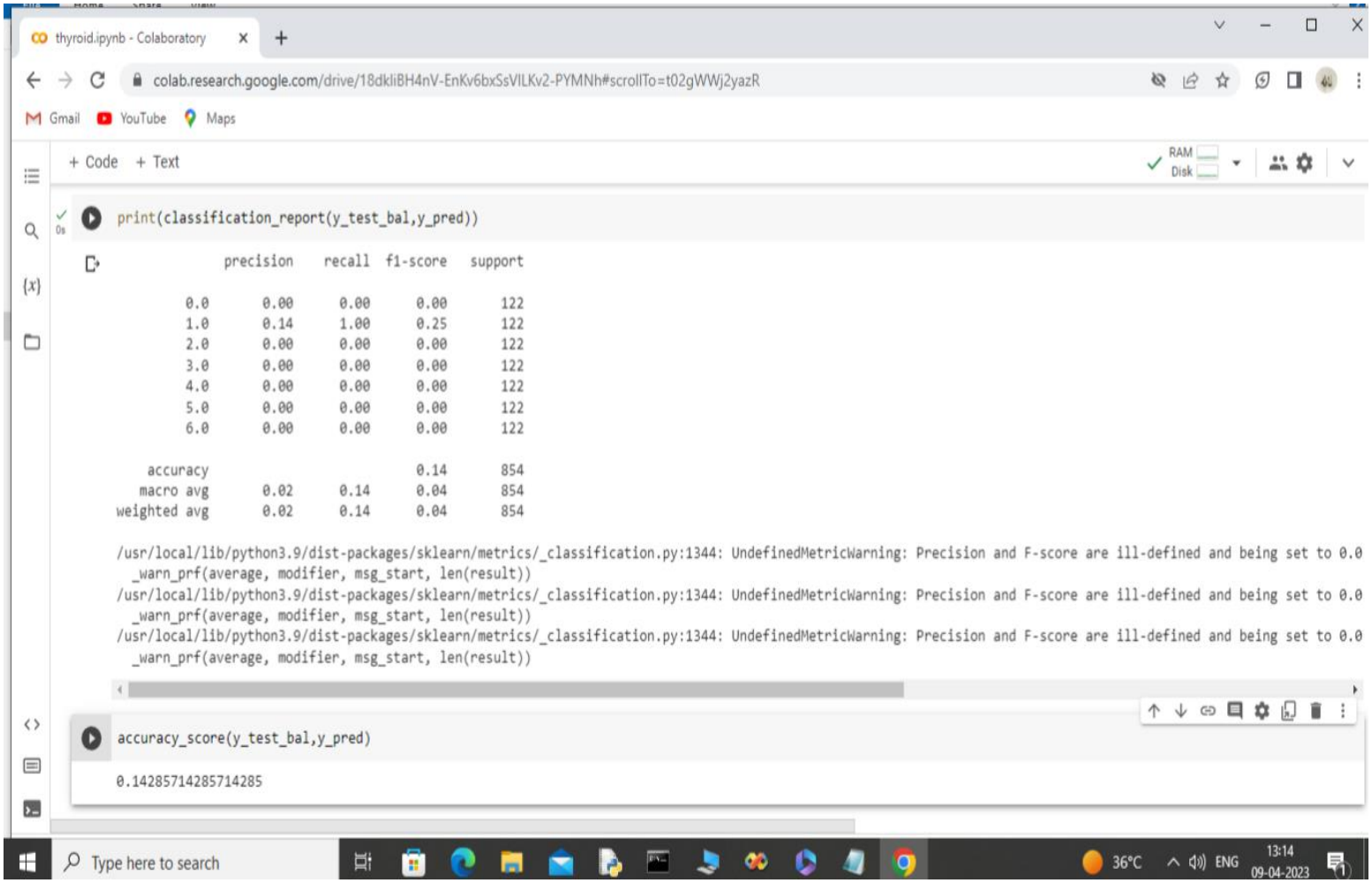
Testing the model





Compare Model

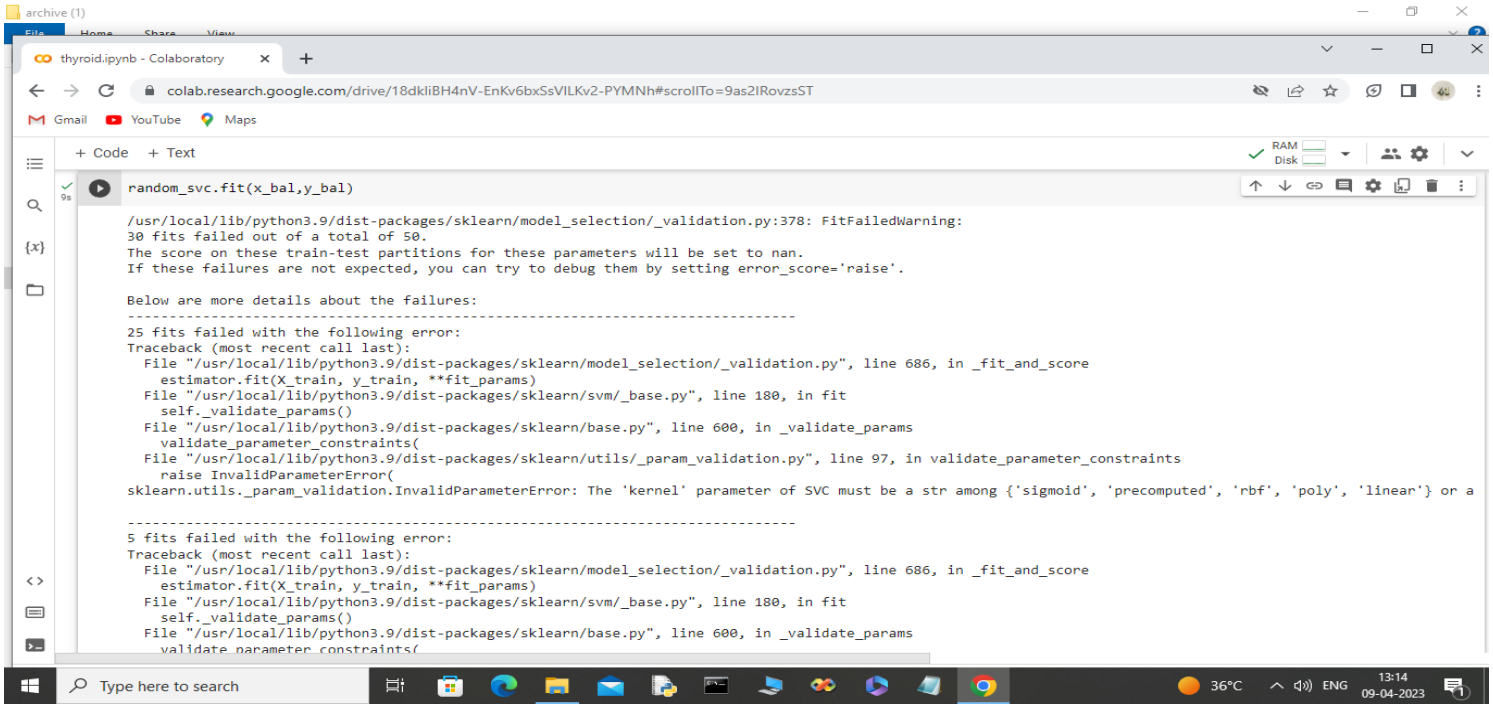


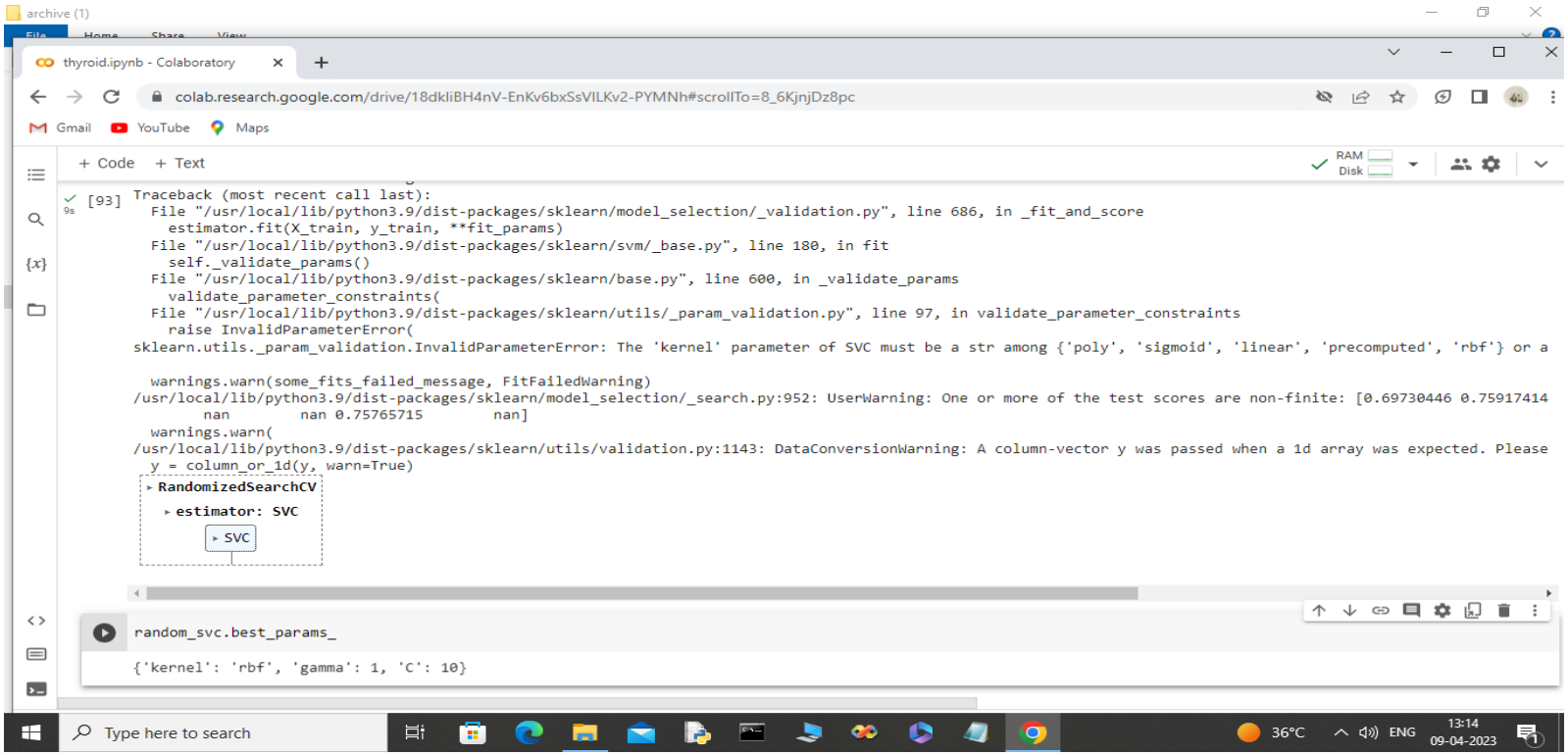

thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=t02gWWj2yazR
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
print(classification_report(y_test_bal,y_pred))
precision recall f1-score support
0.0 0.00 0.00 0.00 122
1.0 0.14 1.00 0.25 122
2.0 0.00 0.00 0.00 122
3.0 0.00 0.00 0.00 122
4.0 0.00 0.00 0.00 122
5.0 0.00 0.00 0.00 122
6.0 0.00 0.00 0.00 122
accuracy 0.14 854
macro avg 0.02 0.14 0.04 854
weighted avg 0.02 0.14 0.04 854
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
_warn_prf(average, modifier, msg_start, len(result))
accuracy_score(y_test_bal,y_pred)
0.14285714285714285
Type here to search
36°C
ENG
13:14
09-04-2023

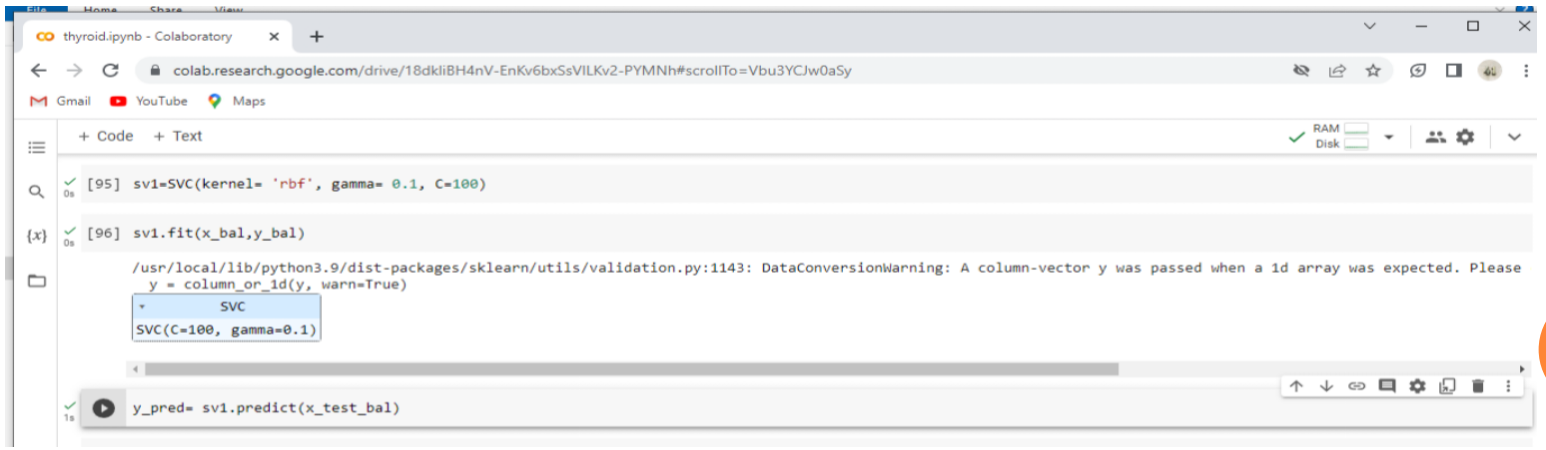
Comparing model accuracy before & after applying hyper parameter tuning

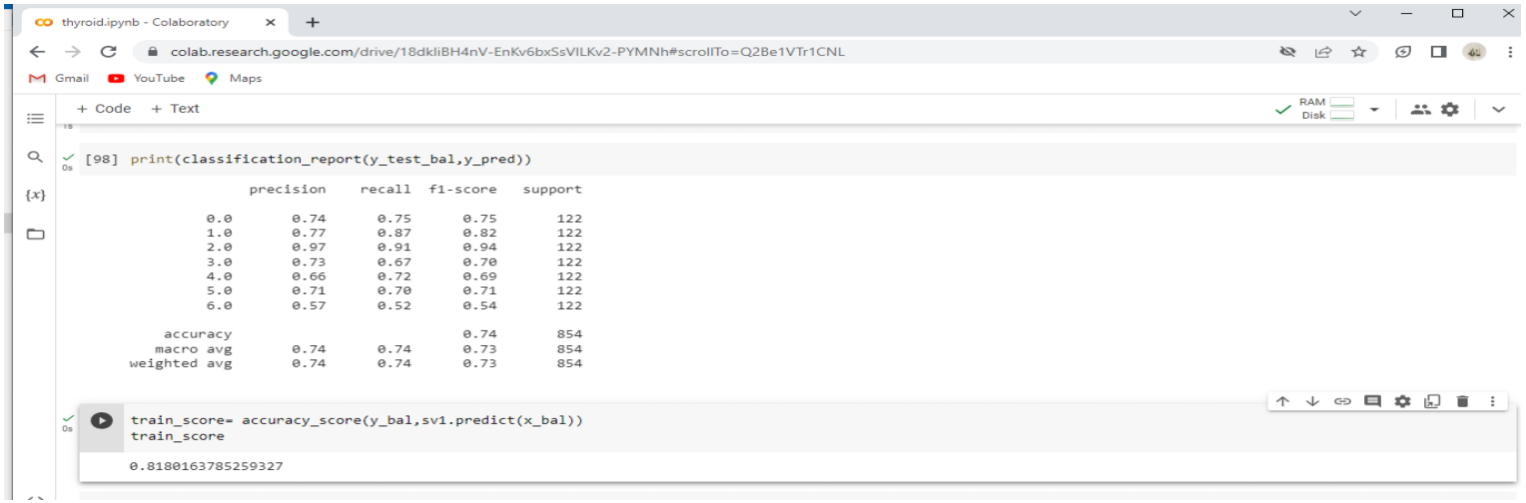
```
[91] params = {

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

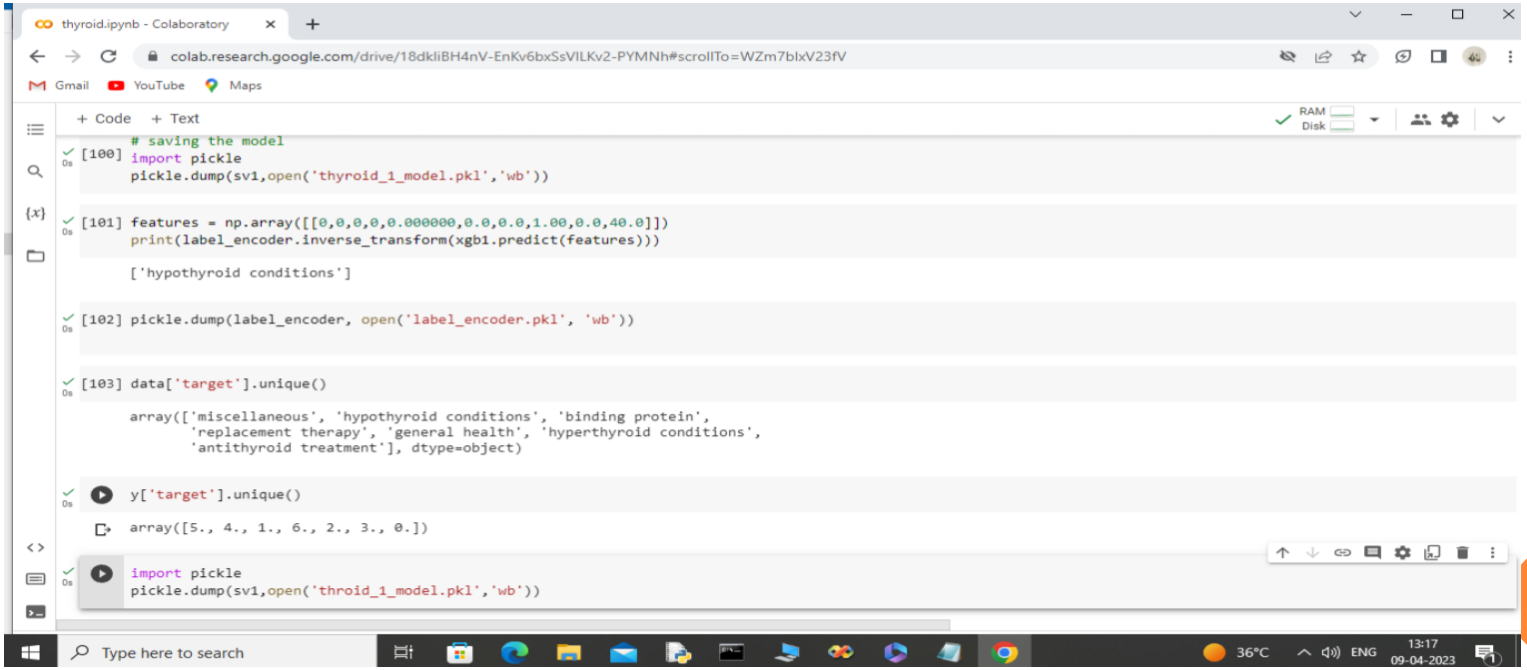
```
from sklearn.model_selection import RandomizedSearchCV
random_svc = RandomizedSearchCV(svm,params, scoring='accuracy',cv=5,n_jobs=-1)
