THYROID DISEASE CLASSIFICATION USING MACHINE LEARNING

Submitted in partial fulfillment of requirements for the award of the Degree

Bachelor of Computer Science

In the faculty of Computer Science of Bharathiar University, Coimbatore

Submitted
by

TEAM ID:
NM2023TMID32702

J. JAYAPRAKASH

M. JAYACHANDRAN

P. KARUPPAN

K. MANIKANDAN

P. KARUPPAN

TABLE OF CONTENTS

1. INTRODUCTION

THYROID DISEASE CLASSIFICATION USING ML

The thyroid gland is a small organ that's located in the front of the neck, wrapped around the windpipe (trachea). It's shaped like a butterfly, smaller in the middle with two wide wings that extend around the side of your throat. The thyroid is a gland.The two main types of thyroid disease are hypothyroidism and hyperthyroidism. Both conditions can be caused by other diseases that impact the way the thyroid gland works.

Thyroid disease can affect anyone — men, women, infants, teenagers and the elderly. It can be present at birth (typically hypothyroidism) and it can develop as you age (often after menopause in women).

Symptoms of an overactive thyroid (hyperthyroidism) can include Experiencing anxiety, irritability and nervousness, Having trouble sleeping, Losing weight, Having an enlarged thyroid gland or a goiter, Having muscle weakness and tremors, Experiencing irregular menstrual periods or having your menstrual

cycle stop, Feeling sensitive to heat and Having vision problems or eye irritation.

Symptoms of an underactive thyroid (hypothyroidism) can includeFeeling tired (fatigue), Gaining weight, Experiencing forgetfulness, Having frequent and heavy menstrual periods, Having dry and coarse hair, Having a hoarse voice and Experiencing an intolerance to cold temperatures.

Data cleansing methods were used to make the data primitive enough for the analytics to show the risk of patients getting this disease; Machine Learning plays a very deciding role in disease prediction. Machine Learning algorithms (SVM, Random Forest Classifier, K-NN, logistic regression andDT Classifier 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.

2. PROBLEM SELECTION

Predicting thyroid disease using machine learning is an interesting and important project that can have significant implications in the field of healthcare. Here are some steps you can follow to select a suitable project:

1. Research existing work: Before starting any project, it's important to research existing work in the field. Look for similar projects that have been done before and see what approaches were used. This will help you understand the current state-of-the-art and identify any gaps in the literature that your project can address.
2. Define the problem: Once you have a good understanding of the current research, define the problem you want to solve. For example, you could focus on predicting the risk of developing thyroid disease in a particular population or predicting the progression of the disease in patients who have already been diagnosed.
3. Gather data: In order to train a machine learning model, you will need a dataset. Look for publicly available datasets or consider

collecting your own data. Make sure that the data you use is of high quality and is representative of the problem you are trying to solve.

4. Train and evaluate the model: Once you have chosen an algorithm, train the model on your dataset and evaluate its performance using appropriate metrics such as accuracy, precision, recall, and F1 score. If the model's performance is not satisfactory, consider tweaking the algorithm or using a different one.
5. Deploy the model: Once you have a model that performs well, deploy it in a real-world setting. This could involve integrating it into an electronic health.

3. IDEATION

i. **Collect Data:** Collect data from various sources, including medical records, patient information, and clinical studies. Ensure that the data is in a structured format and that it contains all the necessary features for building a model.

ii. **Pre-process the Data**: Once you have the data, pre-process it by cleaning, normalizing, and transforming it into a format that can be used by a machine learning model.

iii. **Feature Selection**: Choose the relevant features that will be used to predict the thyroid disease. These features can include patient demographics, lab test results, and medical history.

iv. **Choose a Model**: Choose an appropriate machine learning algorithm that will be used for prediction. There are various algorithms available, including decision trees, random forests, and support vector machines.

v. **Train the Model:** Use the pre-processed data and the chosen algorithm to train the model. Split the data into training and testing sets to evaluate the model's accuracy.

vi. **Evaluate the Model:** Evaluate the model's performance using different metrics such as accuracy, precision, recall, and F1-score.

vii. **Deploy the Model:** Finally, deploy the model into a production environment where it can be used to predict the thyroid disease of new patients.

4. REQUIREMENT ANALYSIS

Data Collection:

Gather a large dataset of patient information, including demographics, medical history, symptoms, lab test results, and imaging data (if available). The dataset should include both positive and negative cases of thyroid disease to ensure the model is trained on a balanced dataset.

Data Pre-processing:

Clean the data, remove any missing or irrelevant information, and encode categorical variables.

Feature Selection:

Identify the most important features that contribute to the prediction of thyroid disease.

Algorithm Selection:

Select the appropriate machine learning algorithm for the task, such as logistic regression, decision trees, random forests, or support vector machines.

Model training:

Split the dataset into training and validation sets, train the model on the training set, and evaluate the performance on the validation set. Adjust the model hyper-parameters to optimize performance.

Model Evaluation:

Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score. Use cross-validation to ensure the model's generalizability to new data.

Model Deployment:

Deploy the trained model in a user-friendly interface, such as a web application or mobile app, to enable healthcare professionals to make accurate predictions of thyroid disease in their patients.

Maintenance and Updates:

Regularly update the model with new data and retrain as necessary to ensure the accuracy of the predictions.

6. INPUT DESIGN

The input design is the process of entering data to the system. The input design goal is to enter to the computer as accurate as possible. Here inputs are designed effectively so that errors made by the operation are minimized.

The inputs to the system have been designed in such a way that manual forms and the inputs are coordinated where the data elements are common to the sources document and to the input. The input is acceptable and understandable by the users who are using it.

Input design is the process of converting user-originated inputs to a computer-based format input data are collected and organized into group of similar data. Once identified, appropriate input media are selected for processing.

Input design means the physical and performance requirements of a device that are used as a basis for device design.Input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc

5.1. INPUT DESIGN DESCRIPTION

HOME

In the home page a simple concept of thyroid is displayed and it shows a predict option to get to the predict page.

PREDICT

In this page it asks for input of certain data that is necessary to predict the thyroid type it also shows a submit button which redirects it to the submit page.

7. OUTPUT DESIGN

A design output is a drawing or specification or manufacturing instruction. Design outputs describe all the components, parts, and pieces that go into your device. Design outputs describe all assemblies and subassemblies of product.

Output design is the process of converting data into hard copy that is understood by all. The various outputs have been in such a way that they represent the same format that the office and management used to.

Computer output is the most important and direct source of information to the user. Efficient, intelligible output design should improve the systems relationships with the user and help in decision making. A major form of out is the hardcopy from the printer. Output requirements are designed during system analysis.

6.1. OUTPUT DESIGN DESCRIPTION

SUBMIT

In this page after enter necessary data in predict page then click the submit button. It displays the result of the data entered like (Normal, hypothyroid and hyperthyroid).

7. DESCRIPTION OF MODULES

Modules are unit of code written in access basic language.

- HOME
- PREDICT
- SUBMIT

HOME

- In the home page a simple concept of thyroid is displayed and it shows a predict option to get to the predict page.

PREDICT

- In this page it asks for input of certain data that is necessary to predict the thyroid type.

SUBMIT

- In this page the result of the predicted thyroid type is displayed like whether the user is normal, hyperthyroid or hypothyroid.

8. PROJECT PLANNING PHASE

Define the problem:

Define the problem you want to solve. For example, you maywant to build a model that can accurately predict the risk of thyroid disease in patients based on certain clinical and demographic features.

Gather Data:

Identify the data sources that can be used to build the machine learning model. This may involve gathering data from electronic health records, medical imaging, or patient surveys.

Choose a Machine Learning Algorithm:

There are several machine learning algorithms that can be used to build a predictive model. You will need to choose an algorithm that is appropriate for your data and problem.

Train the model:

Once you have chosen an algorithm, you will need to train the model on the data. This involves dividing the data into training and validation sets, and using the training set to optimize the model's parameters.

Evaluate the model:

After training the model, you will need to evaluate its performance on the validation set. This will give you an idea of how well the model will perform on new data.

Deploy the model:

Once you have a model that performs well on the validation set, you can deploy it in a clinical setting. This may involve integrating it into an electronic health record system, or creating a standalone application.

9. PROJECT DEVELOPMENT PHASE

Data Collection and Preparation:

This involves gathering relevant data from various sources and preparing it for use in machine learning algorithms. In the case of thyroid disease prediction, this might include medical records, lab results, and patient demographics.

Feature Selection and Engineering:

Once the data has been collected, the next step is to select the most relevant features (i.e., variables or attributes) that are likely to be predictive of thyroid disease. This might involve domain expertise from medical professionals, as well as statistical techniques such as correlation analysis or principal component analysis.

Model Selection and Training:

With the features selected, the next step is to choose an appropriate machine learning algorithm to use for prediction. This might include traditional models such as logistic regression or more advanced methods such as deep learning

Model Evaluation and Tuning:

Once the model has been trained, it is evaluated using various metrics such as accuracy, precision, and recall

Deployment and Monitoring:

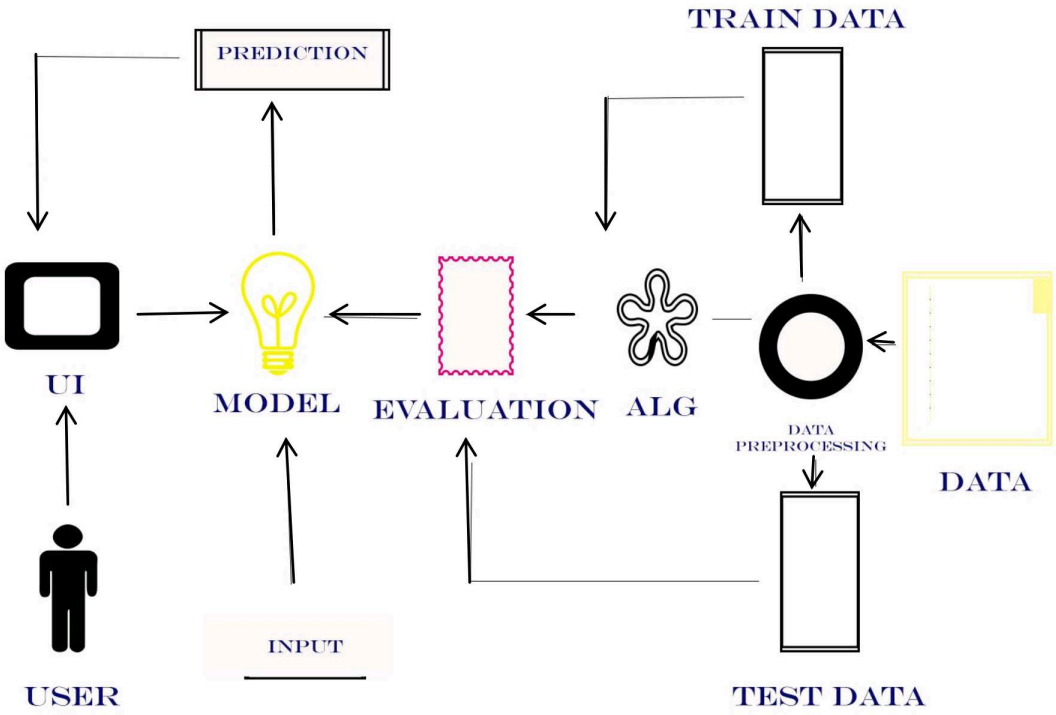
Finally, the trained and validated model can be deployed for use in clinical settings. Ongoing monitoring and evaluation of the model's performance are critical to ensure that it remains accurate and effective over time.

