```python
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
```

```python
data=pd.read_csv("/content/thyroidDF.csv")
data.head()
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	query
0	29	F	f	f	f	f	f	f	f	
1	29	F	f	f	f	f	f	f	f	
2	41	F	f	f	f	f	f	f	f	
3	36	F	f	f	f	f	f	f	f	
4	32	F	f	f	f	f	f	f	f	

5 rows × 31 columns

```
            'K': 'general health',
            'L': 'replacement therapy',
            'M': 'replacement therapy',
            'N': 'replacement therapy',
            'O': 'antithyroid treatment',
            'P': 'antithyroid treatment',
            'Q': 'antityroid treatment',
            'R': 'miscellaneous',
            'S': 'miscellaneous',
            'T': 'miscellaneous'}
data['target']=data['target'].map(diagnoses)
```

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

```
data['target'].value_counts()
```

```
hypothyroid conditions     593
general health             436
binding protein            376
replacement therapy        336
miscellaneous              281
hyperthyroid coditions     147
hyperthyroid conditions     35
antithyroid treatment       19
antityroid treatment        14
```

Code + Text

```
[ ] x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2237 entries, 4 to 9169
Data columns (total 30 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   age                   2237 non-null   float64
 1   sex                   2237 non-null   object
 2   on_thyroxine          2237 non-null   object
 3   query_on_thyroxine    2237 non-null   object
 4   on_antithyroid_meds   2237 non-null   object
 5   sick                  2237 non-null   object
 6   pregnant              2237 non-null   object
 7   thyroid_surgery       2237 non-null   object
 8   I131_treatment        2237 non-null   object
 9   query_hypothyroid     2237 non-null   object
 10  query_hyperthyroid    2237 non-null   object
 11  lithium               2237 non-null   object
 12  goitre                2237 non-null   object
 13  tumor                 2237 non-null   object
 14  hypopituitary         2237 non-null   object
 15  psych                 2237 non-null   object
 16  TSH_measured          2237 non-null   object
 17  TSH                   2087 non-null   float64
 18  T3_measured           2237 non-null   object
 19  T3                    1643 non-null   float64
 20  TT4_measured          2237 non-null   object
```

+ Code + Text

RAM Disk

[ ]

3	3
4	4
...	...
2232	2232
2233	2233
2234	2234
2235	2235
2236	2236

2237 rows × 1 columns

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

[ ]
```
x
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	qu
4	32	F	f	f	f	f	f	f	f	

+ Code + Text

RAM
Disk

```
[ ] from sklearn.preprocessing import StandardScaler
    sc = StandardScaler()
    x_bal = sc.fit_transform(x_bal)
    x_test_bal= sc.transform(x_test_bal)

[ ] x_bal

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

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

[ ] x_bal

[ ] from sklearn.inspection import permutation_importance
    results=permutation_importance(rfr,x_bal,y_bal,scoring='accuracy')

[ ] feature_importance=['age','sex','on_thyroxine','query_on_thyroxine','on_antithyroid_meds','sick','pregnant','thyroid_sur
    importance=results.importances_mean
    importance=np.sort(importance)
    for i,v in enumerate(importance):
```

+ Code + Text

RAM
Disk

```
[ ] y_pred=xgb1.predict(x_test_os)

[ ] print(classification_report(y_test_os,y_pred))

[ ] accuracy_score(y_test_os,y_pred)

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

[ ] sv.fit(x_bal,y_bal)

[ ] y_pred=sv.predict(x_test_bal)

[ ] print(classification_report(y_test_bal,y_pred))

[ ] train_score=accuracy_score(y_bal,sv.predict(x_bal))
    train_score
```

