

**CHIKKANNA GOVERNMENT ARTS COLLEGE**  
**DEPARTMENT OF BACHELOR OF COMPUTER**  
**APPLICATION**

**TIRUPUR-641602**

**(AFFILIATED TO BHARATHIAR UNIVERSITY)**



**TEAM MEMBERS NAME :**

**CHRISTYPUNITHA Y (2022J0001)**

**KOWSALYA G (2022J0004)**

**PREETHIKA P (2022J0006)**

**PRIYA S S (2022J0007)**

# **THYROID DISEASE CLASSIFICATION USING ML**

## **1. INTRODUCTION :**

### **1.1 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. A thyroid-related Blood test is used to detect this disease but it is often blurred and noise will be present. 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 - 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.

## **1.2. PURPOSE :**

### **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

➤ **Performance Testing & Hyperparameter Tuning**

- Testing model with multiple evaluation metrics
- Comparing model accuracy before & after applying hyperparameter tuning

➤ **Model Deployment**

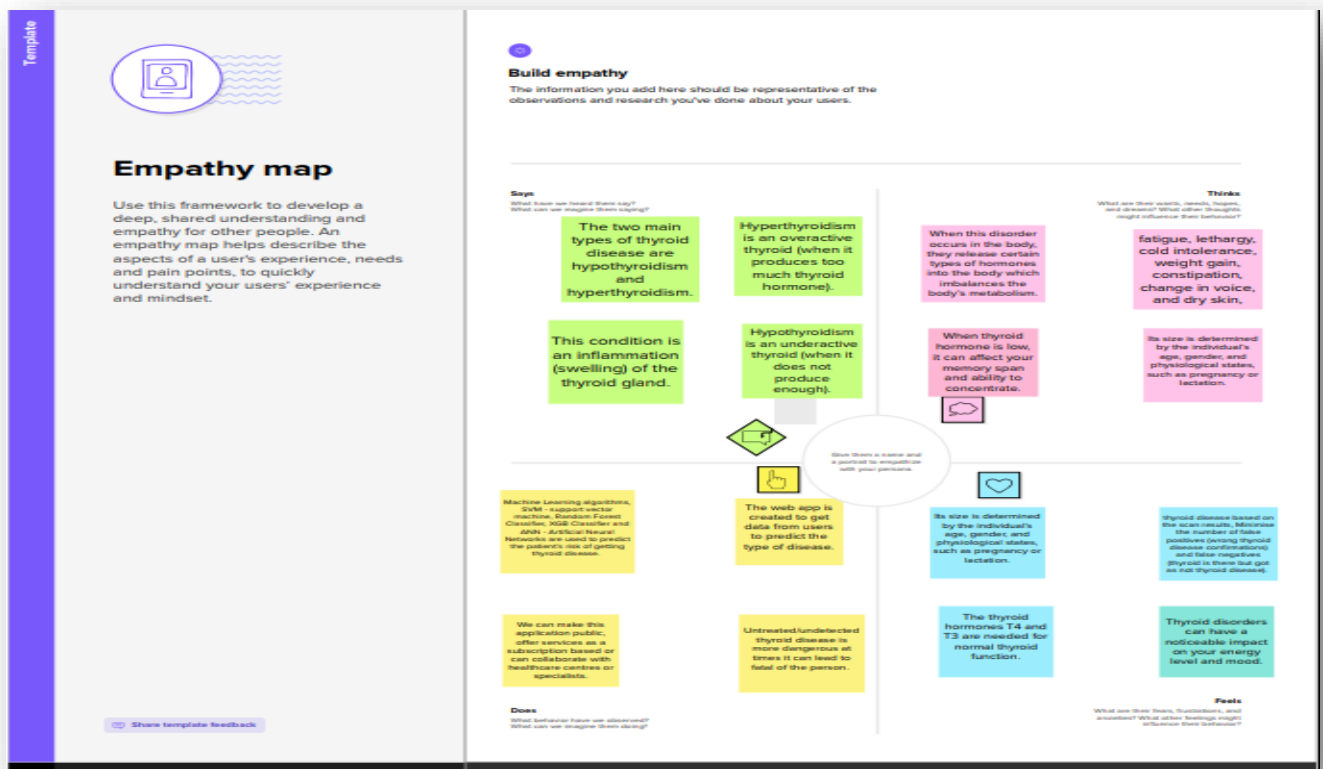
- Save the best model
- Integrate with Web Framework