10. CONCLUSION

Machine Learning plays a very deciding role in disease prediction. Machine Learning algorithms such as SVM, Random Forest Classifier, K-NN, logistic regression andDT Classifier 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 thyroid disease. It predicts with 84% accuracy score.

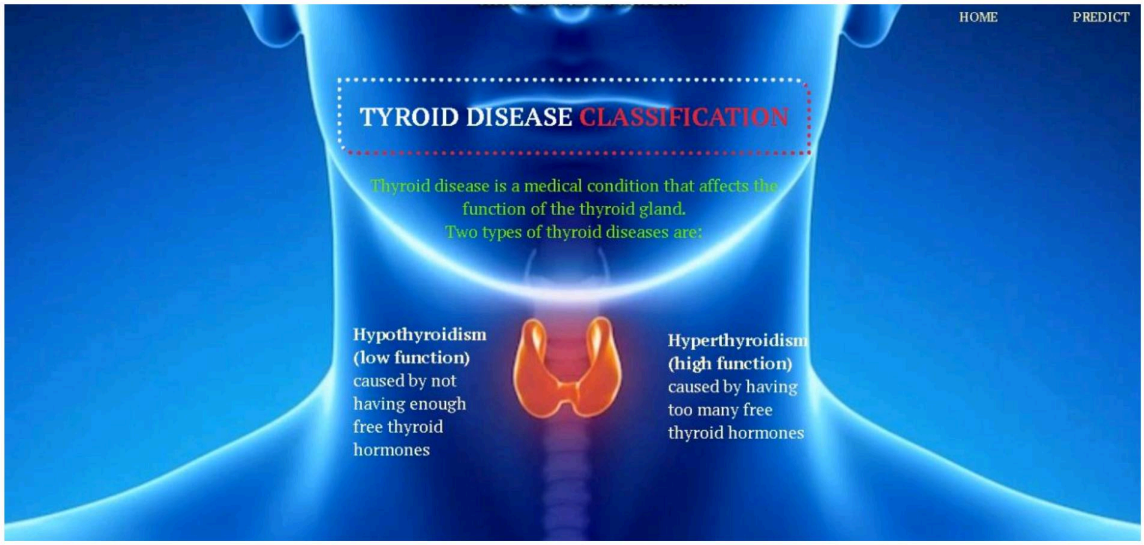
APPENDICES

A. TECHNICAL FLOW

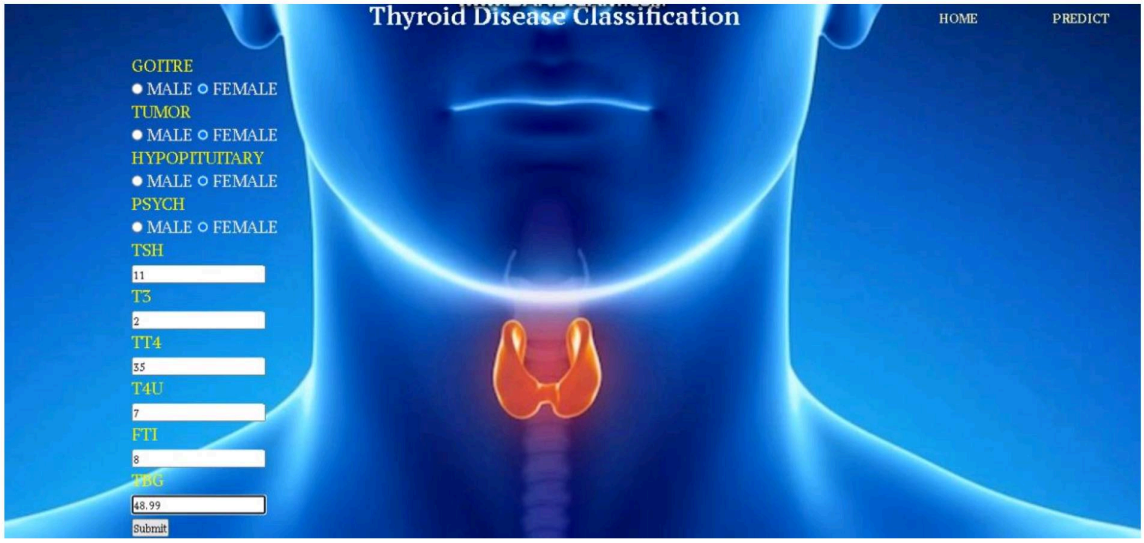


B. SAMPLE INPUT

HOME

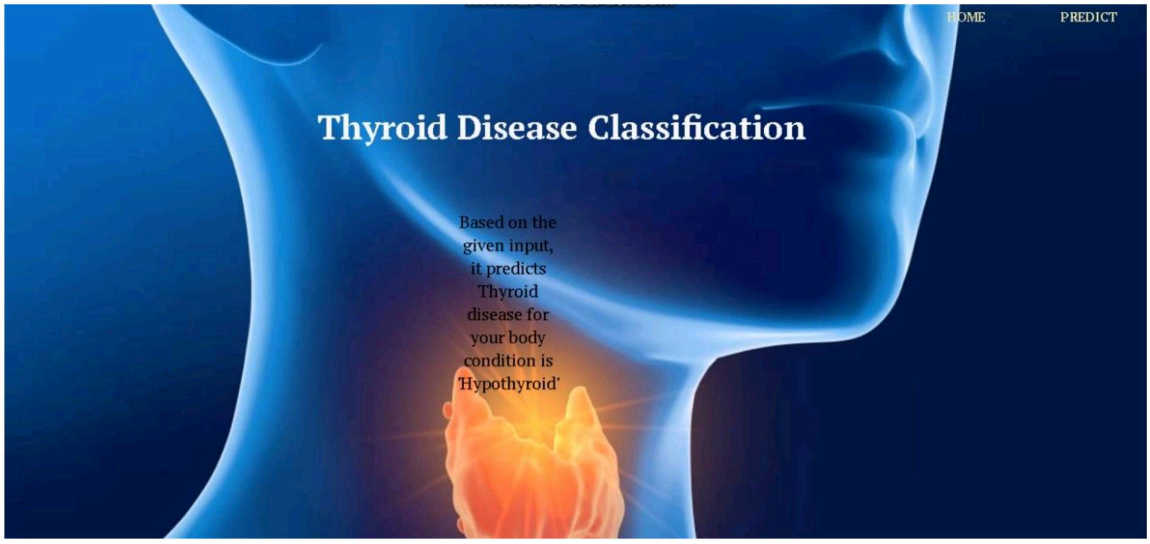


PREDICT

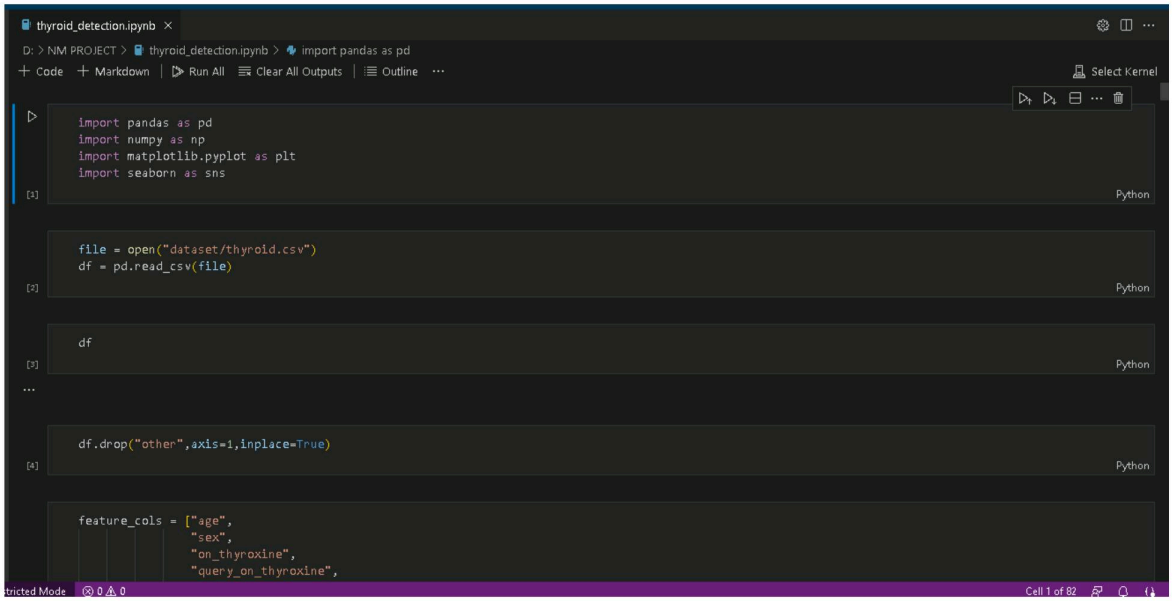


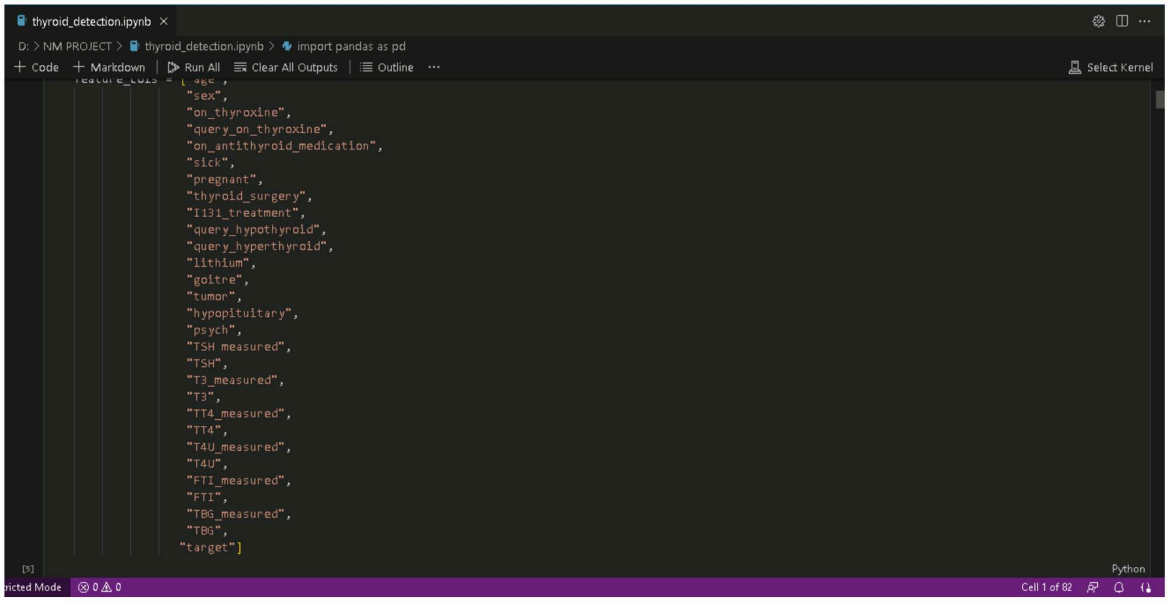
C. SAMPLE OUTPUT

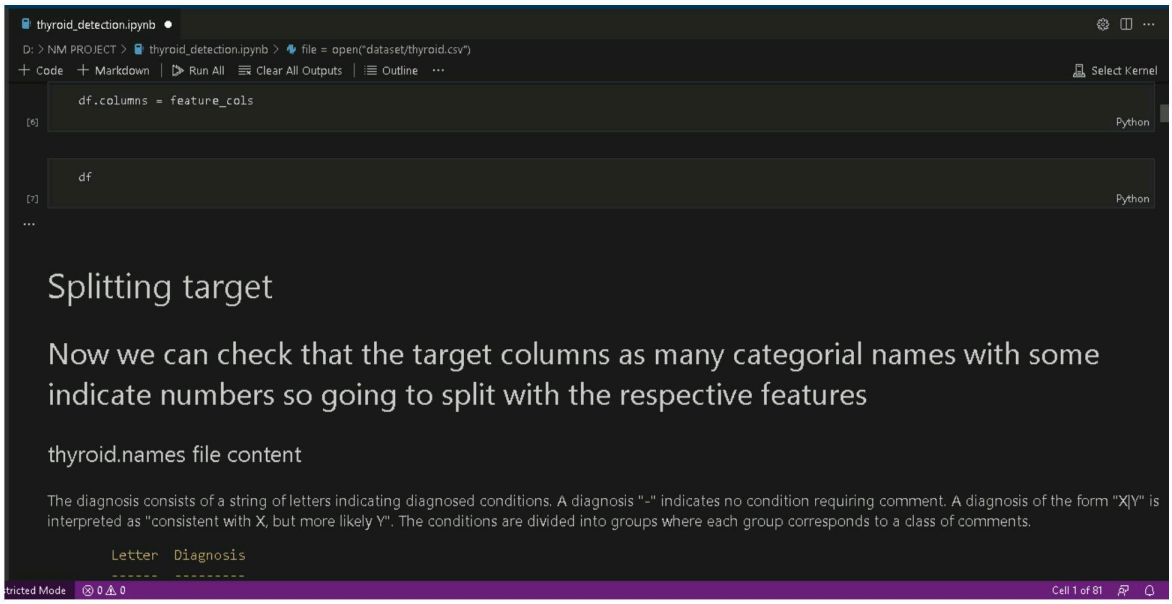
SUBMIT

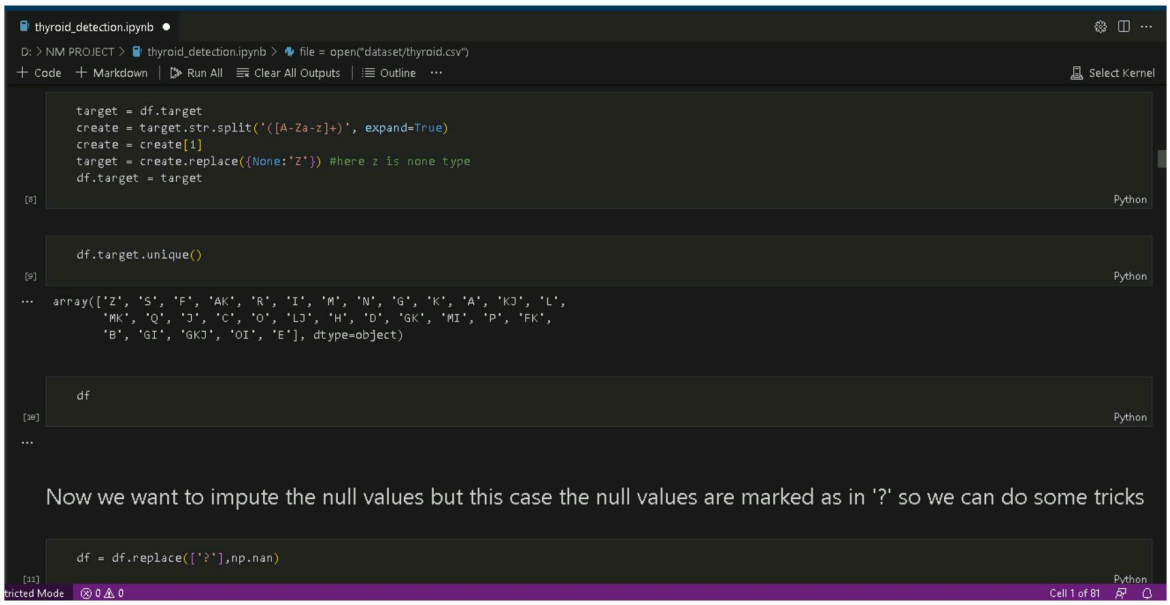


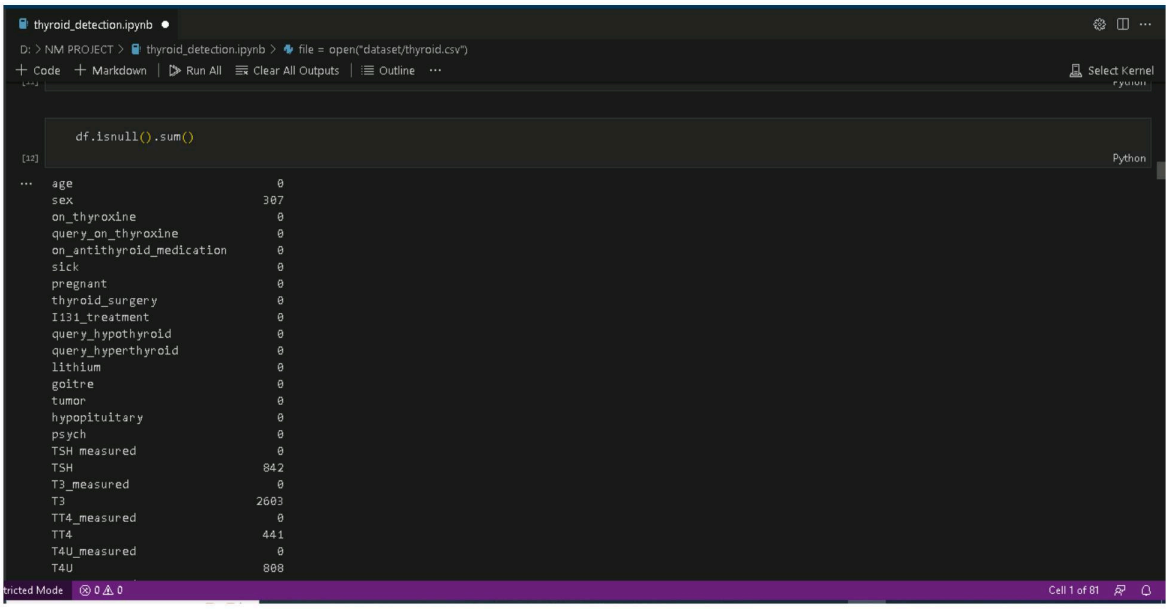
D. SAMPLE CODING

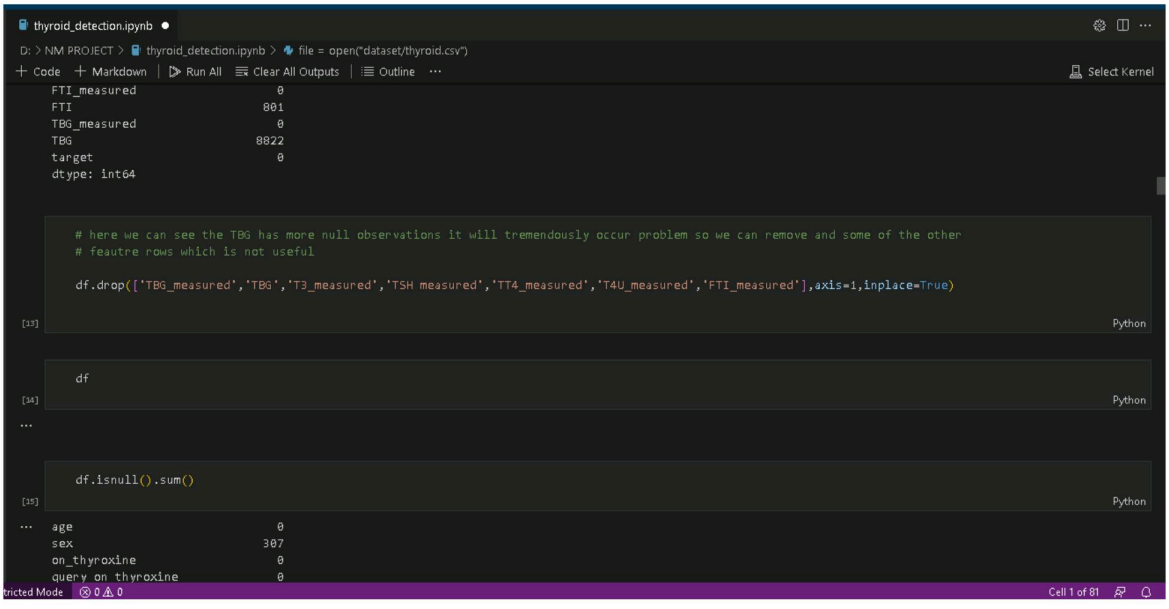


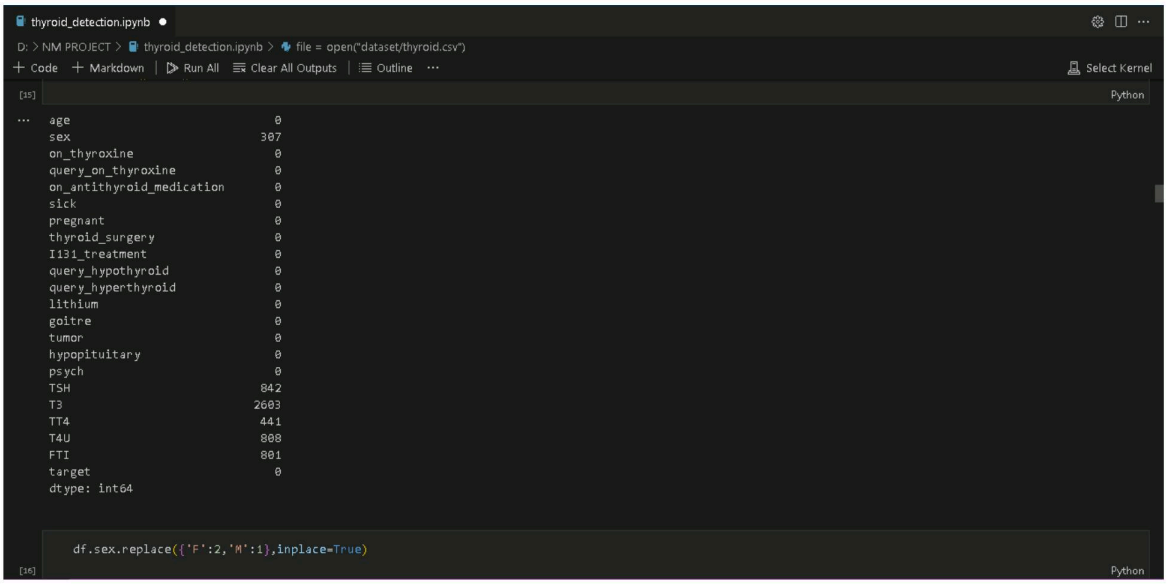


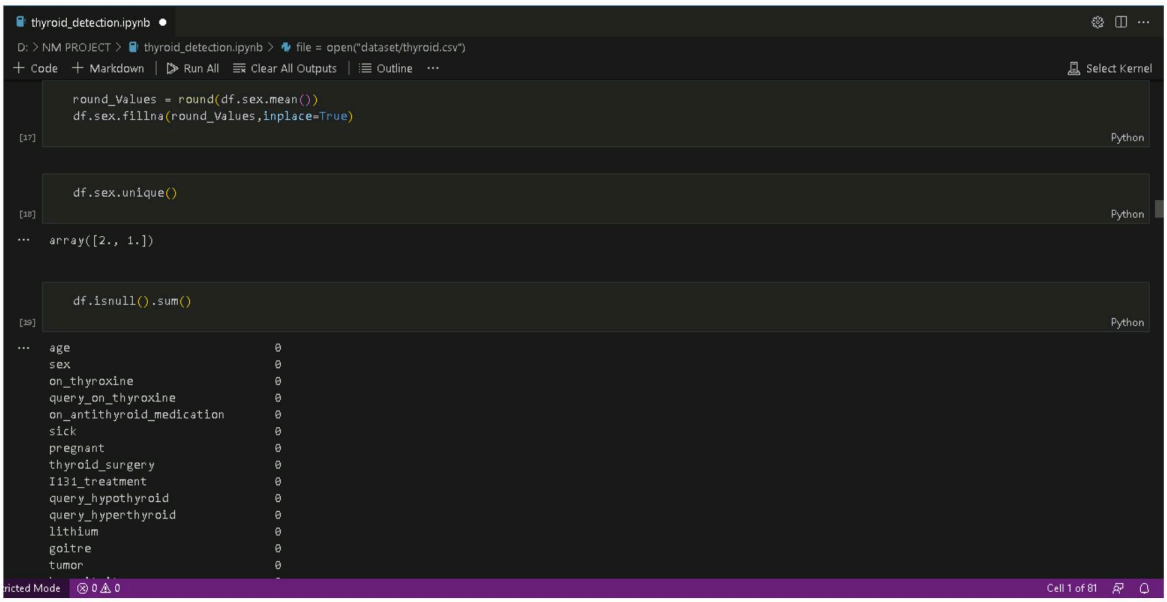