+ Code + Text

RAM Disk

```
[ ] col=['goitre','tumor','hypopituitary','psych','TSH','T3','TT4','T4U','FTI','TBG']
    da=[[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]]
    da1=pd.DataFrame(data=da,columns=col)
    xgb1.predict(da1)

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

[ ] print(classification_report(y_test_bal,y_pred))

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

[ ] train_score

[ ] y_pred=xgb.predict(x_test_bal)

[ ] print(classification_report(y_test_bal,y_pred))

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

+ Code + Text

RAM
Disk

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

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

[ ] y_pred=model.predict(x_test_bal)

[ ] print(classification_report(y_test_bal,y_pred))

[ ] accuracy_score(y_test_bal,y_pred)

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

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

[ ] random.svc.fit(x_bal,y_bal)
```

```
[ ] random_svc.best_params_

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

[ ] sv1.fit(x_bal,y_bal)

[ ] y_pred=sv1.predict(x_test_bal)

[ ] print(classification_report(y_test_bal,y_pred))

[ ] train_score=accuracy_score(y_bal,sv1.predict(x_bal))
    trsin_score

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

[ ] features=np.array([[0,0,0,0,0.000000,0.0,0.0,1.00,0.0,40.0]])
    print(label_encoder.inverse_transform(xgb1.predict(features)))
```

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

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

```
[ ] In [32]: runfile('C:/Users/SmartBridge-PC/Downloads/Thyroid/app.py',wdir='C:/users/SmartBridge-PC/Downloads/Thyroid')
```

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

[ ] data['target'].unique()

[ ] y['target'].unique()

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

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

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

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

+ Code + Text

RAM
Disk

```
[ ] model.add(Dense(units=128,activation='relu',input_shape=(10,)))

[ ] model.add(Dense(units=128.activation='relu',kernel_initializer='random_uniform'))
    model.add(Dropout(0.2))
    model.add(Dense(units=256,activation='relu'kernel_initializer='random_uniform'))
    model.add(Dropout(0.2))
    model.add(Dense(units=128,activation='relu'kernel_initializer='random_uniform'))

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

[ ] model.summary()

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

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

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

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

+ Code + Text

RAM
Disk

```
[ ]     i=feature_importance[i]
        print('feature:{:<20}score:{}'.format(i,v))
    plt.figure(figsize=(10,10))
    plt.bar(x=feature_importance,height=importance)
    plt.xticks(rotation=30,ha='right')
    plt.show()
```

```
[ ] x.head()
```

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

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

```
[ ] x_bal.head()
```

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

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

+ Code + Text

RAM
Disk

```
[ ] from sklearn.ensemble import RandomForestClassifier
    rfr1=RandomForestClassifier().fit(x_os,y_os.values.ravel())
    y_pred=rfr1.predict(x_test_os)
    rfr1=RandomForestClassifier()

[ ] rfr1.fit(x_os,y_os.values.ravel())

[ ] y_pred=rfr1.predict(x_test_os)

[ ] y_pred=rfr1.predict(x_test_os)

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

[ ] train_score=accuracy_score(y_os,rfr1.predict(x_os))
    train_score

[ ] from xgboost import XGBClassifier
    xgb1 = XGBClassifier()
    xgb1.fit(x_os,y_os)
```

```
[ ] 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()
```

```
    841108102    1
    850503041    1
    841120013    1
    850102019    1
    850530045    1
                ..
    850509071    1
    860916057    1
    850527065    1
    861118071    1
    860707017    1
    Name: patient_id, Length: 1789, dtype: int64
```

```
[ ] 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)
```

[ ]

9162	36	F	f	f	f	f	f	f	f	f
9169	69	M	f	f	f	f	f	f	f	f

2237 rows × 30 columns

[ ]
```
label_encoder=LabelEncoder()
y_dt=label_encoder.fit_transform(y)
```

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

[ ]
```
y
```

	target
0	0
1	1
2	2

Code + Text

RAM
Disk

32	41	M	f	f	f	f	f	f	f
33	71	F	t	f	f	f	f	f	f
39	55	F	t	f	f	f	f	f	f
...	...	...	...	...	...	...	...	...	...
9153	64	M	f	f	f	f	f	f	f
9157	60	M	f	f	t	f	f	f	f
9158	64	M	f	f	f	f	f	f	f
9162	36	F	f	f	f	f	f	f	f
9169	69	M	f	f	f	f	f	f	f

2237 rows × 30 columns

