Capstone Project

Project Title: Thyroid Disease Prediction Using ML Technique

Abstract

Thyroid disease (TD) is one of the most progressive endocrine disorders in the human population today. According to statistics, thyroid disorders are on the rise in India. Approximately 1 in 10 Indian adults suffer from hypothyroidism, a condition in which the thyroid gland does not produce enough thyroid hormones to meet the needs of the body, and also one-third of thyroid patients are unaware of their condition. Prediction of endocrine disease is a critical task in the field of clinical data analysis.

Machine Learning (ML) has shown effective results in the decision-making and predictions from the enormous data generated by the healthcare domain. This project has proposed four models based on the primary thyroid disease dataset collected from 3772 patients from the Garvan Institute in Sydney, Australia. In these models, we have compared 4 ML algorithms (Logistic Regression, KNN, Support Vector Machine, Decision Tree) for effective classification and built an optimum model to predict hypothyroidism disease.

Submitted By:

Bhumika S

TABLE OF CONTENT

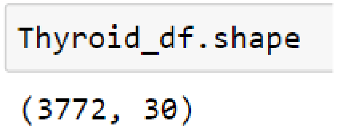
1. INTRODUCTION

Advanced machine biology is used in the area of healthcare. It allowed the collection of the stored patient data for medical disease prediction. For early-stage disease detection, various intelligent prediction algorithms are used. The Medical information system is rich in data sets, but there are no intelligent systems that can easily analyze the disease. Over time, machine learning algorithms play a crucial role in solving complex and nonlinear problems in developing a prediction model. In any disease prediction model, the features that can be selected from the different datasets are required, which can easily be used as a classification of unhealthy patients as precisely as possible. Otherwise, misclassification may result in an unhealthy patient who missed out on treatment. The reality of forecasting any condition associated with thyroid illness is also of the greatest cardinal number

2. IMPORTANCE OF PROJECT

Disease diagnosis involves analyzing symptoms and detecting whether a disease persists in a body, but analyzing symptoms itself is a complex task. Providing disease diagnosis at early stages with higher accuracy is an important task. If left untreated hypothyroidism can cause elevated cholesterol levels, an increase in blood pressure, cardiovascular complications, decreased fertility, and depression. In pregnant women, thyroid disorders can lead to placental abnormalities and increased risks for the baby's well-being. If this disease is detected in an earlier stage, then doctors can give proper treatment to the patients. Collecting all the past data, analyzing it with the help of different algorithms, and comparing the results.

3. DATASET

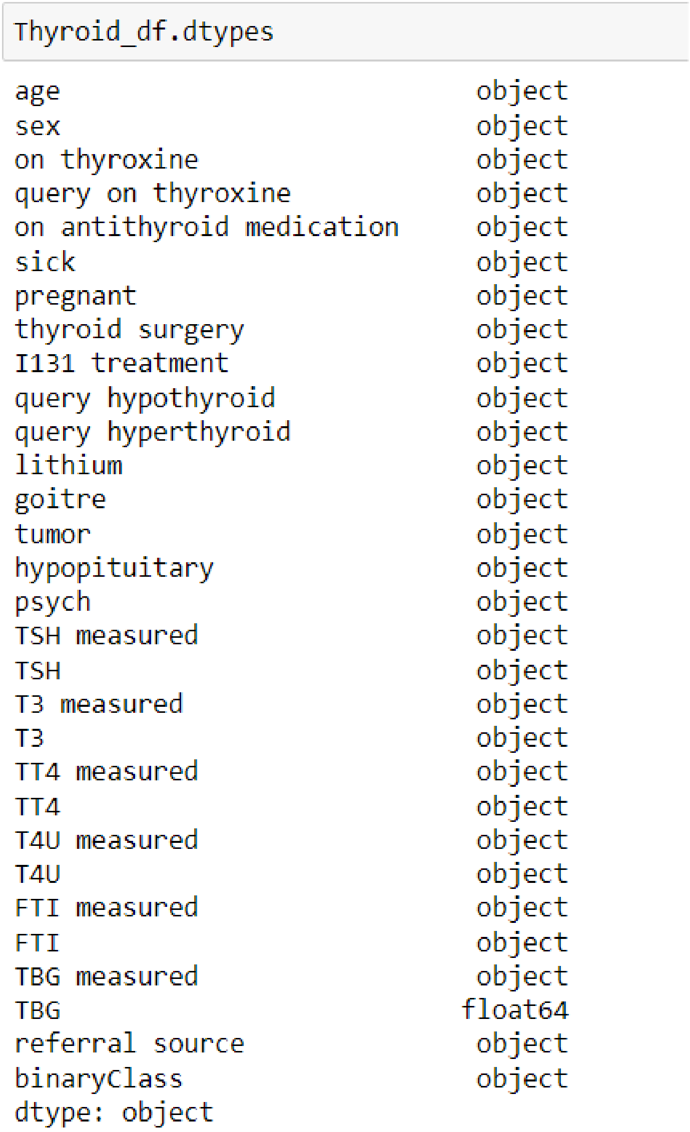


```
Thyroid_df.shape
```

```
(3772, 30)
```