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Select Kernel
df.columns = feature_cols
[6]
Python
df
[7]
Python
...
Splitting target
Now we can check that the target columns as many categorial names with some indicate numbers so going to split with the respective features
thyroid.names file content
The diagnosis consists of a string of letters indicating diagnosed conditions. A diagnosis "-" indicates no condition requiring comment. A diagnosis of the form "X|Y" is interpreted as "consistent with X, but more likely Y". The conditions are divided into groups where each group corresponds to a class of comments.
Letter Diagnosis
tricted Mode
0 0
Cell 1 of 81

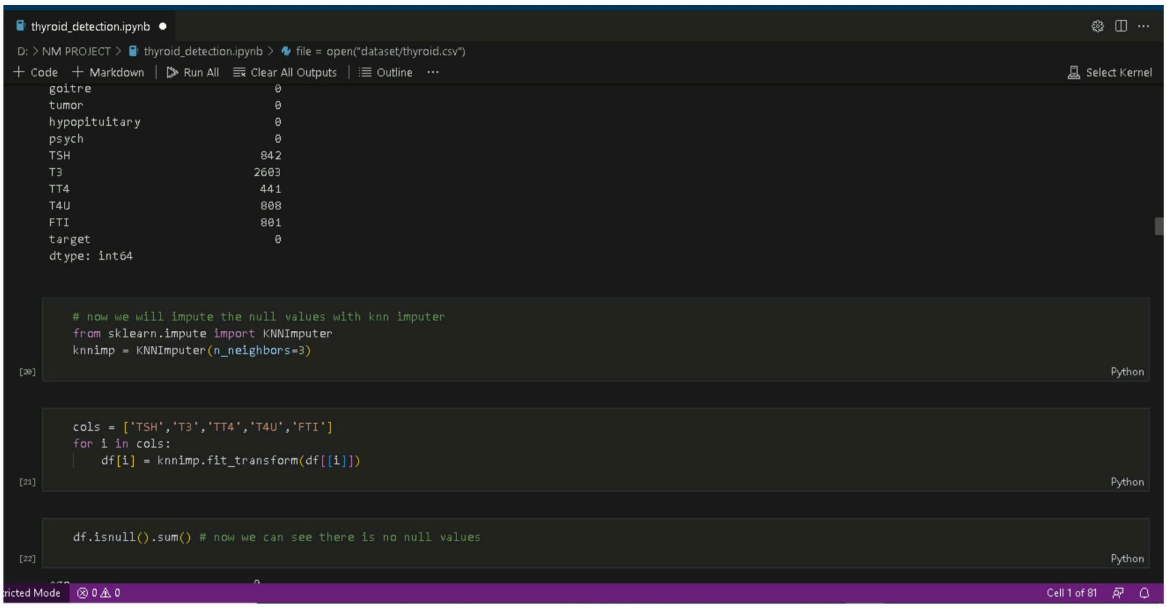

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Select Kernel
target = df.target
create = target.str.split('([A-Za-z]+)', expand=True)
create = create[1]