➤ **Project Demonstration & Documentation**

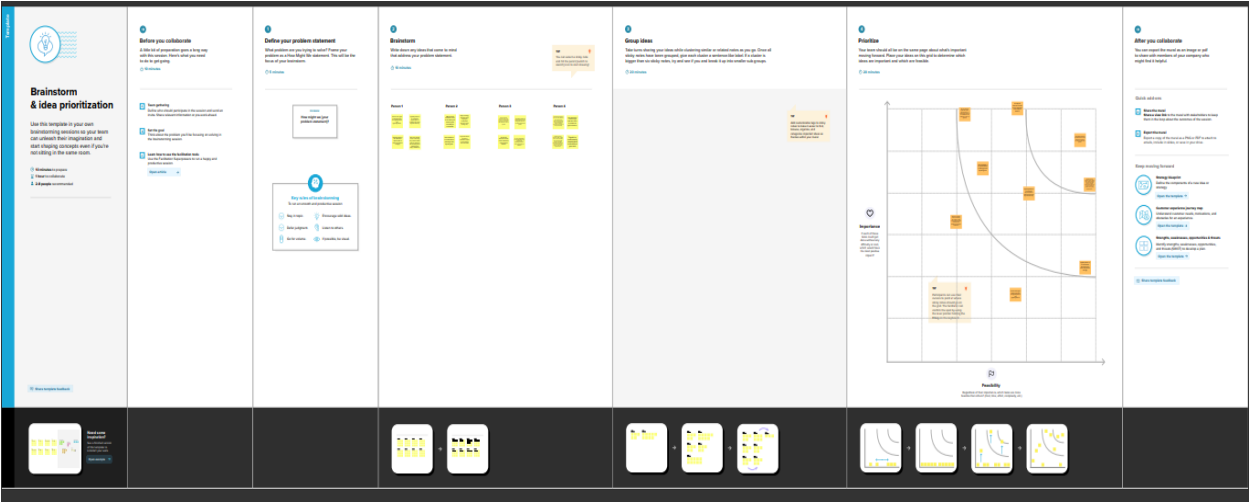
- Record explanation Video for project end to end solution
- Project Documentation-Step by step project development procedure.