We see the data frame has 3772 observations and 30 features, where 29 features are independent features and one target feature is binary class i.e., (P or N) categorical.

1. All 28 independent features are categorical



```
Thyroid_df.dtypes
```

```
age                            object
sex                            object
on thyroxine                   object
query on thyroxine             object
on antithyroid medication      object
sick                           object
pregnant                       object
thyroid surgery                object
I131 treatment                 object
query hypothyroid              object
query hyperthyroid             object
lithium                        object
goitre                         object
tumor                          object
hypopituitary                  object
psych                          object
TSH measured                   object
TSH                            object
T3 measured                    object
T3                             object
TT4 measured                   object
TT4                            object
T4U measured                   object
T4U                            object
FTI measured                   object
FTI                            object
TBG measured                   object
TBG                           float64
referral source                object
binaryClass                    object
dtype: object
```

But as per data attribute information we have, "age", "FTI", "TSH", "T3", "TT4", and "T4U" variables are numerical, which are wrongly interpreted as categorical, so we convert these variables into "numeric" data type.

Independent Features description:

1. **Age:** Age of the patient - (Numerical)
2. **Sex:** Gender of the patient - (Male, Female) - (Categorical)
3. **On Thyroxine:** Details of patients whether taking thyroxine medication or not, which is used to treat hypothyroidism - (T, F) - (Categorical)
4. **Query on Thyroxine:** Checking whether the patient taking thyroxine or not - (T, F) - (Categorical)

5. **On Antithyroid medication:** Details of patients whether taking an antithyroid medication or not, which is used to treat hyperthyroidism - (T, F) -(Categorical)
6. **Sick:** Details of patients whether they are sick or not - (T, F) - (Categorical)
7. **Pregnant:** Details of patients whether they are pregnant or not - (T, F) - (Categorical)
8. **Thyroid surgery:** Details of patients whether they have undergone thyroid surgery or not - (T, F) - (Categorical)
9. **I131 treatment:** Details of patients whether taking I131 treatment or not, which is used to treat hyperthyroidism or thyroid cancer - (T, F) - (Categorical)
10. **Query hypothyroid:** Checking whether the patient has hypothyroid or not - (T, F) - (Categorical)
11. **Query hyperthyroid:** Checking whether the patient has hyperthyroid or not - (T, F) - (Categorical)
12. **Lithium:** The presence of lithium inpatient or not - (T, F) - (Categorical)
13. **Goiter:** The presence of Goiter inpatient or not - (T, F) - (Categorical)
14. **Tumor:** The presence of tumor inpatient or not - (T, F) - (Categorical)
15. **Hypopituitary:** The presence of hypopituitary inpatient or not - (T, F) - (Categorical)
16. **Psych:** The presence of psych inpatient or not - (T, F) - (Categorical)
17. **TSH measured:** Checking whether TSH is measured inpatient or not - (T, F) - (Categorical)
18. **TSH:** Patient's TSH level (mIU/L) - (Numerical)
19. **T3 measured:** Checking whether T3 is measured inpatient or not - (T, F) - (Categorical)
20. **T3:** Patient's T3 level (ng/dl) - (Numerical)
21. **TT4 measured:** Checking whether TT4 is measured inpatient or not - (T, F) - (Categorical)
22. **TT4:** Patient's TT4 level (µg/dL) - (Numerical)
23. **T4U measured:** Checking whether T4U is measured inpatient or not - (T, F) - (Categorical)
24. **T4U:** Patient's T4U level - (Numerical)
25. **FTI measured:** Checking whether FTI is measured inpatient or not - (T, F) - (Categorical)
26. **FTI:** Patient's FTI level - (Numerical)
27. **TBG measured:** Checking whether TBG is measured inpatient or not - (T, F) - (Categorical)
28. **TBG:** Patient's TT4 level (mg/dl) - (Numerical)
29. **Referral source:** Patient's referral source - (Categorical)

