###### THYROID DISEASE CLASSIFICATION USING ML

1. INTRODUCTION
   1. OVERVIEW

* Thyroid disease is a general term for a medical condition that **keeps your thyroid from making the right amount of hormones**.
* Thyroid gland diseases are among the most prevalent endocrine disorders in the world, second only to diabetes, according to the World Health Organization. Hyper function hyperthyroidism and hypothyroidism affect about 2% and 1% of individuals, respectively.
* Men have about a tenth of the prevalence of women. Hyper-and hypothyroidism may be caused by thyroid gland dysfunction, secondary to pituitary gland failure, or tertiary to hypothalamic malfunction.
* Due to dietary iodine deficiency, goiter or active thyroid nodules may become prevalent in some regions, with a prevalence of up to 15%. The thyroid gland can also be the location of different kinds of tumors and can be a dangerous place where endogenous antibodies wreak havoc (autoantibodies).
* Early disease detection, diagnosis, and care, according to doctors, are vital in preventing disease progression and even death. For several different forms of anomalies, earlyidentification and differential diagnosis raises the odds of good treatment. Despite multiple trials, clinical diagnosis is often thought to be a difficult task.
* The thyroid is a butterfly-shaped gland in the front of the neck.
* Its primary function is to produce the hormones triiodothyronine (T3) and thyroxine (T4), which help regulate metabolism and aid in brain development, digestive function, muscle control, and mood regulation.