## 2. PROBLEM DEFINITION & DESIGN THINKING

### 2.1. EMPATHY MAP

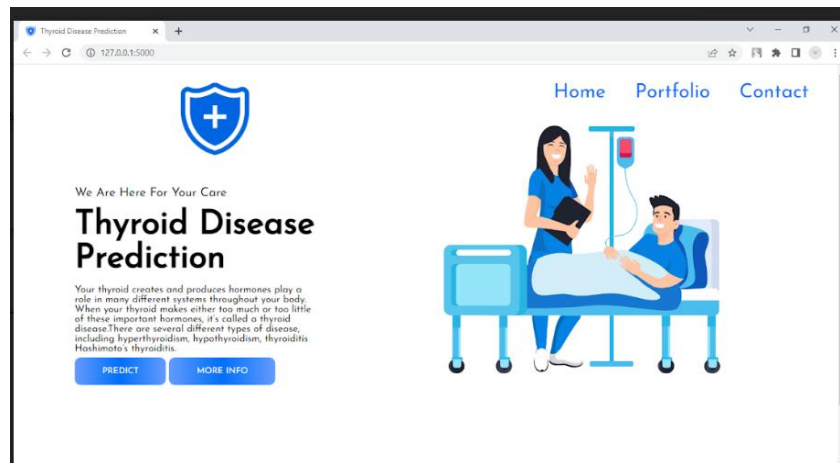


# 2.2 IDEATION & BRAINSTORMINGS MAP

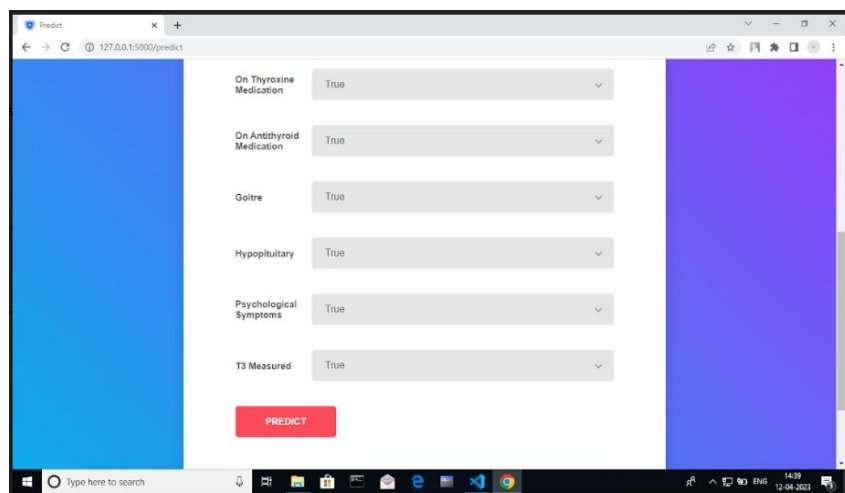


# 1.RESULT

## HOME PAGE :



## PREDICTION FORM :



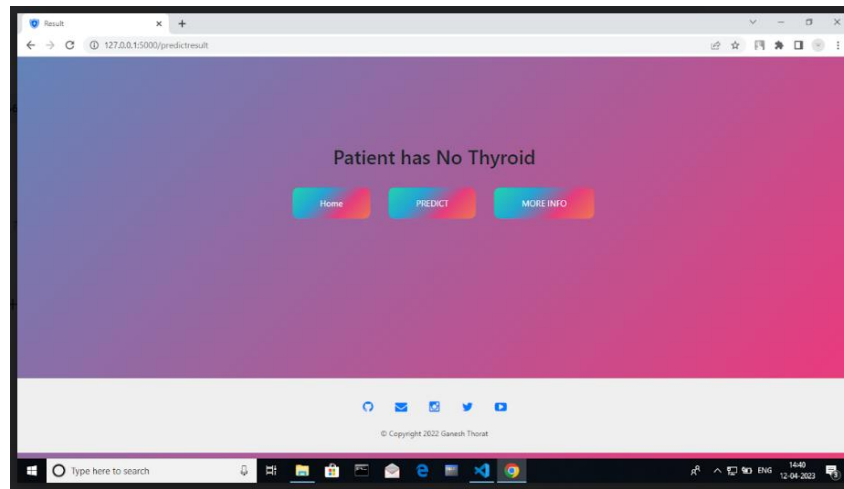
The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/predict". The page contains a form with six input fields, each with a dropdown menu set to "True". The fields are labeled: "On Thyroxine Medication", "On Antithyroid Medication", "Goitre", "Hypopituitary", "Psychological Symptoms", and "T3 Measured". Below these fields is a red button labeled "PREDICT". The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system clock in the bottom right corner indicates the time is 14:19 on 12-04-2023.

On Thyroxine Medication	True
On Antithyroid Medication	True
Goitre	True
Hypopituitary	True
Psychological Symptoms	True
T3 Measured	True

**PREDICT**



## RESULT PAGE :



## **4.ADVANTAGES & DISADVANTAGES**

### **ADVANTAGES :**

- It is readily available, so the patient doesn't need to wait.
- No chance of rejection.
- No need for major surgery.
- No need to take drugs, such as immuno-suppressants.

### **DISADVANTAGES :**

Thyroid disease is very common, with an estimated 20 million people in the United States having some type of thyroid disorder. A woman is about five to eight times more likely to be diagnosed with a thyroid condition than a man.

You may be at a higher risk of developing a thyroid condition if you:

- Have a family history of thyroid disease.
- Have a medical condition (these can include pernicious anemia, Type 1 diabetes, primary adrenal insufficiency, lupus, rheumatoid arthritis, Sjögren's syndrome and Turner syndrome).
- Take a medication that's high in iodine (amiodarone).
- Are older than 60, especially in women.
- Have had treatment for a past thyroid condition or cancer (thyroidectomy or radiation).

## 5. APPLICATION

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 T<sub>4</sub> and T<sub>3</sub> 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.