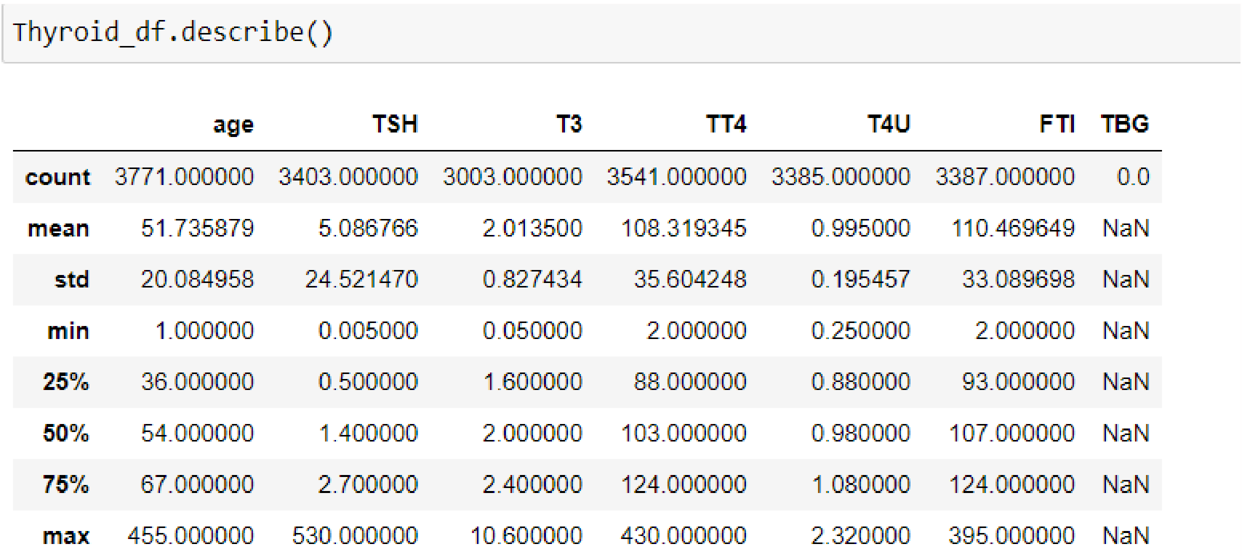
Units:

1. **mIU/L:** Milli-international units per liter.
2. **ng/dl:** Nanograms per deciliter.
3. **µg/dL:** Microgram per deciliter.
4. **mg/dl:** Milligrams per deciliter.

Target Feature description:

1. **Target:** Patient has Thyroid disease or not. Positive – 1, Negative – 0

4. SUMMARY OF DATA



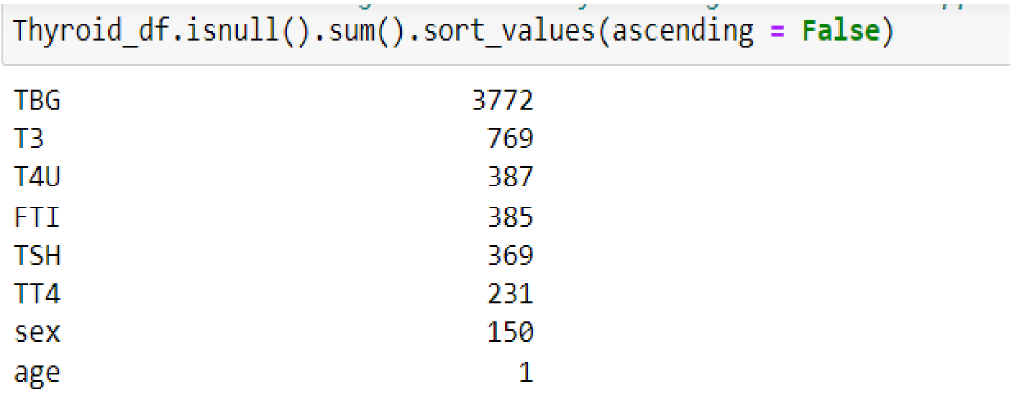
```
Thyroid_df.describe()
```

	age	TSH	T3	TT4	T4U	FTI	TBG
count	3771.000000	3403.000000	3003.000000	3541.000000	3385.000000	3387.000000	0.0
mean	51.735879	5.086766	2.013500	108.319345	0.995000	110.469649	NaN
std	20.084958	24.521470	0.827434	35.604248	0.195457	33.089698	NaN
min	1.000000	0.005000	0.050000	2.000000	0.250000	2.000000	NaN
25%	36.000000	0.500000	1.600000	88.000000	0.880000	93.000000	NaN
50%	54.000000	1.400000	2.000000	103.000000	0.980000	107.000000	NaN
75%	67.000000	2.700000	2.400000	124.000000	1.080000	124.000000	NaN
max	455.000000	530.000000	10.600000	430.000000	2.320000	395.000000	NaN

- As there are a total of 3772 patients in a survey.
- The variable "TBG" has complete missing values, so maybe removing these variables will be a good option
- The age of the patients ranges from a year old to a maximum of 455 years old. Which indicates outliers present in age, which need to be treated.
- The variables "age", "TSH", "T3", "TT4", "T4U", "FTI", "sex", has few missing values which can be treated in later steps.

5. EXPLORATORY DATA ANALYSIS

5.1. Treating missing data

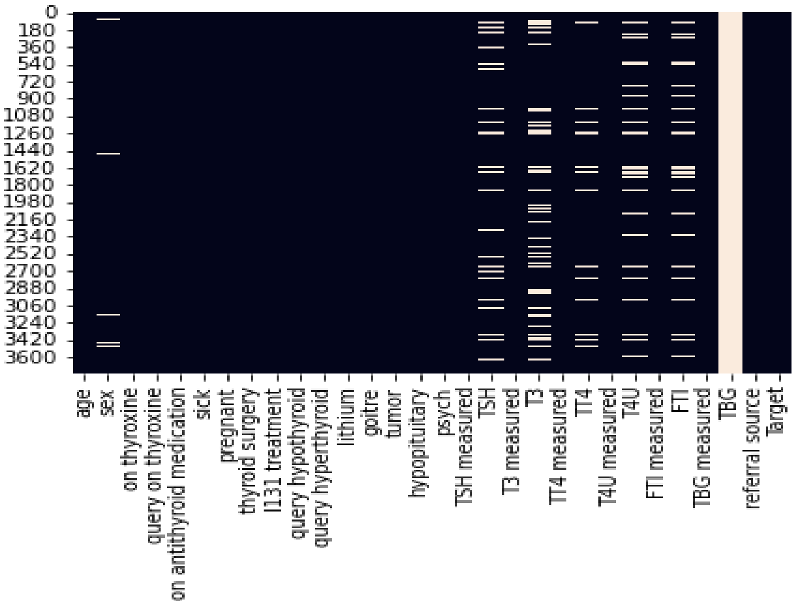


```
Thyroid_df.isnull().sum().sort_values(ascending = False)
```

```
TBG         3772
T3           769
T4U          387
FTI          385
TSH          369
TT4          231
sex          150
age            1
```

- isnull().sum()"" returns the number of missing values in each variable.

visualization of missing values using the heatmap:



There are horizontal lines in the heatmap for "age","TSH","T3","TT4","T4U","FTI","sex", and "TBG", which would correspond to probable missing values.