* When the thyroid produces either too much or too little of these hormones, it causes the gland to work inefficiently, leading to disorders such as hyperthyroidism and hypothyroidism, the two most common types of thyroid disease.
* Other thyroid-related conditions include thyroiditis, thyroid nodules, goiter, and thyroid cancer.
* Thyroid disorders can affect heart rate, mood, energy level, metabolism, bone health, pregnancy and many other functions.
  1. purpose

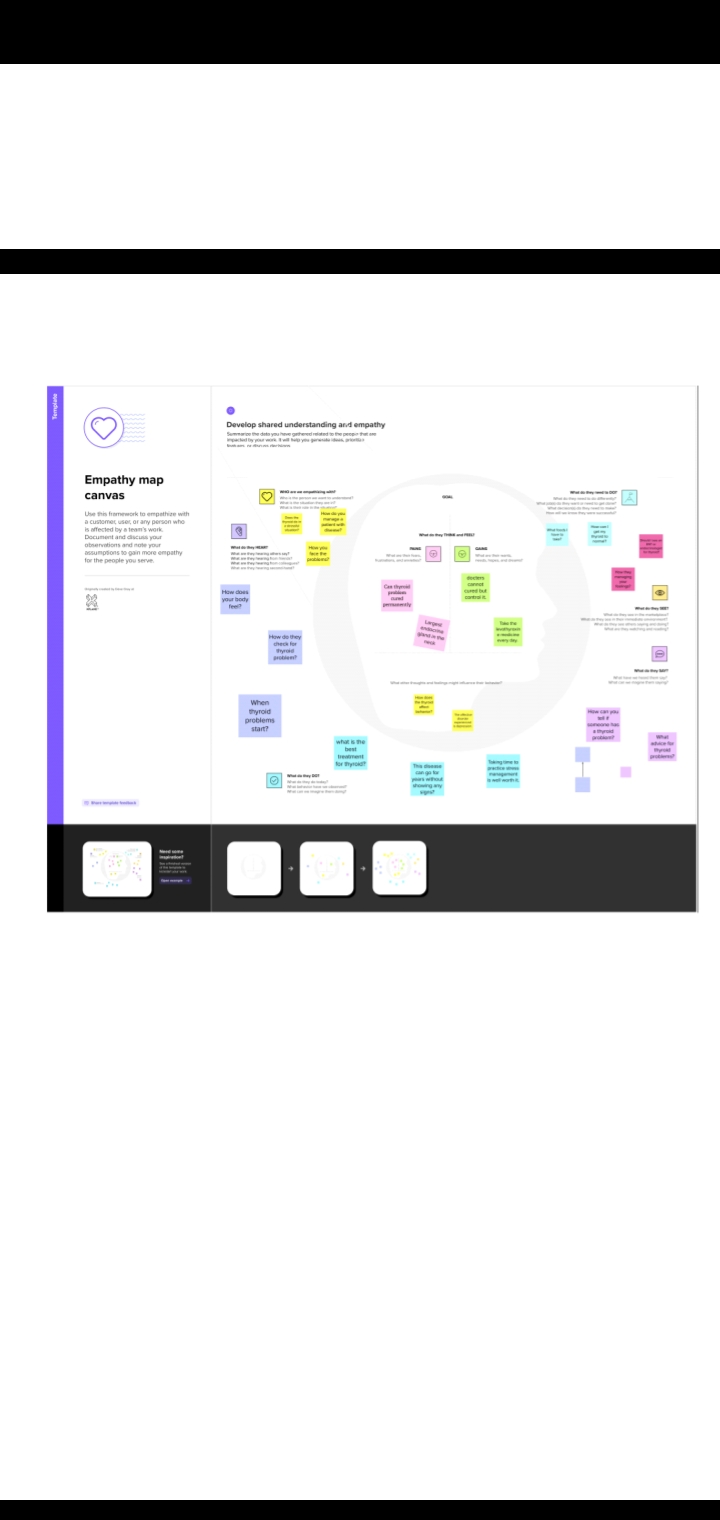
Thyroid Detection using Machine Learning is a project idea that aims a smart and precise way to predict thyroid disease. We have made use of logistic regression algorithm to train our dataset and to predict thyroid disease with more accuracy. Here the machine is trained to detect whether the person normal, hyperhypothyroidism based on the user’s input. So when user enters data in web app the data will be processed in backend (model) and the result will be displayed on the screen. Our objective was to give society an efficient and precise way of machine learning which can be used in applications aiming to perform disease detection. Further development can be do by using image processing of ultrasonic scanning of thyroid images to predict thyroid nodules and cancer, which cannot be recognized in blood test report. By combining both the results, thyroid disease prediction can cover all thyroid related disease.