target = create.replace({None:'Z'}) #here z is none type
df.target = target
[8]
Python
df.target.unique()
[9]
Python
array(['Z', 'S', 'F', 'AK', 'R', 'I', 'M', 'N', 'G', 'K', 'A', 'KJ', 'L',
'MK', 'Q', 'J', 'C', 'O', 'LJ', 'H', 'D', 'GK', 'MI', 'P', 'FK',
'B', 'GI', 'GKJ', 'OI', 'E'], dtype=object)
df
[10]
Python
...
Now we want to impute the null values but this case the null values are marked as in '?' so we can do some tricks
df = df.replace(['?'],np.nan)
[11]
Python
tricted Mode
0 0
Cell 1 of 81

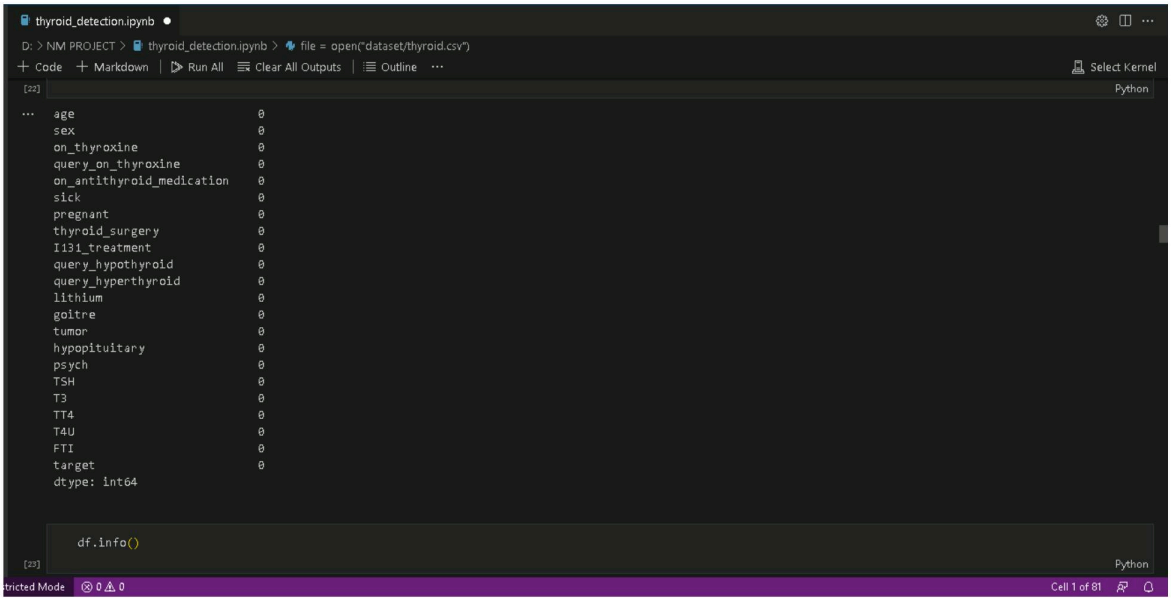

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
df.isnull().sum()
[12]
Python
age 0
sex 307
on_thyroxine 0
query_on_thyroxine 0
on_antithyroid_medication 0
sick 0
pregnant 0
thyroid_surgery 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 2603
TT4_measured 0
TT4 441
T4U_measured 0
T4U 808
tricted Mode
0 0
Cell 1 of 81