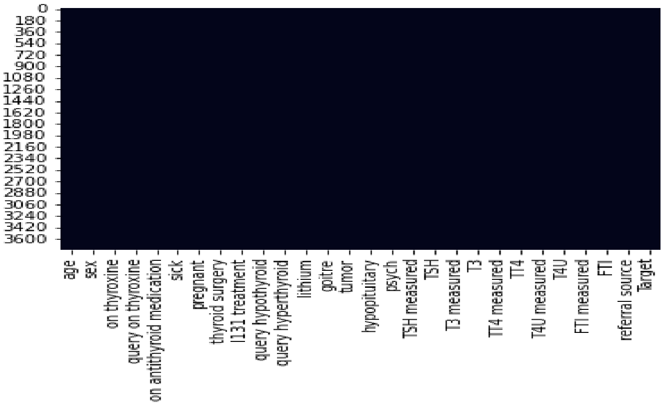
Why do you need to fill in the missing data?

1. Because most of the machine learning models that you want to use will provide an error if you pass NaN values into it.

Methods used to treat missing values:

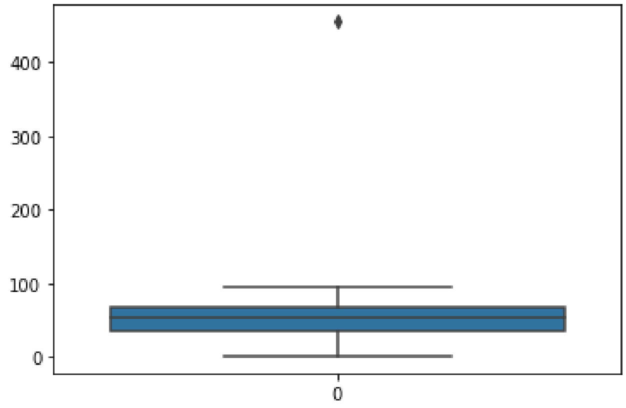
1. Fill the missing data with the mean or median value if it's a numerical variable.
2. Filling the missing data with mode if it's a categorical value.
3. Deleting the columns with complete missing values.

visualization of missing values using the heatmap after treatment:



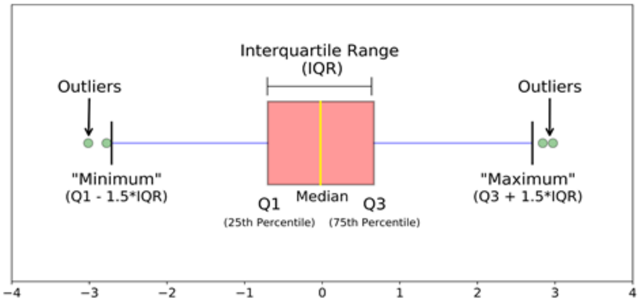
5.2. Discover Outliers for age using Box plot

An **Outlier** is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is vastly larger or smaller than the remaining values in the set.



Importance of treating outliers

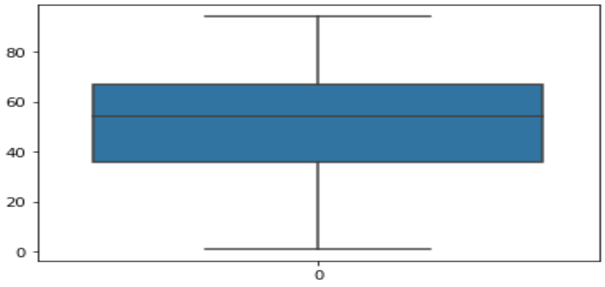
1. If the outliers are not removed, the model accuracy may decrease.



How to treat outliers:

1. calculate the 1st and 3rd quartiles (Q1, Q3)
2. compute IQR=Q3-Q1
3. compute lower bound = (Q1–1.5*IQR), upper bound = (Q3+1.5*IQR)

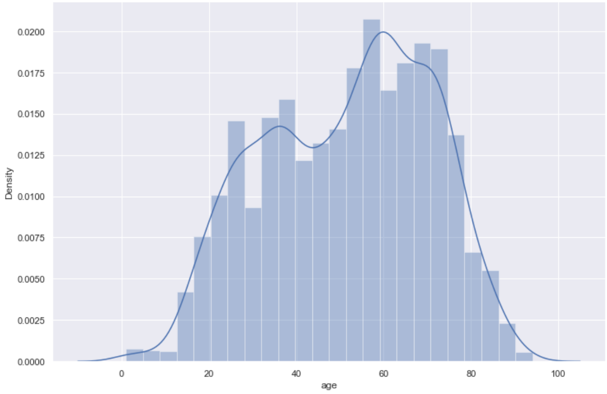
visualizing boxplot for age after treatment:



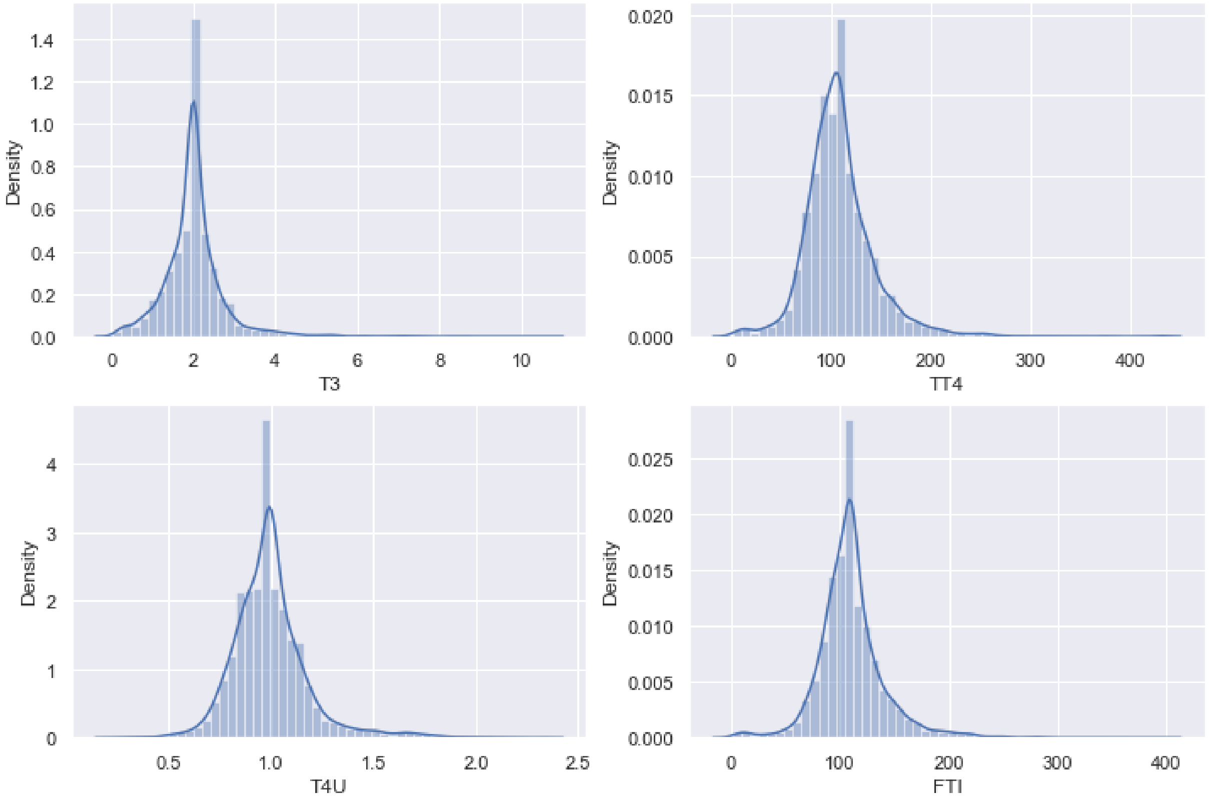
5.3. Distribution of Variables

Distribution of numeric independent variables using dist plot:

distplot() function is used to plot the distplot. The distplot represents the univariate distribution of data i.e., the data distribution of a variable against the density distribution.



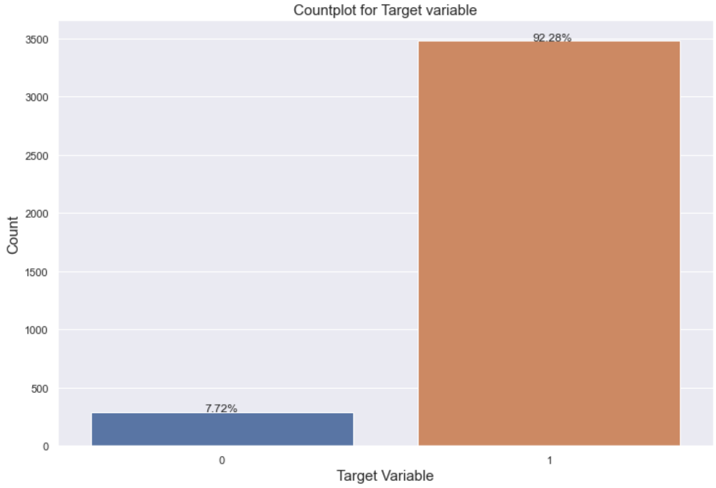
The variable "age" is near normally distributed