**Problem Definition & Design Thinking**

**2.1 Empathy Map**

**1. In the ideation phase we have empathized as our client in our project "Thyroid disease classification using ML", and acquired the following Details and we represent them in the following empathy map.**

**2. Which are represented in the empathy map given below.**

****

**Brainstorm & idea prioritize template:**

1. Under this activity our team members have gathered and discussed various ideas to solve our project problems, each member contributed 6 to 10 ideas.
2. After gathering all ideas we have assessed the impact and feasibility of each point.
3. Finally we have assigned the priority, for each point based on these impact values.

****

**ADVANTAGES OF THYROID DISEASE**

* Surgery is an excellent, and seemingly superior treatment for Graves' disease, resulting in lower all-cause death rate, less medical and psychological complications, minimal relapse risk, and lower overall health care costs long-term.
* The only downside is the expected permanent hypothyroidism.

**DISADVANTAGES OF THYROID DISEASE**

* If your entire thyroid is removed, your body can no longer make thyroid hormone. Without medication, you'll develop symptoms of underactive thyroid (hypothyroidism).
* These symptoms may include dry skin, fatigue and weight gain.

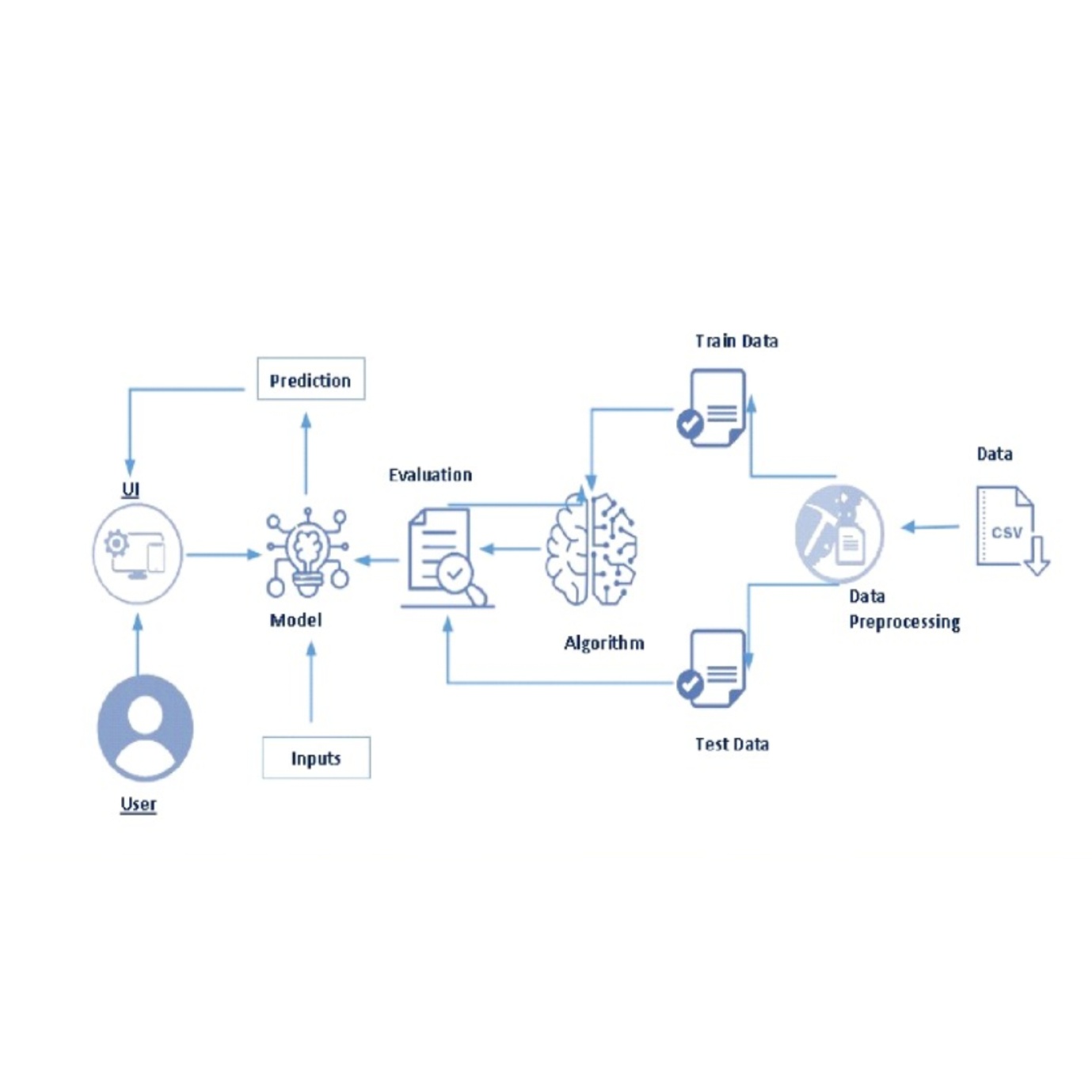
**CONCLUSION**

Thyroid disease is one of the diseases that afflict the world’s population, and the number of cases of this disease is increasing. Because of medical reports that show serious imbalances in thyroid diseases, our study deals with the classification of thyroid disease between hyperthyroidism and hypothyroidism.

**FUTURE SCOPE**

Extensive experiments show that the extra tree classifier-based selected feature yields the best results with 0.99 accuracy and an F1 score when used with the random forest classifier. Results suggest that the machine learning models are a better choice for thyroid disease detection regarding the provided accuracy and the computational complexity. K-fold cross-validation and performance comparison with existing studies corroborate the superior performance of the proposed approach.

**TECHNICAL ARCHITECTURE:**

****

**Project Flow:**

● The user interacts with the UI to enter the input.

● Entered input is analysed by the model which is integrated.

● Once the model analyses the input the prediction is showcased on the UI

**To accomplish this, we have to complete all the activities listed below**

**● Define Problem / Problem Understanding**

○ Specify the business problem

○ Business requirements

○ Literature Survey

○ Social or Business Impact.