```




archive (1)
thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=8_6KjnjDz8pc
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[93]
9s
Traceback (most recent call last):
File "/usr/local/lib/python3.9/dist-packages/sklearn/model_selection/_validation.py", line 686, in _fit_and_score
estimator.fit(X_train, y_train, **fit_params)
File "/usr/local/lib/python3.9/dist-packages/sklearn/svm/_base.py", line 180, in fit
self._validate_params()
File "/usr/local/lib/python3.9/dist-packages/sklearn/base.py", line 600, in _validate_params
validate_parameter_constraints(
File "/usr/local/lib/python3.9/dist-packages/sklearn/utils/_param_validation.py", line 97, in validate_parameter_constraints
raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'kernel' parameter of SVC must be a str among {'poly', 'sigmoid', 'linear', 'precomputed', 'rbf'} or a
warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.9/dist-packages/sklearn/model_selection/_search.py:952: UserWarning: One or more of the test scores are non-finite: [0.69730446 0.75917414
nan nan 0.75765715 nan]
warnings.warn(
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
y = column_or_1d(y, warn=True)
RandomizedSearchCV
estimator: SVC
SVC
random_svc.best_params_
{'kernel': 'rbf', 'gamma': 1, 'C': 10}
Type here to search
36°C
ENG
13:14
09-04-2023


thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=Vbu3YCJw0aSy
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[95]
0s
sv1=SVC(kernel= 'rbf', gamma= 0.1, C=100)
[96]
0s
sv1.fit(x_bal,y_bal)
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
y = column_or_1d(y, warn=True)
SVC
SVC(C=100, gamma=0.1)
y_pred= sv1.predict(x_test_bal)
1s


thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=Q2Be1VTr1CNL
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
[98] print(classification_report(y_test_bal,y_pred))
precision recall f1-score support
0.0 0.74 0.75 0.75 122
1.0 0.77 0.87 0.82 122
2.0 0.97 0.91 0.94 122
3.0 0.73 0.67 0.70 122
4.0 0.66 0.72 0.69 122
5.0 0.71 0.70 0.71 122
6.0 0.57 0.52 0.54 122
accuracy 0.74 854
macro avg 0.74 0.74 0.73 854
weighted avg 0.74 0.74 0.73 854
train_score= accuracy_score(y_bal,sv1.predict(x_bal))
train_score
0.8180163785259327


thyroid.ipynb - Colaboratory
colab.research.google.com/drive/18dkliBH4nV-EnKv6bxSsVILKv2-PYMNh#scrollTo=WZm7blxV23fV
Gmail
YouTube
Maps
+ Code
+ Text
RAM
Disk
saving the model
[100] import pickle