## 6. 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. This disease was classified using algorithms. Machine learning showed us good results using several algorithms and was built in the form of two models. In the first model, all the characteristics consisting of 16 inputs and one output were taken, and the result of the accuracy of the random forest algorithm was 98.93, which is the highest accuracy among the other algorithms. In the second embodiment, the following characteristics were omitted based on a previous study. The removed attributes were 1- query\_thyroxine 2- query\_hypothyroid 3-query\_hyperthyroid. Here we have included the increased accuracy of some algorithms, as well as the retention of the accuracy of others. It was observed that the accuracy of Naive Bayes algorithm increased the accuracy by 90.67. The highest precision of the MLP algorithm was 96.4 accuracy.

## 7. FUTURE SCOPE

disease is one of the most prevalent diseases worldwide, and it is mostly caused by a deficiency of iodine, but it may also be caused by other factors. The thyroid gland is an endocrine gland that secretes hormones and passes them through the bloodstream. It is situated in the middle of the front of the body. Thyroid gland hormones are responsible for aiding in digestion as well as maintaining the body moist, balanced, and so on. Thyroid gland treatments such as T3 (triiodothyronine), T4 (thyroid hormone), and TSH (thyroid stimulating hormone) are used to assess thyroid activity (thyroid stimulating hormone). Thyroid disorder is classified into two types: hypothyroidism and hyperthyroidism. Data mining is a semi-automated method of looking for correlations in massive datasets. Machine learning algorithms are one of the best solutions to many problems that are difficult to solve. Classification is a data extraction technique (machine learning) used to predict and identify many diseases, such as thyroid disease, which we researched and classified here because machine learning algorithms play a significant role in classifying thyroid disease and because these algorithms are high performing and efficient and aid in classification. Although the application of computer learning and artificial intelligence in medicine dates back to the early days of the field, there has been a new movement to consider the need for machine learning-driven healthcare

solutions. As a result, analysts predict that machine learning will become commonplace in healthcare in the near future. Hyperthyroidism is a disorder in which the thyroid gland releases so many thyroid hormones. Hyperthyroidism is caused by an increase in thyroid hormone levels. Dry skin, elevated temperature sensitivity, hair thinning, weight loss, increased heart rate, high blood pressure, heavy sweating, neck enlargement, nervousness, menstrual cycles shortening, irregular stomach movements, and hands shaking are some of the signs. Hypothyroidism is a condition in which the thyroid gland is underactive. Hypothyroidism is caused by a decline in thyroid hormone production. Hypo means deficient or less in medical terms. Inflammation and thyroid gland injury are the two primary causes of hypothyroidism. Obesity, low heart rate, increased temperature sensitivity, neck swelling, dry skin, hand numbness, hair issues, heavy menstrual cycles, and intestinal problems are some of the symptoms. If not treated, these symptoms can escalate over time.

## 8. APPENDIX

### A.SOURCE CODE :

```
!pip install scikit-learn
!pip install imbalanced-learn
!pip install mlxtend
!pip install pandas-profiling
```

Import Libraries:

```
import pandas as pd
import seaborn as sns
import sklearn
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from mlxtend.feature_selection import SequentialFeatureSelector as sfs
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from pandas_profiling import ProfileReport
import warnings
warnings.filterwarnings("ignore")

df=pd.read_csv('/content/drive/MyDrive/allhypoDATA.CSV')
df.head()

#Split classes column
col_mod=df.classes.str.split('.',expand=True)
df[['last','final']]=col_mod
data=df.drop(['classes','final'],axis=1)

df.info()

df.describe()

df.shape

#Save Dataframe to csv file
df.to_csv('Thyroid_Data.csv')

#Number of Null values in Dataset
df.isnull().sum()

#Percentage of null values in dataset
```



```
df.isnull().sum()/df.shape[0]*100
```

```
#TBG having 100% null values so will drop that column
```

```
df=df.drop('TBG_measured',axis=1)
```

```
categorical_features=df.select_dtypes(exclude='number')
```

```
categorical_features.describe()
```

```
for feature in categorical_features:
```

```
    print('-----')
```

```
    print(f"{feature}:{categorical_features[feature].unique()}")
```

```
from matplotlib import pyplot as plt
```

```
fig, axes = plt.subplots(3,2, figsize=(18, 10))
```

```
sns.boxplot(ax=axes[0, 0], data=df, x='classes', y='age')
```

```
sns.boxplot(ax=axes[0, 1], data=df, x='classes', y='TSH')
```

```
sns.boxplot(ax=axes[1, 0], data=df, x='classes', y='T3')
```

```
sns.boxplot(ax=axes[1, 1], data=df, x='classes', y='T4')
```

```
fig, axes = plt.subplots(3,2, figsize=(18, 10))
```

```
from sklearn.impute import SimpleImputer
```

```
#Handle numerical features
```

```
simple_imputer=SimpleImputer(strategy='median')
```

```
numerical_missing=pd.DataFrame(simple_imputer.fit_transform(df.select_dtypes(
exclude='O')))
```

```
#Handle categorical features
```

```
cat_imputation=SimpleImputer(strategy='most_frequent')
```

```
categorical_missing=pd.DataFrame(cat_imputation.fit_transform(df.select_dtypes(  
exclude='number')))
```

```
numerical_missing.columns=df.select_dtypes(exclude='O').columns
```

```
categorical_missing.columns=df.select_dtypes(exclude='number').columns
```

```
data=pd.concat([numerical_missing,categorical_missing],axis=1)
```

```
data.head()
```

```
data.isnull().sum()
```