● **Data Collection & Preparation**

○ Collect the dataset

○ Data Preparation

**● Exploratory Data Analysis**

○ Descriptive statistical

○ Visual Analysis

**● Model Building**

○ Training the model in multiple algorithms

○ Testing the model

**● Performance Testing & Hyperparameter Tuning**

○ Testing model with multiple evaluation metrics

○ Comparing model accuracy before & after applying hyperparameter tuning

**● Model Deployment**

○ Save the best model

○ Integrate with Web Framework

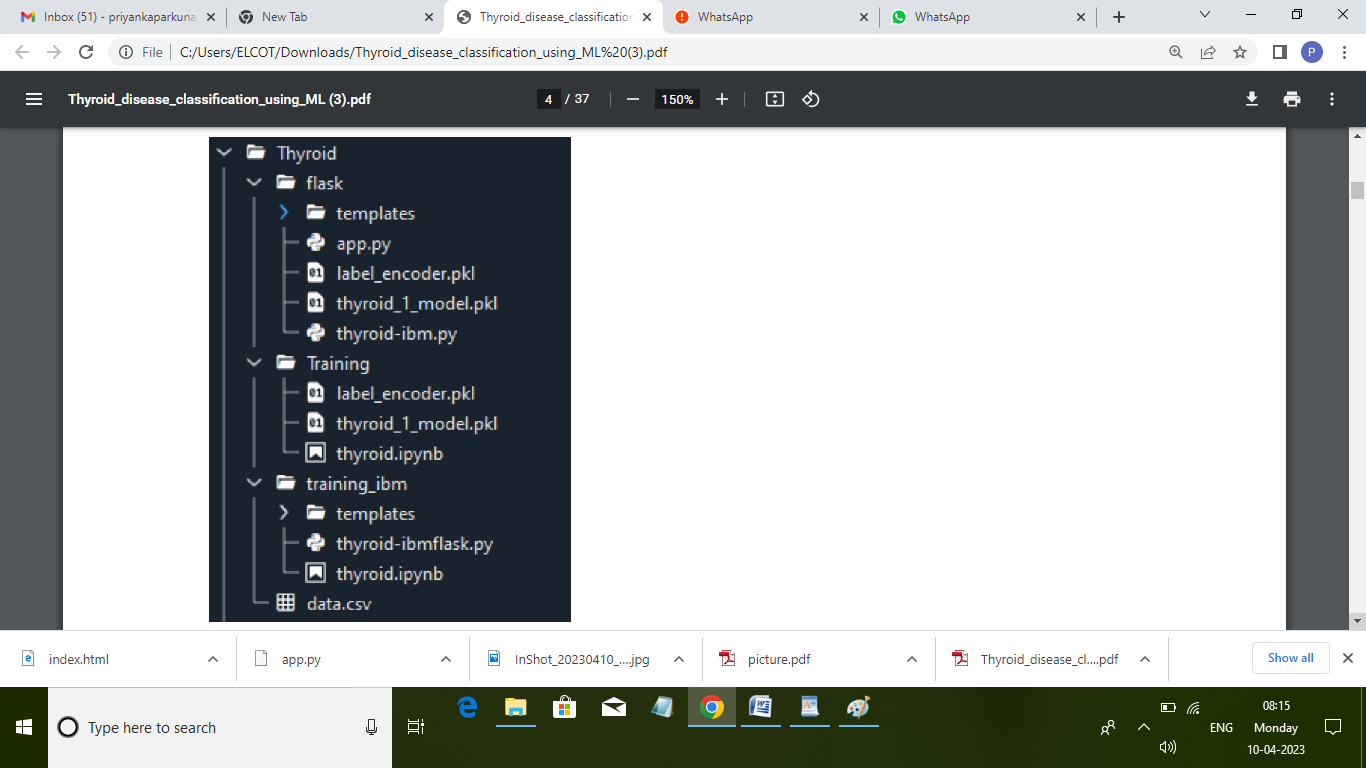
**● Project Demonstration & Documentation**

○ Record explanation Video for project end to end solution

○ Project Documentation-Step by step project development procedure

**Project Structure:**

**Create the Project folder which contains files as shown below**

****

**Milestone 1: Define Problem / Problem Understanding**

**Activity 1: Specify the business problem**

Refer to Project Description

**Activity 2: Business requirements**

* The business requirements for a machine learning model to predict thyroid disease include the ability to accurately predict thyroid disease based on the scan results, Minimise the number of false positives (wrong thyroid disease confirmations) and false negatives (thyroid is there but got as not thyroid disease). Provide an explanation for the model's decision, to comply with regulations and improve transparency.

**Activity 3: Literature Survey**

* The thyroid gland is one of the body’s most visible endocrine glands. Its size is determined by the individual’s age, gender, and physiological states, such as pregnancy or lactation. It is divided into two lobes (right and left) by an isthmus (a band of tissue). It is imperceptible in everyday life yet can be detected when swallowing. The thyroid hormones T4 and T3 are needed for normal thyroid function. These hormones have a direct effect on the body’s metabolic rate. It contributes to the stimulation of glucose, fatty acid, and other molecule consumption. Additionally, it enhances oxygen consumption in the majority of the body’s cells by assisting in the processing of uncoupling proteins, which contributes to an improvement in the rate of cellular respiration. Thyroid conditions are difficult to detect in test results, and only trained professionals can do so. However, reading such extensive reports and predicting future results is difficult. Assume a machine learning model can detect the thyroid disease in a patient. The thyroid disease can then be easily identified based on the symptoms in the patient’s history. Currently, models are evaluated using accuracy metrics on a validation dataset that is accessible.

**Activity 4: Social or Business Impact.**

* Social Impact:- Untreated/undetected thyroid disease is more dangerous at times it can lead to fatal of the person. So, we can detect it at the earliest then people canget treatment and get cured. Business Model/Impact:- We can make this application public, offer services as a subscription based or can collaborate with healthcare centres or specialists.