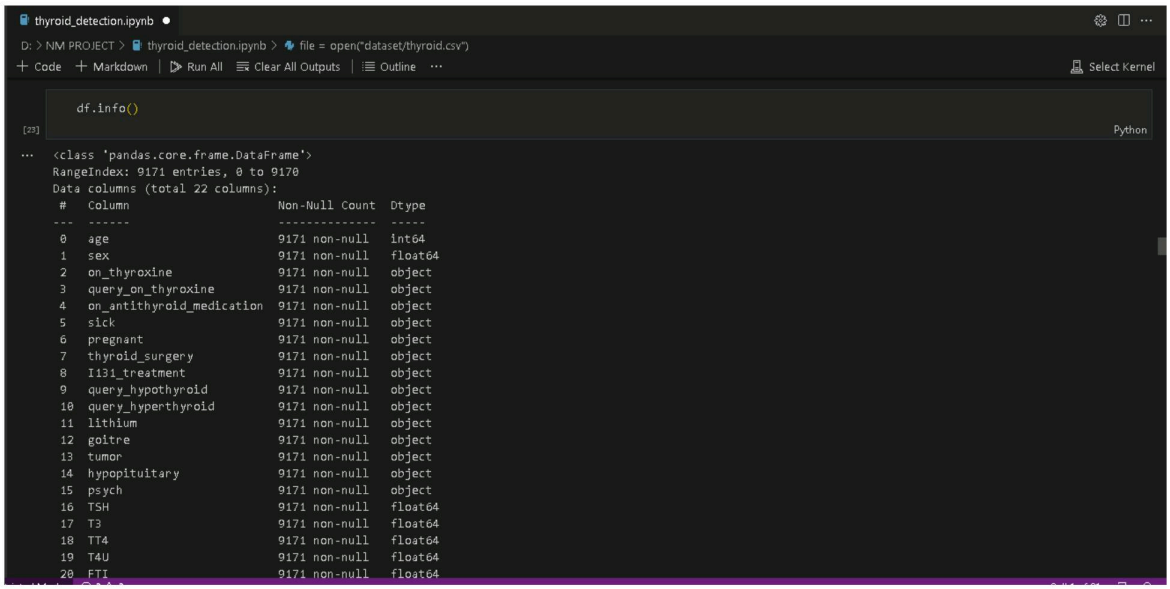

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
FTI_measured 0
FTI 801
TBG_measured 0
TBG 8822
target 0
dtype: int64
# here we can see the TBG has more null observations it will tremendously occur problem so we can remove and some of the other
# feautre rows which is not useful
df.drop(['TBG_measured','TBG','T3_measured','TSH measured','TT4_measured','T4U_measured','FTI_measured'],axis=1,inplace=True)
[13]
Python
df
[14]
Python
df.isnull().sum()
[15]
Python
age 0
sex 307
on_thyroxine 0
query_on_thyroxine 0
tricted Mode
0 0
Cell 1 of 81

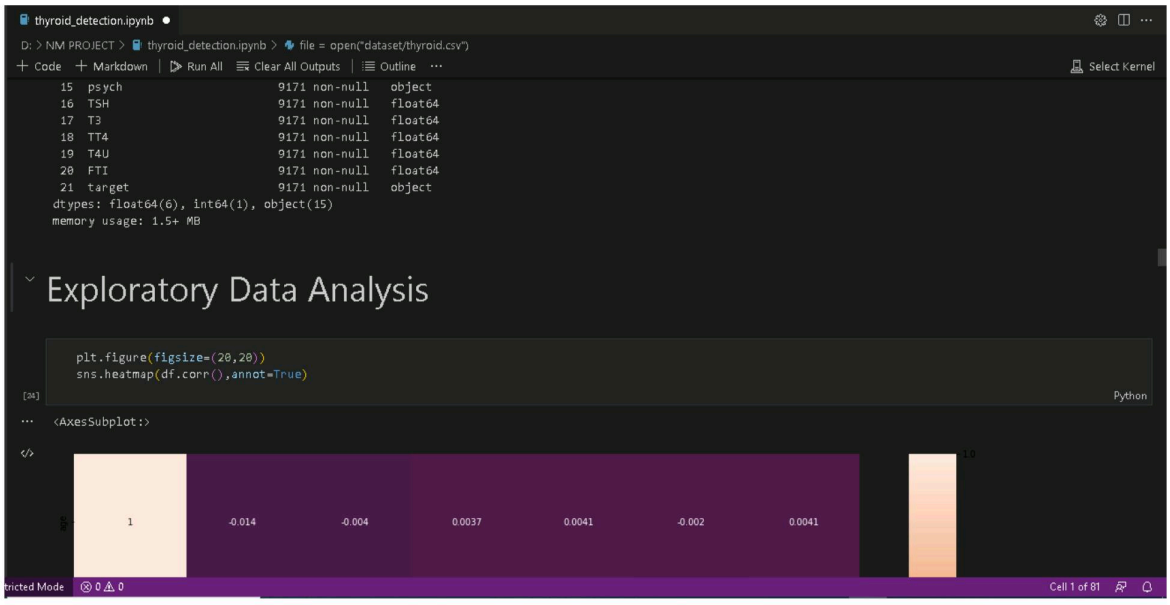

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code Markdown Run All Clear All Outputs Outline
Select Kernel
[15]
Python
age 0
sex 307
on_thyroxine 0
query_on_thyroxine 0
on_antithyroid_medication 0
sick 0
pregnant 0
thyroid_surgery 0
I131_treatment 0
query_hypothyroid 0
query_hyperthyroid 0
lithium 0
goitre 0
tumor 0
hypopituitary 0
psych 0
TSH 842
T3 2603
TT4 441
T4U 808
FTI 801
target 0
dtype: int64
df.sex.replace({'F':2,'M':1},inplace=True)
[16]
Python

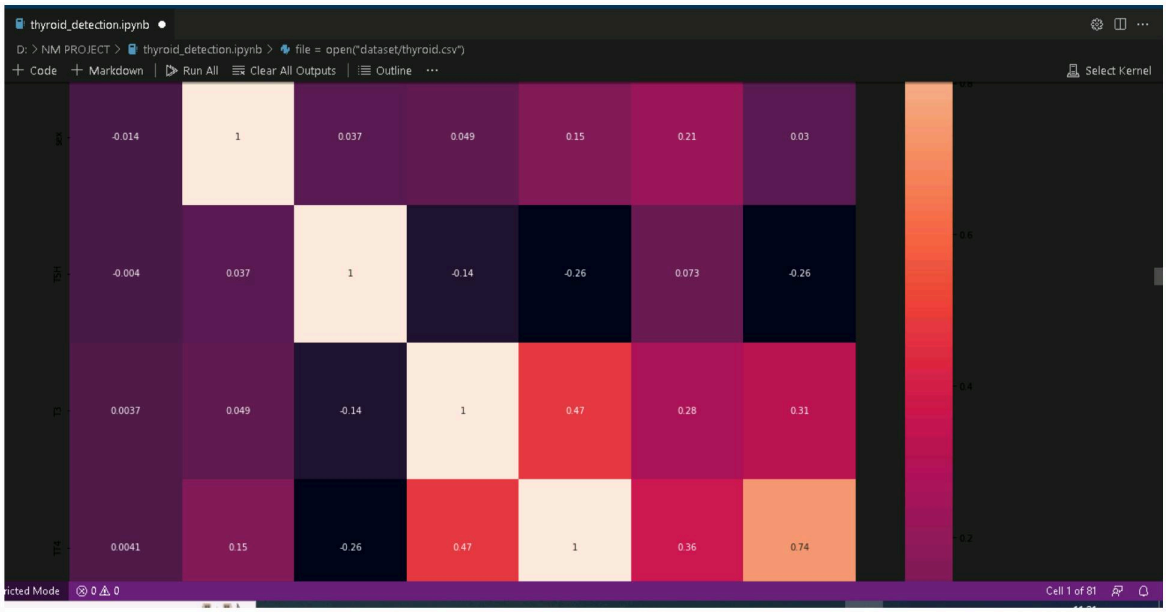

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code Markdown Run All Clear All Outputs Outline
Select Kernel
round_Values = round(df.sex.mean())
df.sex.fillna(round_Values,inplace=True)
[17]
Python
df.sex.unique()
[18]
Python
array([2., 1.])
df.isnull().sum()
[19]
Python
age 0
sex 0
on_thyroxine 0
query_on_thyroxine 0
on_antithyroid_medication 0
sick 0
pregnant 0
thyroid_surgery 0
I131_treatment 0
query_hypothyroid 0
query_hyperthyroid 0
lithium 0
goitre 0
tumor 0
Cell 1 of 81