```
[ ]  21  TT4                  2140 non-null   float64
     22  T4U_measured         2237 non-null   object
     23  T4U                  2059 non-null   float64
     24  FTI_measured         2237 non-null   object
     25  FTI                  2060 non-null   float64
     26  TBG_measured         2237 non-null   object
     27  TBG                  98 non-null     float64
     28  referral_source      2237 non-null   object
     29  target               2237 non-null   object
    dtypes: float64(7), object(23)
    memory usage: 541.8+ KB
```

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

```
[ ] x
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	qu
4	32.0	0	0	0	0	0	0	0	0	
18	63.0	0	1	0	0	1	0	0	0	
32	41.0	1	0	0	0	0	0	0	0	

+ Code + Text

x

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment
4	32	F	f	f	f	f	f	f	f
18	63	F	t	f	f	t	f	f	f
32	41	M	f	f	f	f	f	f	f
33	71	F	t	f	f	f	f	f	f
39	55	F	t	f	f	f	f	f	f
...	...	...	...	...	...	...	...	...	...
9153	64	M	f	f	f	f	f	f	f
9157	60	M	f	f	t	f	f	f	f
9158	64	M	f	f	f	f	f	f	f
9162	36	F	f	f	f	f	f	f	f

+ Code + Text

RAM Disk

[ ]

33	71.0	0	1	0	0	0	0	0	0
39	55.0	0	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
9153	64.0	1	0	0	0	0	0	0	0
9157	60.0	1	0	0	1	0	0	0	0
9158	64.0	1	0	0	0	0	0	0	0
9162	36.0	0	0	0	0	0	0	0	0
9169	69.0	1	0	0	0	0	0	0	0

2237 rows × 30 columns

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

+ Code + Text

RAM
Disk

```
data['target'].value_counts()
```

```
hypothyroid conditions     593
general health             436
binding protein            376
replacement therapy        336
miscellaneous              281
hyperthyroid coditions     147
hyperthyroid conditions     35
antithyroid treatment       19
antityroid treatment        14
Name: target, dtype: int64
```

```
data[data.age>100]
```

age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	query_

0 rows × 31 columns

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

+ Code + Text

RAM
Disk

9153	64	M	f	f	f	f	f	f	f
9157	60	M	f	f	t	f	f	f	f
9158	64	M	f	f	f	f	f	f	f
9162	36	F	f	f	f	f	f	f	f
9169	69	M	f	f	f	f	f	f	f

2237 rows × 30 columns

```
[ ] x['sex'].unique()

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

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

[ ] x['sex'].value_counts()
```

+ Code + Text

Disk

x

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery	I131_treatment	qu
4	32	F	f	f	f	f	f	f	f	
18	63	F	t	f	f	t	f	f	f	
32	41	M	f	f	f	f	f	f	f	
33	71	F	t	f	f	f	f	f	f	
39	55	F	t	f	f	f	f	f	f	
...	...	...	...	...	...	...	...	...	...	
9153	64	M	f	f	f	f	f	f	f	
9157	60	M	f	f	t	f	f	f	f	
9158	64	M	f	f	f	f	f	f	f	

+ Code + Text

RAM
Disk

```
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
referral_source      0
target               0
patient_id           0
dtype: int64
```

```
diagnoses={'A': 'hyperthyroid coditions',
           'B': 'hyperthyroid conditions',
           'C': 'hyperthyroid conditions',
           'D': 'hyperthyroid conditions',
           'E': 'hypothyroid conditions',
           'F': 'hypothyroid conditions',
           'G': 'hypothyroid conditions',
           'H': 'hypothyroid conditions',
           'I': 'binding protein',
           'J': 'binding protein',
           'K': 'general health',
```

+ Code + Text

```
data.shape
```

```
(9172, 31)
```

```
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
```