**Milestone 2: Data collection & preparation**

**Activity 1: Download the dataset:**

**Link:** [**https://www.kaggle.com/prathamtripathi/drug-classification**](https://www.kaggle.com/prathamtripathi/drug-classification)

**Activity 1.1: Importing the libraries:**

**Import the necessary libraries as shown below image.**

#pandas

import pandas as pd

#numpy

import numpy as np

#matplotlib

import matplotlib.pyplot as plt

#seaborn

import seaborn as sns

#sklearn

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.metrics import classification\_report,f1\_score

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import LabelEncoder

from imblearn.over\_sampling import SMOTE

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

import pickle

import warnings

warnings.filterwarnings('ignore')

from google.colab import files

uploades = files.upload()

**Activity 1.2: Read the dataset:**

* Our dataset format might be in .csv, excel files, .txt, .json, etc..

thyroid\_df = pd.read\_csv('/content/hypothyroid.csv')

thyroid\_df.head()thyroid\_df = pd.read\_csv('/content/hypothyroid.csv')

thyroid\_df.head()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | age | | sezx | on thyroxine | query on thyroxine | on antithyroid medication | sick |  | pr  eg  na  nt | thyroid surgery | I131 treatment | query hypothyroid | ... |  | TT  4  me  as  ur  ed | TT4 | T4U measured | T4U | FTI measured | FTI |  | T  B  G  m  ea  su  re  d |  | T  B  G | referral source |  | bin  ary  Cla  ss |
| 0 | 41 | F | f | F | f |  | f | f | f | f | F |  | ... | t | 125 | t | 1.14 | t |  | 10  9 |  | f | ? |  | SV  H  C | P |
| 1 | 23 | F | f | F | f |  | f | f | f | f | F |  | ... | t | 102 | f | ? | f |  | ? |  | f | ? |  | oth  er | P |
| 2 | 46 | M | f | F | f |  | f | f | f | f | F |  | ... | t | 109 | t | 0.91 | t |  | 12  0 |  | f | ? |  | oth  er | P |
| 3 | 70 | F | t | F | f |  | f | f | f | f | F |  | ... | t | 175 | f | ? | f |  | ? |  | f | ? |  | oth  er | P |
| 4 | 70 | F | f | F | f |  | f | f | f | f | F |  | ... | t | 61 | t | 0.87 | t |  | 70 |  | f | ? |  | SV  I | P |

5 rows × 30 columns

**Activity:2 Data cleaning:**

* **The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results.**
* **Handling missing values**
* **Descriptive analysis**
* **Splitting the dataset as x and y**
* **Handling categorical values**
* **Checking correlation**
* **Converting data type**
* **Splitting dataset into training and test set**
* **Handled Imbalanced Data**
* **Applying standard scalar.**

thyroid\_df.rename(columns={'binaryClass':'Label'},inplace=True)

thyroid\_df.replace('?',np.nan,inplace=True)

thyroid\_df.drop(['T3 measured','TSH measured','TT4 measured','T4U measured','FTI measured','TBG measured','TBG','referral source','on thyroxine','query on thyroxine','on antithyroid medication','query hypothyroid', 'query hyperthyroid','hypopituitary', 'psych'],axis=1,inplace=True)

thyroid\_df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 3772 entries, 0 to 3771

Data columns (total 15 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 age 3771 non-null object

1 sex 3622 non-null object

2 sick 3772 non-null object

3 pregnant 3772 non-null object

4 thyroid surgery 3772 non-null object

5 I131 treatment 3772 non-null object

6 lithium 3772 non-null object

7 goitre 3772 non-null object

8 tumor 3772 non-null object

9 TSH 3403 non-null object

10 T3 3003 non-null object

11 TT4 3541 non-null object

12 T4U 3385 non-null object

13 FTI 3387 non-null object

14 binaryClass 3772 non-null object

dtypes: object(15)

memory usage: 442.2+ KB

thyroid\_df['thyroid surgery'].value\_counts()

f 3719 t 53 Name: thyroid surgery, dtype: int64

cols = ['age','FTI','TSH','T3','TT4','T4U']

for i in cols:

    thyroid\_df[i] = pd.to\_numeric(thyroid\_df[i])

thyroid\_df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 3772 entries, 0 to 3771

Data columns (total 15 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 age 3771 non-null float64

1 sex 3622 non-null object

2 sick 3772 non-null object

3 pregnant 3772 non-null object

4 thyroid surgery 3772 non-null object

5 I131 treatment 3772 non-null object

6 lithium 3772 non-null object

7 goitre 3772 non-null object

8 tumor 3772 non-null object

9 TSH 3403 non-null float64

10 T3 3003 non-null float64

11 TT4 3541 non-null float64

12 T4U 3385 non-null float64

13 FTI 3387 non-null float64

14 binaryClass 3772 non-null object

dtypes: float64(6), object(9)

memory usage: 442.2+ KB

* **Handling missing values:**

:thyroid\_df.isnull().sum()

|  |  |
| --- | --- |
| Age | 1 |
| Sex | 150 |
| Sick | 0 |
| Pregnant | 0 |
| Thyroid surgery | 0 |
| T131 | 0 |
| Lithium | 0 |
| Goitre | 0 |
| Tumor | 0 |
| TSH | 369 |
| T3 | 769 |
| TT4 | 231 |
| T4U | 387 |
| FTI | 385 |
| Binary class | 0 |
| Dtype | Int64 |

for i in miss\_cols:

thyroid\_df.dropna(inplace=True)

thyroid\_df.isnull().sum()

age 0 sex 0 sick 0 pregnant 0 thyroid surgery 0 I131 treatment 0 lithium 0 goitre 0 tumor 0 TSH 0 T3 0 TT4 0 T4U 0 FTI 0 binaryClass 0 dtype: int64

thyroid\_df = thyroid\_df.drop(1364)

thyroid\_df.age = thyroid\_df.age.astype(int)