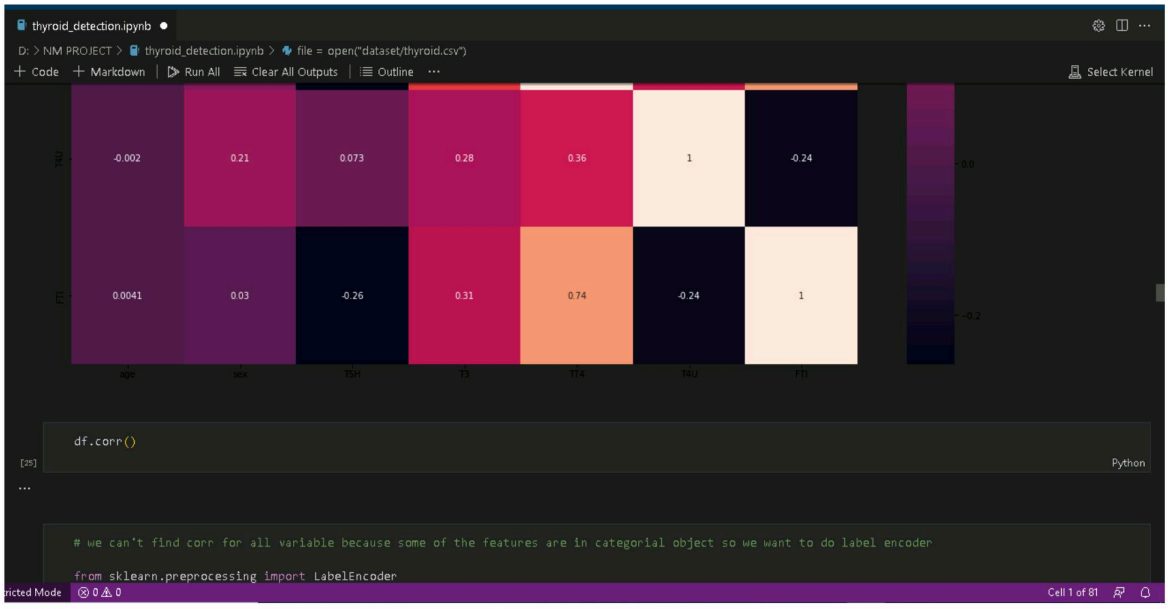

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code Markdown Run All Clear All Outputs Outline
Select Kernel
goitre 0
tumor 0
hypopituitary 0
psych 0
TSH 842
T3 2603
TT4 441
T4U 808
FTI 801
target 0
dtype: int64
# now we will impute the null values with knn imputer
from sklearn.impute import KNNImputer
knnimp = KNNImputer(n_neighbors=3)
[20]
Python
cols = ['TSH','T3','TT4','T4U','FTI']
for i in cols:
df[i] = knnimp.fit_transform(df[[i]])
[21]
Python
df.isnull().sum() # now we can see there is no null values
[22]
Python
Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code Markdown Run All Clear All Outputs Outline
Select Kernel
[22]
Python
age 0
sex 0
on_thyroxine 0
query_on_thyroxine 0
on_antithyroid_medication 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 0
T3 0
TT4 0
T4U 0
FTI 0
target 0
dtype: int64
df.info()
[23]
Python
Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
df.info()
[23]
Python
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9171 entries, 0 to 9170
Data columns (total 22 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 age 9171 non-null int64
1 sex 9171 non-null float64
2 on_thyroxine 9171 non-null object
3 query_on_thyroxine 9171 non-null object
4 on_antithyroid_medication 9171 non-null object
5 sick 9171 non-null object
6 pregnant 9171 non-null object
7 thyroid_surgery 9171 non-null object
8 I131_treatment 9171 non-null object
9 query_hypothyroid 9171 non-null object
10 query_hyperthyroid 9171 non-null object
11 lithium 9171 non-null object
12 goitre 9171 non-null object
13 tumor 9171 non-null object
14 hypopituitary 9171 non-null object
15 psych 9171 non-null object
16 TSH 9171 non-null float64
17 T3 9171 non-null float64
18 TT4 9171 non-null float64
19 T4U 9171 non-null float64
20 FTI 9171 non-null float64


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
15 psych 9171 non-null object
16 TSH 9171 non-null float64
17 T3 9171 non-null float64
18 TT4 9171 non-null float64
19 T4U 9171 non-null float64
20 FTI 9171 non-null float64
21 target 9171 non-null object
dtypes: float64(6), int64(1), object(15)
memory usage: 1.5+ MB
Exploratory Data Analysis
plt.figure(figsize=(20,20))
sns.heatmap(df.corr(),annot=True)
[24]
Python
<AxesSubplot:>
1
-0.014
-0.004
0.0037
0.0041
-0.002
0.0041
1.0
tricted Mode
0 0
Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
-0.014
1
0.037
0.049
0.15
0.21
0.03
-0.004
0.037
1
-0.14
-0.26
0.073
-0.26
0.0037
0.049
-0.14
1
0.47
0.28
0.31
0.0041
0.15
-0.26
0.47
1
0.36
0.74
Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
-0.002
0.21
0.073
0.28
0.36
1
-0.24
0.0041
0.03
-0.26
0.31
0.74
-0.24
1
age
sex
TSH
T3
TT4
T4U
FTI
df.corr()
[25]
Python
# we can't find corr for all variable because some of the features are in categorial object so we want to do label encoder
from sklearn.preprocessing import LabelEncoder
Cell 1 of 81

thyroid_detection.ipynb ●

D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")

+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

```
# we can't find corr for all variable because some of the features are in categorial object so we want to do label encoder

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

[26] Python

```
cols = df.select_dtypes(include=['object'])
```

[27] Python

```
for i in cols.columns:
    try:
        df[i] = le.fit_transform(df[i])
    except:
        continue
```

[28] Python

```
# now we can see their is correlation in some features
for a in range(len(df.corr())):
    for b in range(a):
        if((df.corr().iloc[a,b]) >= 0.7):
            print(df.corr().columns[b])
```

[29] Python

··· TT4

ricted Mode ⊗ 0 ⚠ 0 Cell 1 of 81

thyroid_detection.ipynb ●

D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")

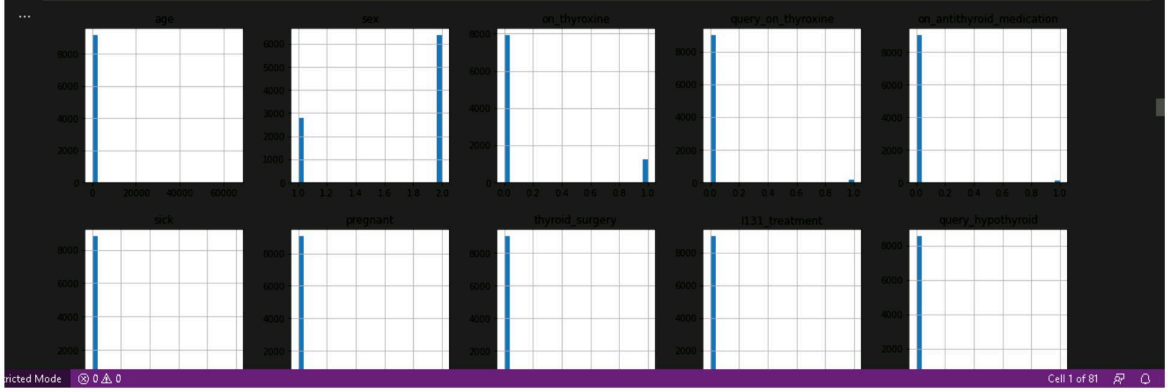
+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