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

```
data=data.drop(data.age.index[1364])
```

```
outliers_data=outliers_removal(data)
```

```
outliers_data
```

```
new_df=data.drop(outliers_data.index)
```

```
new_df.head()
```

```
new_df.shape
```

```
new_df.columns
```

```
new_df.to_csv('Preprocessed_data.csv',index=False)
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2,  
random_state=1)
```

```
X_train.shape,X_test.shape,Y_train.shape,Y_test.shape
```

```
ordinal_encoder = OrdinalEncoder()
```

```
X_train_cat_encoded =
```

```
pd.DataFrame(ordinal_encoder.fit_transform(X_train.select_dtypes(exclude='num  
ber')))
```

```
X_train_cat_encoded.columns = X_train.select_dtypes(exclude='number').columns
```

```
X_test_cat_encoded =
```

```
pd.DataFrame(ordinal_encoder.transform(X_test.select_dtypes(exclude='number'))  
)
```

```
X_test_cat_encoded.columns = X_test.select_dtypes(exclude='number').columns
```

```
label_encoder = LabelEncoder()
```

```
Y_train_cat_encoded= pd.DataFrame(label_encoder.fit_transform(Y_train))
```

```
print(Y_train_cat_encoded.value_counts())
```

```
print(Y_train.value_counts())
```

```
Y_test_cat_encoded = pd.DataFrame(label_encoder.transform(Y_test))
```

```
sc = StandardScaler()
```

```
X_train_sc=pd.DataFrame(sc.fit_transform(X_train.select_dtypes(exclude='O')))
X_test_sc=pd.DataFrame(sc.transform(X_test.select_dtypes(exclude='O')))
```

```
X_train_sc.columns=X_train.select_dtypes(exclude='O').columns
X_test_sc.columns=X_test.select_dtypes(exclude='O').columns
```