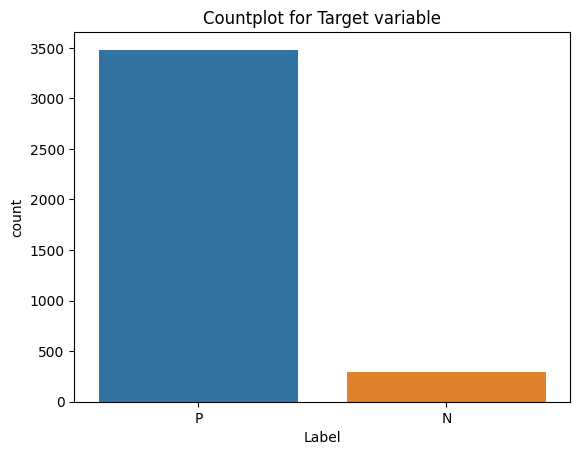
**Milestone 3: Exploratory Data Analysis**

**Activity 1: Descriptive Analysis:**

* Descriptive analysis is to study the basic features of data with the statistical process.
* Here pandas have a worthy function called describe.
* With this described function we can find mean, std, min, max and percentile values of continues features.

sns.countplot(x='Label',data=thyroid\_df)

plt.title("Countplot for Target variable");

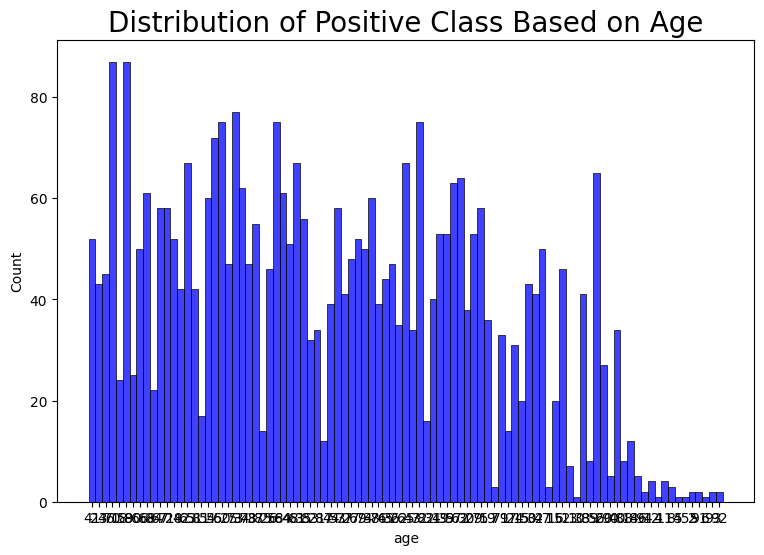
****

positive\_df = thyroid\_df[thyroid\_df.Label=='P']

plt.figure(figsize=(9,6))

sns.histplot(x='age',data=positive\_df,color='blue')

plt.title("Distribution of Positive Class Based on Age",{'fontsize':20});

****

plt.figure(figsize=(10,8))

plt.pie(x=positive\_df.sex.value\_counts(),

        labels=['Female','Male'],

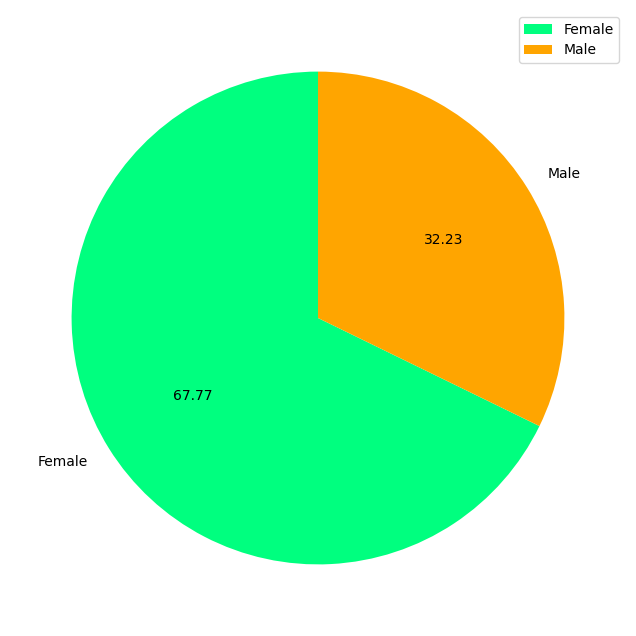
        startangle = 90,

        colors=['springgreen','orange'],

        autopct='%.2f'