```
df.drop('TT4',axis=1,inplace=True)
```

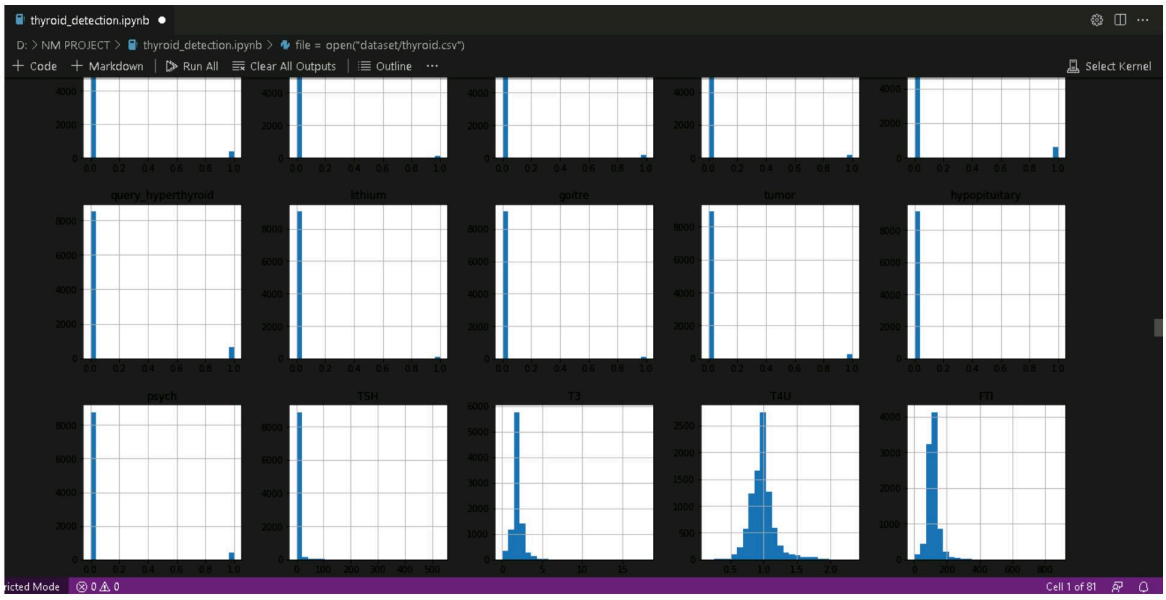
[30] Python

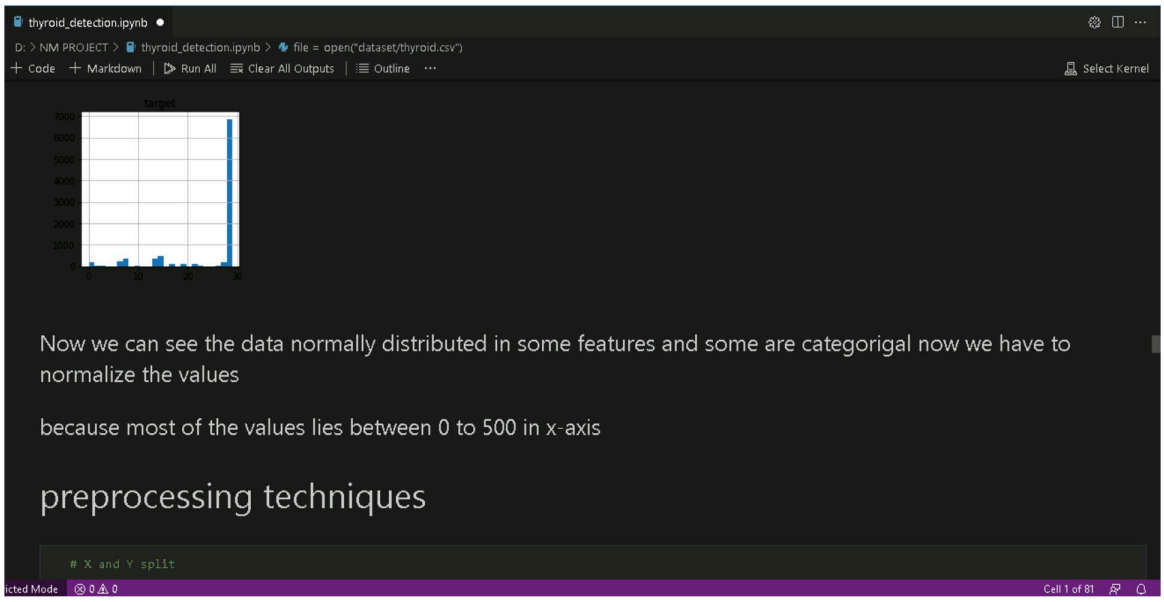
```
df.hist(bins=25,figsize=(20,20));
```

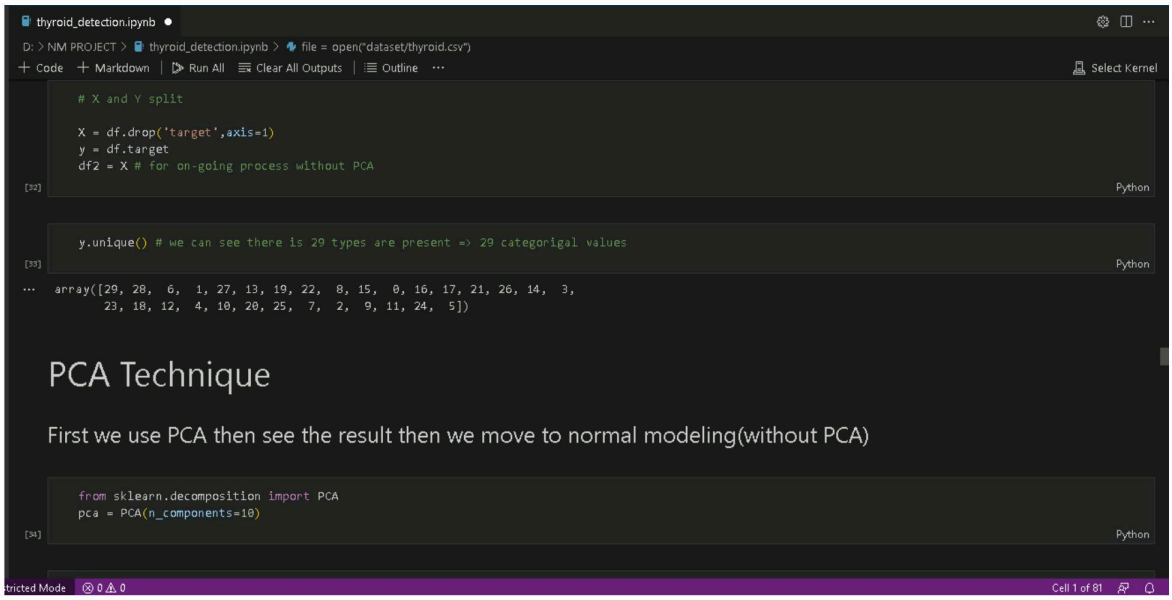
[31] Python

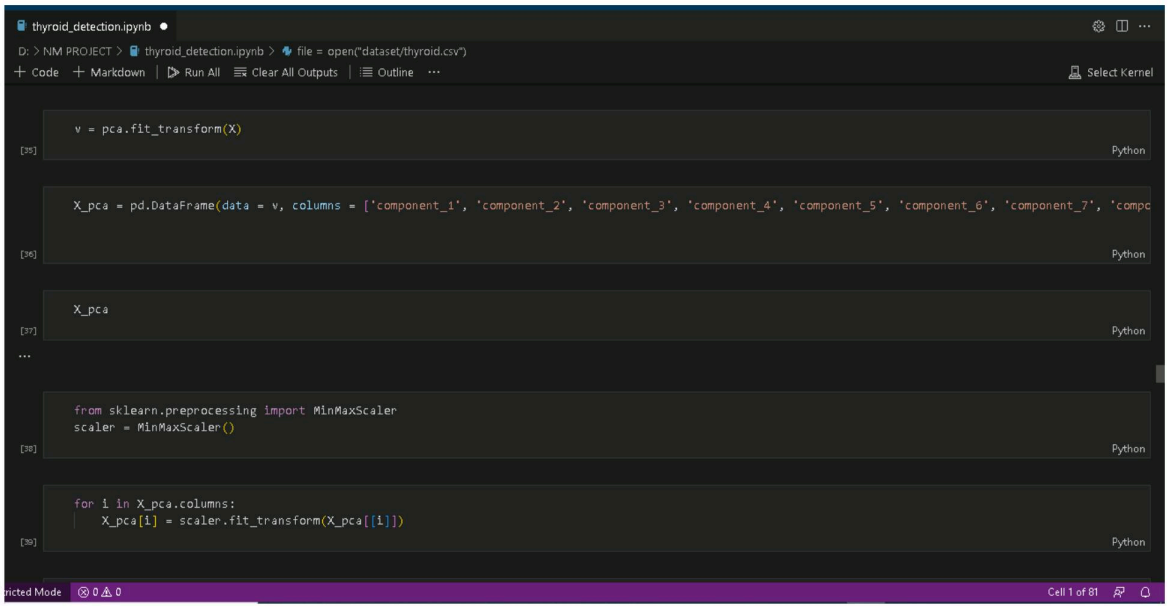


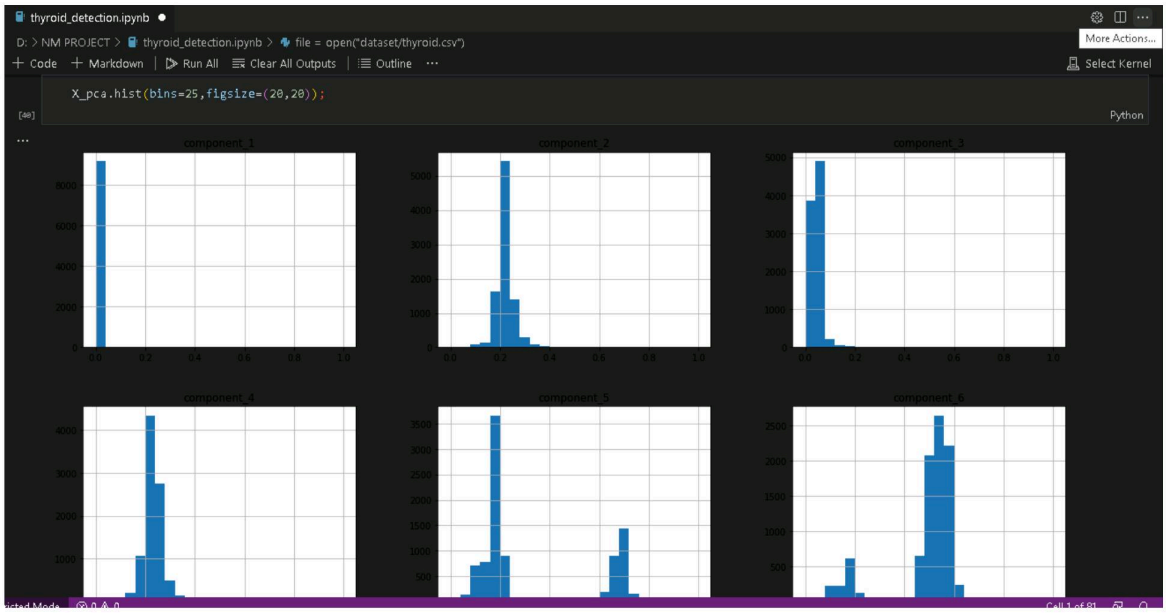
ricted Mode ⊗ 0 ⚠ 0 Cell 1 of 81

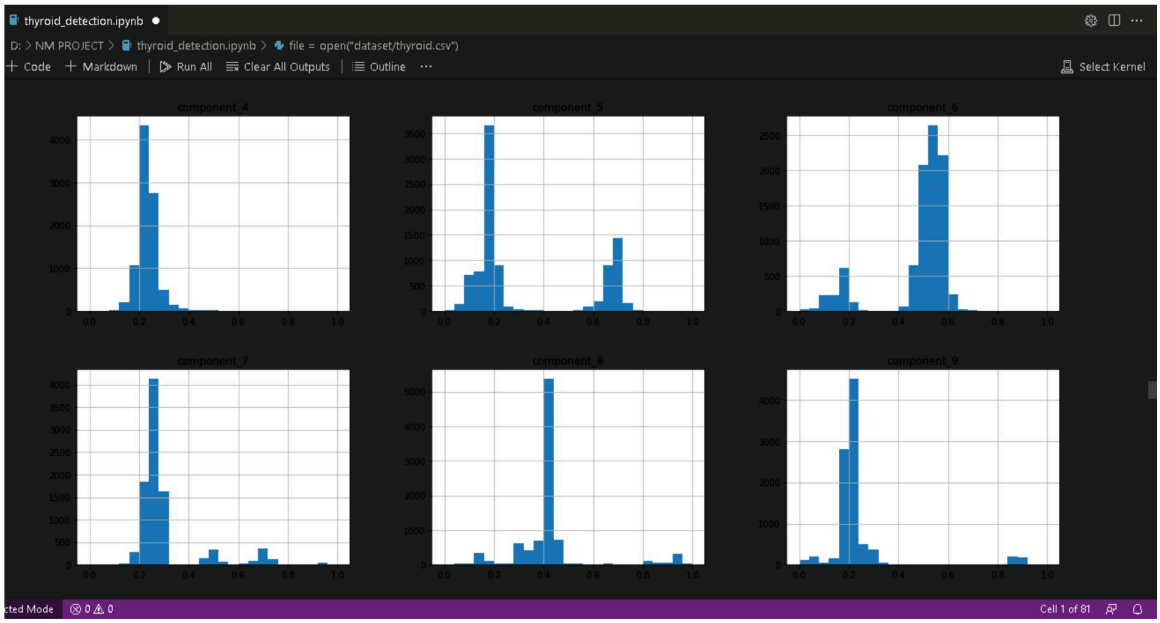

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
query_hyperthyroid
lithium
goitre
tumor
hypopituitary
psych
TSH
T3
T4U
FTI
Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
target
Now we can see the data normally distributed in some features and some are categorigal now we have to normalize the values
because most of the values lies between 0 to 500 in x-axis
preprocessing techniques
X and Y split
Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
# X and Y split
X = df.drop('target',axis=1)
y = df.target
df2 = X # for on-going process without PCA
[32]
Python
y.unique() # we can see there is 29 types are present => 29 categorigal values
[33]
Python
array([29, 28, 6, 1, 27, 13, 19, 22, 8, 15, 0, 16, 17, 21, 26, 14, 3,
23, 18, 12, 4, 10, 20, 25, 7, 2, 9, 11, 24, 5])
PCA Technique
First we use PCA then see the result then we move to normal modeling(without PCA)
from sklearn.decomposition import PCA
pca = PCA(n_components=10)
[34]
Python
Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
v = pca.fit_transform(X)
[35]
Python
X_pca = pd.DataFrame(data = v, columns = ['component_1', 'component_2', 'component_3', 'component_4', 'component_5', 'component_6', 'component_7', 'compc
[36]
Python
X_pca
[37]
Python
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
[38]
Python
for i in X_pca.columns:
X_pca[i] = scaler.fit_transform(X_pca[[i]])
[39]
Python
Cell 1 of 81

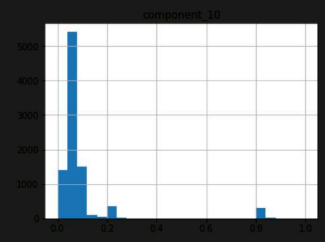

thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
More Actions...
X_pca.hist(bins=25,figsize=(20,20));
Python
component_1
component_2
component_3
component_4
component_5
component_6


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
component_4
component_5
component_6
component_7
component_8
component_9
Cell 1 of 81

thyroid_detection.ipynb ●

D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")

+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel




```
X_pca
```

[41] Python

···

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X_pca,y,test_size=0.33,random_state=42)
```

[42] Python

Model Selection

ricted Mode ⊗ 0 ⚠ 0 Cell 1 of 81

thyroid_detection.ipynb ●

D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")