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

```
X_train_resample,Y_train_resample=SMOTE(random_state=0,k_neighbors=1).fit_
_resample(X_train_final,Y_train_cat_encoded)
X_test_resample,Y_test_resample=SMOTE(random_state=0,k_neighbors=1).fit_r
esample(X_test_final,Y_test_cat_encoded)
```

```
X_train_resample.shape,X_test_resample.shape,Y_train_resample.shape,Y_test_re
sample.shape
```

```
from mlxtend.feature_selection import SequentialFeatureSelector as sfs
from sklearn.ensemble import RandomForestClassifier
```

```
print('Training dataset shape:', X_train_resample.shape, Y_train_resample.shape)
print('Testing dataset shape:', X_test_resample.shape, Y_test_resample.shape)
```

```
Y_train_resample_flat = Y_train_resample.to_numpy().ravel()
Y_test_resample_flat = Y_test_resample.to_numpy().ravel()
```

```
print('Training dataset shape:', X_train_resample.shape,
Y_train_resample_flat.shape)
```

```
print('Testing dataset shape:', X_test_resample.shape, Y_test_resample_flat.shape)
```

```
rfc = RandomForestClassifier(n_estimators=100, max_depth=5)
```

```
forward_fs = sfs(rf ,
```

```
k_features=10,forward=True,floating=False,verbose=2,scoring='accuracy',cv=5)
```

```
forward_fs = forward_fs.fit(X_train_resample, Y_train_resample_flat)
```

```
feat_names = list(forward_fs.k_feature_names_)
```

```
print(feat_names)
```

```
X_train_new=X_train_resample[['age','sex','TSH', 'TT4', 'FTI', 'on_thyroxine',  
'on_antithyroid_medication', 'goitre', 'hypopituitary', 'psych', 'T3_measured',  
'referral_source']]
```

```
X_test_new=X_test_resample[['age','sex','TSH', 'TT4', 'FTI', 'on_thyroxine',  
'on_antithyroid_medication', 'goitre', 'hypopituitary', 'psych', 'T3_measured',  
'referral_source']]
```

```
rf_model=rf.fit(X_train_new,Y_train_resample_flat)
```

```
def print_Score(clf,x_train,x_test,y_train,y_test,train=True):
```

```
    if train:
```

```
        pred=clf.predict(x_train)
```

```
clf_report=pd.DataFrame(classification_report(y_train,pred,output_dict=True))
```

```
    print("Train Result:\n=====")
```

```

print(f"Accuracy Score:{accuracy_score(y_train,pred)*100:.2f}% ")
print("-----")
print(f"Classification Report:\n{clf_report}")
print("-----")
print(f"Confusion Matrix:\n{confusion_matrix(y_train,pred)}\n")
elif train==False:
    pred=clf.predict(x_test)
    clf_report=pd.DataFrame(classification_report(y_test,pred,output_dict=True))
    print("Test Result:\n=====")
    print(f"Accuracy Score:{accuracy_score(y_test,pred)*100:.2f}% ")
    print("-----")
    print(f"Classification Report:\n{clf_report}")
    print("-----")
    print(f"Confusion Matrix:\n{confusion_matrix(y_test,pred)}\n")

print_Score(rf_model,X_train_new,X_test_new,Y_train_resample_flat,Y_test_resa
mple_flat,train=True)

## Logistic Regression

lr=LogisticRegression(random_state=0,max_iter=10)

lr_model=lr.fit(X_train_new,Y_train_resample_flat)

lr_train_score=print_Score(lr_model,X_train_new,X_test_new,Y_train_resample_
flat,Y_test_resample_flat,train=True)

```

```
lr_test_score=print_Score(lr_model,X_train_new,X_test_new,Y_train_resample_flat,Y_test_resample_flat,train=False)
```

## Navie Bayes Classification

```
gnb=GaussianNB()
```

```
gnb_model=gnb.fit(X_train_new,Y_train_resample_flat)
```

```
gnb_train_score=print_Score(gnb_model,X_train_new,X_test_new,Y_train_resample_flat,Y_test_resample_flat,train=True)
```

```
gnb_test_score=print_Score(gnb_model,X_train_new,X_test_new,Y_train_resample_flat,Y_test_resample_flat,train=False)
```

## Decision Tree

```
dtc=DecisionTreeClassifier(random_state=0,max_depth=10,min_samples_split=5)
```

```
dt_model=dtc.fit(X_train_new,Y_train_resample_flat)
```

```
dt_train_score=print_Score(dt_model,X_train_new,X_test_new,Y_train_resample_flat,Y_test_resample_flat,train=True)
```

```
dt_test_model=print_Score(dt_model,X_train_new,X_test_new,Y_train_resample_flat,Y_test_resample_flat,train=False)
```

## KNN

```
knn=KNeighborsClassifier()
```

```
knn_model=knn.fit(X_train_new,Y_train_resample_flat)
```

```
knn_train_score=print_Score(knn_model,X_train_new,X_test_new,Y_train_resam  
ple_flat,Y_test_resample_flat,train=True)
```

```
knn_test_score=print_Score(knn_model,X_train_new,X_test_new,Y_train_resamp  
le_flat,Y_test_resample_flat,train=False)
```

```
profile=ProfileReport(data,title="Thyroid Disease Detection",explorative=True)
```

```
profile.to_file("EDA Report.html")
```

```
profile.to_notebook_iframe()
```