       );

plt.legend();

****

**Inference:**

**Female patients who have disease is greater than female patients.**

plt.figure(figsize=(8,8))

plt.pie(x=positive\_df.sick.value\_counts(),

        labels=['Sick','Well'],

        startangle = 20,

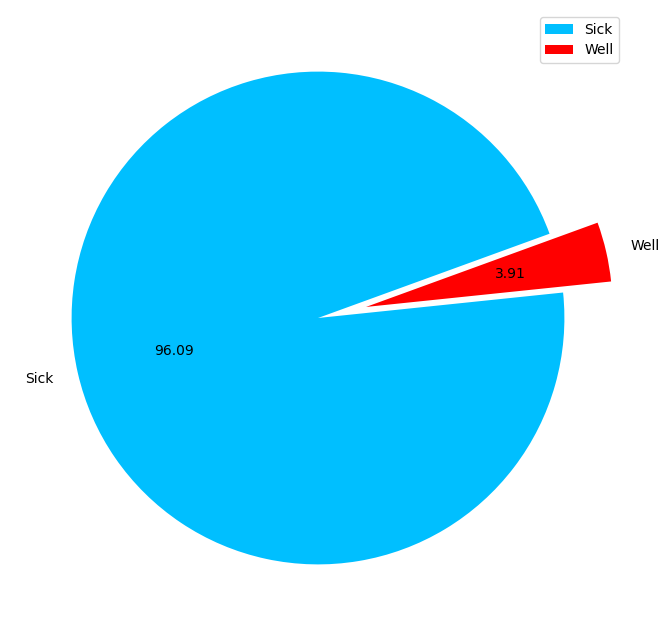
        colors=['deepskyblue','red'],

        autopct='%.2f',

        explode=[0,0.2]

       );

plt.legend();

****

X = thyroid\_df.drop('Label',axis=1)

y = thyroid\_df.Label

**Transform non-numerical labels to numerical labels:**

s\_encoder = LabelEncoder()

si\_encoder = LabelEncoder()

preg\_encoder = LabelEncoder()

th\_encoder = LabelEncoder()

treat\_encoder = LabelEncoder()

lith\_encoder = LabelEncoder()

g\_encoder= LabelEncoder()

tu\_encoder = LabelEncoder()

X['sex'] = s\_encoder.fit\_transform(X.sex)

X['I131 treatment'] = treat\_encoder.fit\_transform(X['I131 treatment'])

X['sick'] = si\_encoder.fit\_transform(X.sick)

X['pregnant'] = preg\_encoder.fit\_transform(X.pregnant)

X['thyroid surgery'] = th\_encoder.fit\_transform(X['thyroid surgery'])

X['lithium'] = lith\_encoder.fit\_transform(X['lithium'])

X['goitre'] = g\_encoder.fit\_transform(X['goitre'])

X['tumor'] = tu\_encoder.fit\_transform(X['tumor'])

def func(df):

    if df == 'P':

        return 1

    else:

        return 0

y = y.apply(func)

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.2,random\_state=11)

**Handled Imbalanced Data:**

smote = SMOTE(random\_state=11)

x\_smote, y\_smote = smote.fit\_resample(X\_train, y\_train)

print("Shape before the Oversampling : ",X\_train.shape)

print("Shape after the Oversampling : ",x\_smote.shape)

**Scaling:**

scalr = MinMaxScaler()

x\_smote.TT4 = scalr.fit\_transform(x\_smote[['TT4']])

x\_smote.age = scalr.fit\_transform(x\_smote[['age']])

x\_smote.FTI = scalr.fit\_transform(x\_smote[['FTI']])

**Model Building:**

models = {

    LogisticRegression(max\_iter=500):'Logistic Regression',

    SVC():"Support Vector Machine",

    RandomForestClassifier():'Random Forest'

}

for m in models.keys():

    m.fit(x\_smote,y\_smote)

for model,name in models.items():

     print(f"Accuracy Score for {name} is : ",model.score(X\_test,y\_test)\*100,"%")