+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

Model Selection

```
from sklearn.metrics import accuracy_score
```

[61] Python

Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(max_depth=3)
clf = tree.fit(X_train,y_train)
treepredict = clf.predict(X_test)
```

[44] Python

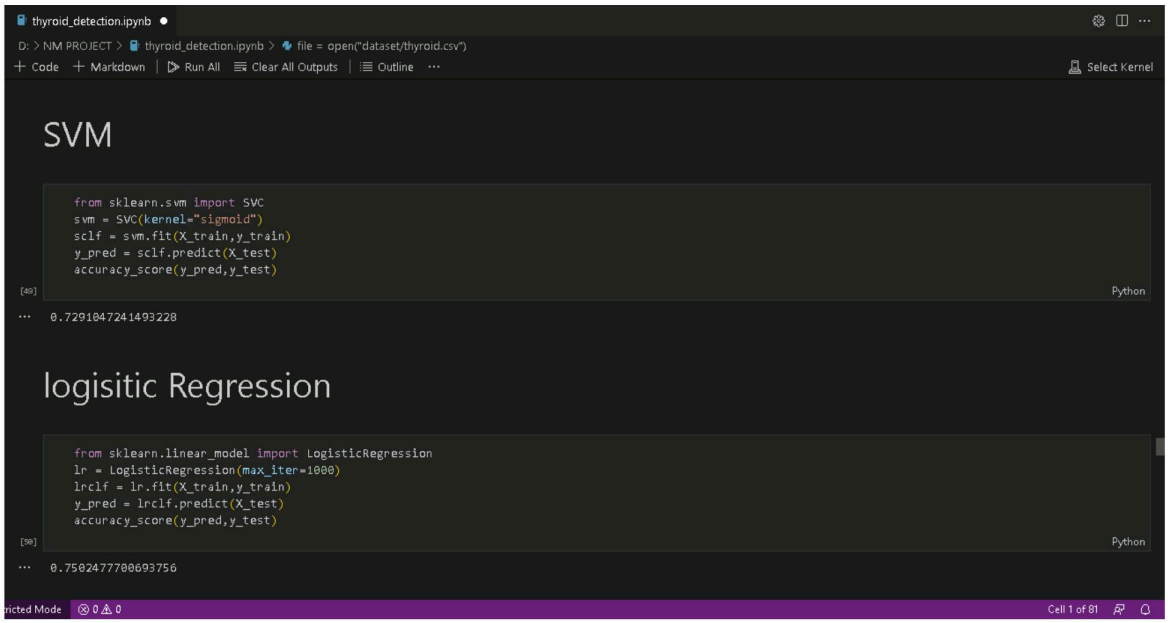
```
accuracy_score(treepredict,y_test)
```

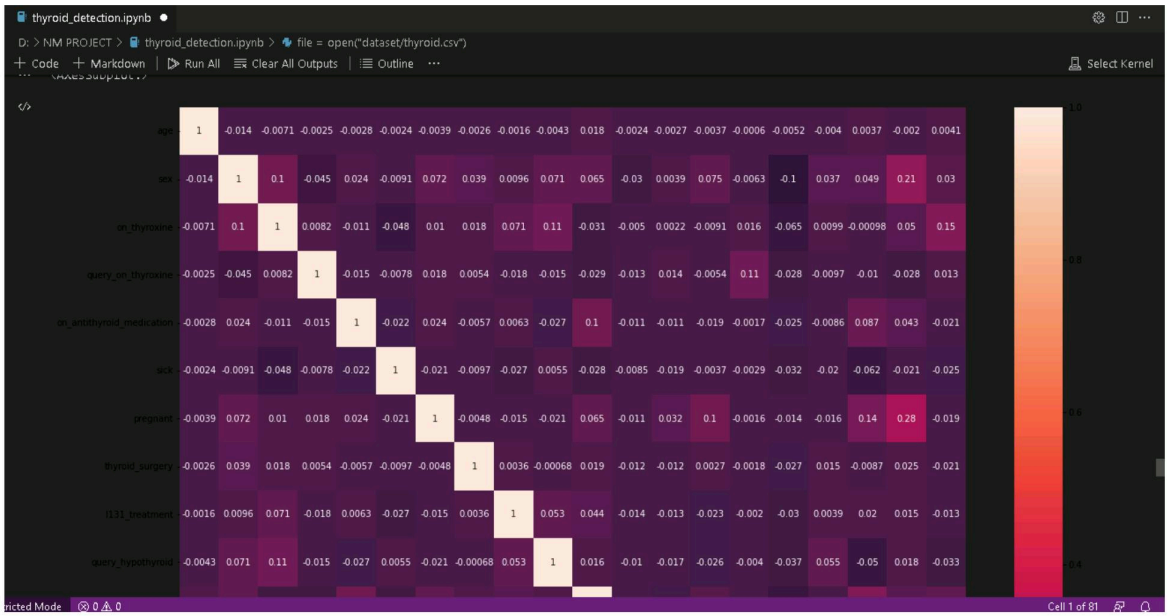
[45] Python

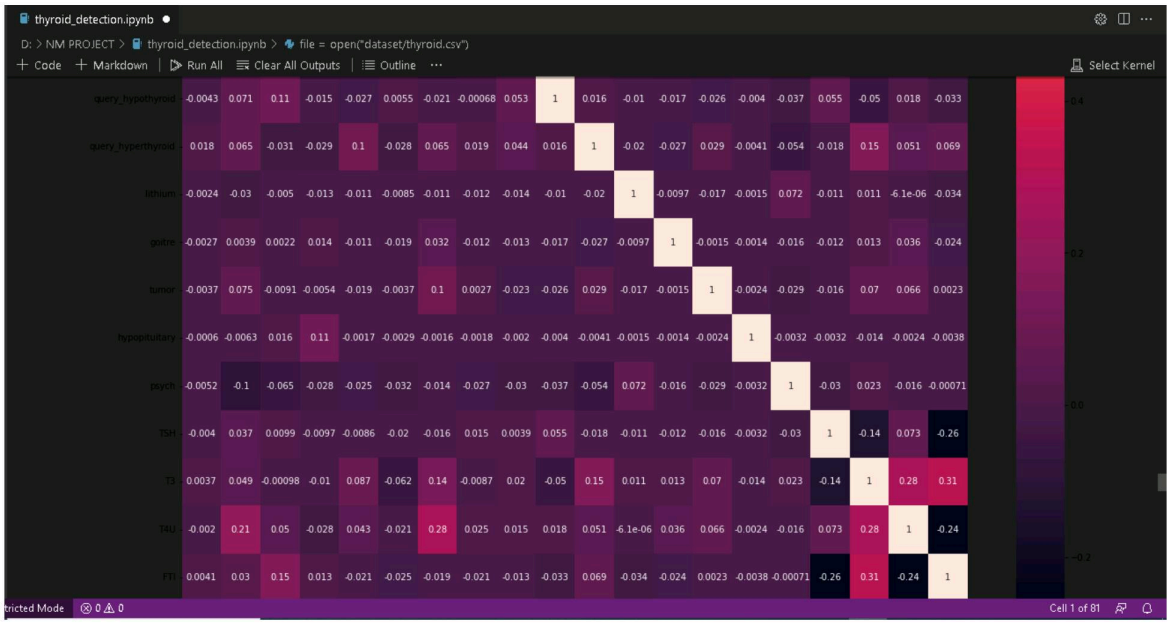
··· 0.7961678229269904

Random Forest Classifier

ricted Mode ⊗ 0 ⚠ 0 Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
SVM
from sklearn.svm import SVC
svm = SVC(kernel="sigmoid")
sclf = svm.fit(X_train,y_train)
y_pred = sclf.predict(X_test)
accuracy_score(y_pred,y_test)
[49]
Python
0.7291047241493228
logisitic Regression
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter=1000)
lrclf = lr.fit(X_train,y_train)
y_pred = lrclf.predict(X_test)
accuracy_score(y_pred,y_test)
[50]
Python
0.7502477700693756
Cell 1 of 81


thyroid_detection.ipynb
D: > NM PROJECT > thyroid_detection.ipynb > file = open("dataset/thyroid.csv")
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
age 1 -0.014 -0.0071 -0.0025 -0.0028 -0.0024 -0.0039 -0.0026 -0.0016 -0.0043 0.018 -0.0024 -0.0027 -0.0037 -0.0006 -0.0052 -0.004 0.0037 -0.002 0.0041
sex -0.014 1 0.1 -0.045 0.024 -0.0091 0.072 0.039 0.0096 0.071 0.065 -0.03 0.0039 0.075 -0.0063 -0.1 0.037 0.049 0.21 0.03
on_thyroxine -0.0071 0.1 1 0.0082 -0.011 -0.048 0.01 0.018 0.071 0.11 -0.031 -0.005 0.0022 -0.0091 0.016 -0.065 0.0099 -0.00098 0.05 0.15
query_on_thyroxine -0.0025 -0.045 0.0082 1 -0.015 -0.0078 0.018 0.0054 -0.018 -0.015 -0.029 -0.013 0.014 -0.0054 0.11 -0.028 -0.0097 -0.01 -0.028 0.013
on_antithyroid_medication -0.0028 0.024 -0.011 -0.015 1 -0.022 0.024 -0.0057 0.0063 -0.027 0.1 -0.011 -0.011 -0.019 -0.0017 -0.025 -0.0086 0.087 0.043 -0.021
sick -0.0024 -0.0091 -0.048 -0.0078 -0.022 1 -0.021 -0.0097 -0.027 0.0055 -0.028 -0.0085 -0.019 -0.0037 -0.0029 -0.032 -0.02 -0.062 -0.021 -0.025
pregnant -0.0039 0.072 0.01 0.018 0.024 -0.021 1 -0.0048 -0.015 -0.021 0.065 -0.011 0.032 0.1 -0.0016 -0.014 -0.016 0.14 0.28 -0.019
thyroid_surgery -0.0026 0.039 0.018 0.0054 -0.0057 -0.0097 -0.0048 1 0.0036 -0.00068 0.019 -0.012 -0.012 0.0027 -0.0018 -0.027 0.015 -0.0087 0.025 -0.021
I131_treatment -0.0016 0.0096 0.071 -0.018 0.0063 -0.027 -0.015 0.0036 1 0.053 0.044 -0.014 -0.013 -0.023 -0.002 -0.03 0.0039 0.02 0.015 -0.013
query_hypothyroid -0.0043 0.071 0.11 -0.015 -0.027 0.0055 -0.021 -0.00068 0.053 1 0.016 -0.01 -0.017 -0.026 -0.004 -0.037 0.055 -0.05 0.018 -0.033
Cell 1 of 81




```
import pandas as
pd
import numpy as
np
import
matplotlib.pyplot
as plt
```