pickle.dump(sv1,open('thyroid_1_model.pkl','wb'))
[101] 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']
[102] pickle.dump(label_encoder, open('label_encoder.pkl', 'wb'))
[103] data['target'].unique()
array(['miscellaneous', 'hypothyroid conditions', 'binding protein',
'replacement therapy', 'general health', 'hyperthyroid conditions',
'antithyroid treatment'], dtype=object)
y['target'].unique()
array([5., 4., 1., 6., 2., 3., 0.])
import pickle
pickle.dump(sv1,open('throid_1_model.pkl','wb'))
Type here to search
36°C
ENG
13:17
09-04-2023

ADVANTAGES

- Accuracy: Machine learning algorithms can accurately classify thyroid diseases based on patterns and relationships found in large datasets, which can improve accuracy compared to traditional diagnostic methods.
- Speed: Machine learning algorithms can process large amounts of data much faster than humans, reducing the time required for diagnosis and treatment.
- Personalization: Machine learning algorithms can be trained to personalize diagnosis and treatment based on individual patient characteristics, such as age, gender, and medical history.
- Consistency: Machine learning algorithms can provide consistent results across different medical professionals and institutions, reducing the risk of misdiagnosis and improving patient outcomes.

APPLICATION

- Automated diagnosis: Machine learning models can be trained on medical images (such as ultrasound or MRI) to automatically diagnose thyroid disease. This could potentially reduce the need for invasive diagnostic procedures, such as biopsies.

- Predictive modeling: Machine learning models can be used to predict the likelihood of thyroid disease in patients based on their medical history, lifestyle factors, and genetic data. This could help clinicians identify high-risk patients and provide targeted preventive care.

- Treatment optimization: Machine learning models can be used to predict the most effective treatment for individual patients based on their medical history and other factors. This could help optimize treatment outcomes and reduce the risk of adverse effects.

CONCLUSION

In conclusion, machine learning algorithms can be effective in classifying thyroid disease based on various features and characteristics. The accuracy of the classification models depends on the quality and quantity of data used for training, the selection of appropriate features, and the choice of suitable algorithms. Different machine learning algorithms such as Random Forest Classifier, SVM classifier, XGB classifier and neural networks have been used in the literature to classify thyroid diseases. Ensemble methods machine learning algorithms have the potential to assist clinicians in the diagnosis and treatment of thyroid diseases by providing accurate and timely predictions.

FUTURE SCOPE

- Feature engineering: One of the critical aspects of machine learning is feature selection or engineering. Researchers can explore the creation of new features or the extraction of existing features from thyroid disease data. This may include demographic data, medical history, lab results, and imaging data, among others.

- Deep learning models: Deep learning models, such as convolution neural networks (CNNs) and recurrent neural networks (RNNs), can learn high-level representations of the data, leading to more accurate classification. Researchers can explore the use of these models for thyroid disease classification

- Explainable AI: Explainable AI (XAI) aims to make machine learning models more interpretable to humans. Researchers can explore the use of XAI techniques such as LIME, SHAP