Accuracy Score for Logistic Regression is : 92.95580110497238 %

Accuracy Score for Support Vector Machine is : 7.458563535911603 %

Accuracy Score for Random Forest is : 92.5414364640884 %

**Classification Report:**

for model,name in models.items():

    y\_pred = model.predict(X\_test)

    print(f"Classification Report for {name}")

    print("----------------------------------------------------------")

    print(classification\_report(y\_test,y\_pred))

    print("----------------------------------------------------------")

Classification Report for Logistic Regression

----------------------------------------------------------

precision recall f1-score support

0 1.00 0.06 0.11 54

1 0.93 1.00 0.96 670

accuracy 0.93 724

macro avg 0.96 0.53 0.53 724

weighted avg 0.93 0.93 0.90 724

----------------------------------------------------------

Classification Report for Support Vector Machine

----------------------------------------------------------

precision recall f1-score support

0 0.07 1.00 0.14 54

1 0.00 0.00 0.00 670

accuracy 0.07 724

macro avg 0.04 0.50 0.07 724

weighted avg 0.01 0.07 0.01 724

----------------------------------------------------------

Classification Report for Random Forest

----------------------------------------------------------

precision recall f1-score support

0 0.00 0.00 0.00 54

1 0.93 1.00 0.96 670

accuracy 0.93 724

macro avg 0.46 0.50 0.48 724

weighted avg 0.86 0.93 0.89 724

----------------------------------------------------------

rf = RandomForestClassifier()

rf.fit(x\_smote,y\_smote)

rf.score(X\_test,y\_test)

0.925414364640884

plt.figure(figsize=(9,7))

feature\_imp1 = rf.feature\_importances\_

sns.barplot(x=feature\_imp1, y=X.columns)

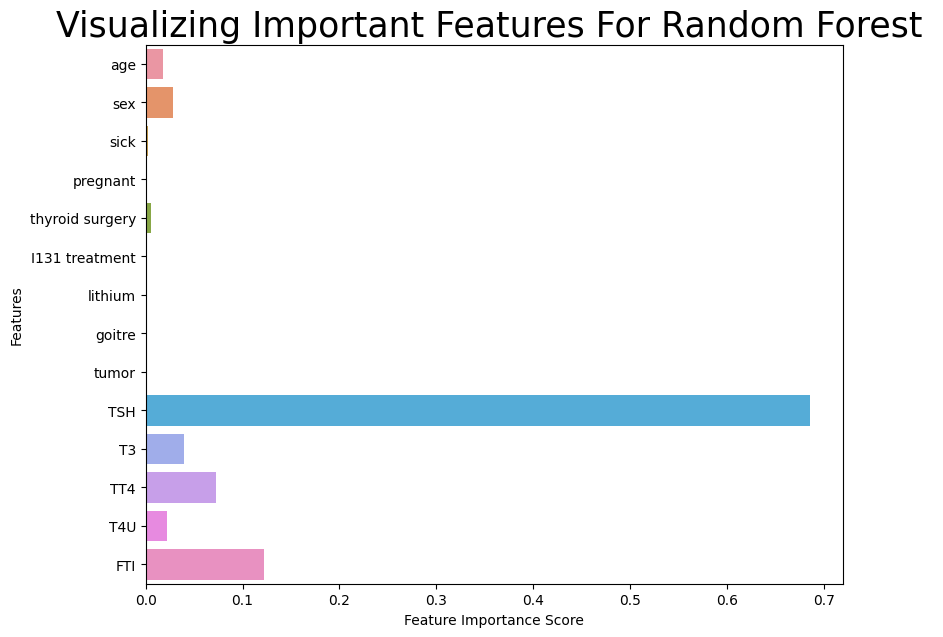
# Add labels to your graph

plt.xlabel('Feature Importance Score')

plt.ylabel('Features')

plt.title("Visualizing Important Features For Random Forest ",{'fontsize':25})

plt.show();



x\_smote.drop(['sick', 'pregnant', 'I131 treatment',

              'lithium', 'goitre', 'tumor'], axis=1, inplace=True)

X\_test.drop(['sick', 'pregnant', 'I131 treatment',

              'lithium', 'goitre', 'tumor'], axis=1, inplace=True)

new\_rf = RandomForestClassifier()

new\_rf.fit(x\_smote,y\_smote)

new\_rf.score(X\_test,y\_test)

0.9406077348066298

thyroid\_df.head()

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Age | Sex | Sick | Pregnant | Thyroid surgery | I131 Treatment | Lithium | Goiter | tumor |
| 0 | 41 | F | f | F | F | f | f | f | f |
| 1 | 23 | F | f | F | F | f | f | f | f |
| 2 | 46 | M | f | F | F | f | f | f | f |
| 3 | 70 | F | f | F | F | f | f | f | f |
| 4 | 70 | F | f | F | F | f | f | f | f |