It can be seen that the variable "T3", "TT4", "T4U", and "FTI" have slight right skew, but has a right tail. They are almost near normally distributed

Distribution of dependent variable using count plot:

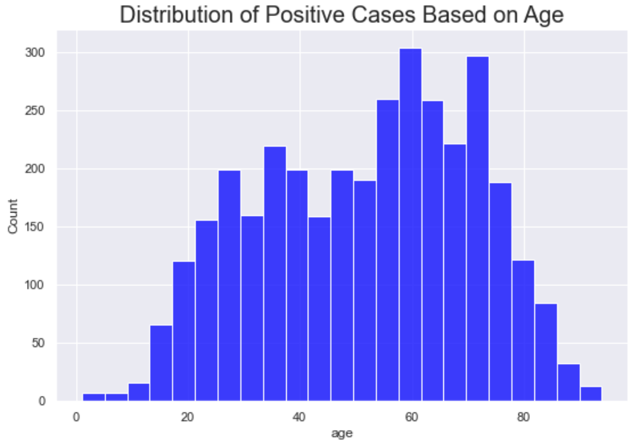
Countplot() get counts of 0's and 1's in the "target" variable



Here a total of 7.72% of patients do not have the disease, and 92.28% of patients are affected by Hypothyroidism. We see that there is a huge imbalance between the two classes of the target variable. Can be handled using smote.

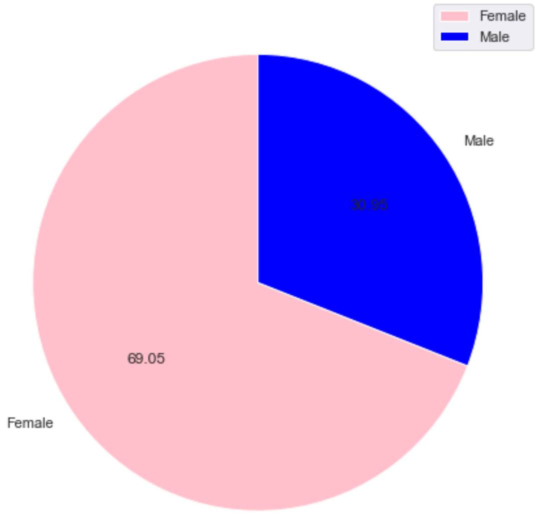
Distribution of Positive Cases Based on "Age" using hist plot:

histplot() is used to illustrate the major features of the distribution of the data in a convenient form.



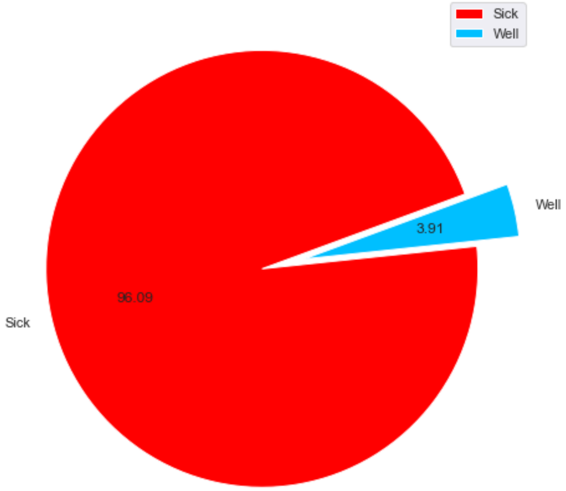
This condition affects people of all age groups but most patients who suffer from thyroid belong to the age group between 20 and 70.

Distribution of Positive Cases Based on GENDER:



Female patients who have a disease are greater than male patients.

Distribution of Positive Cases Based on SICK:

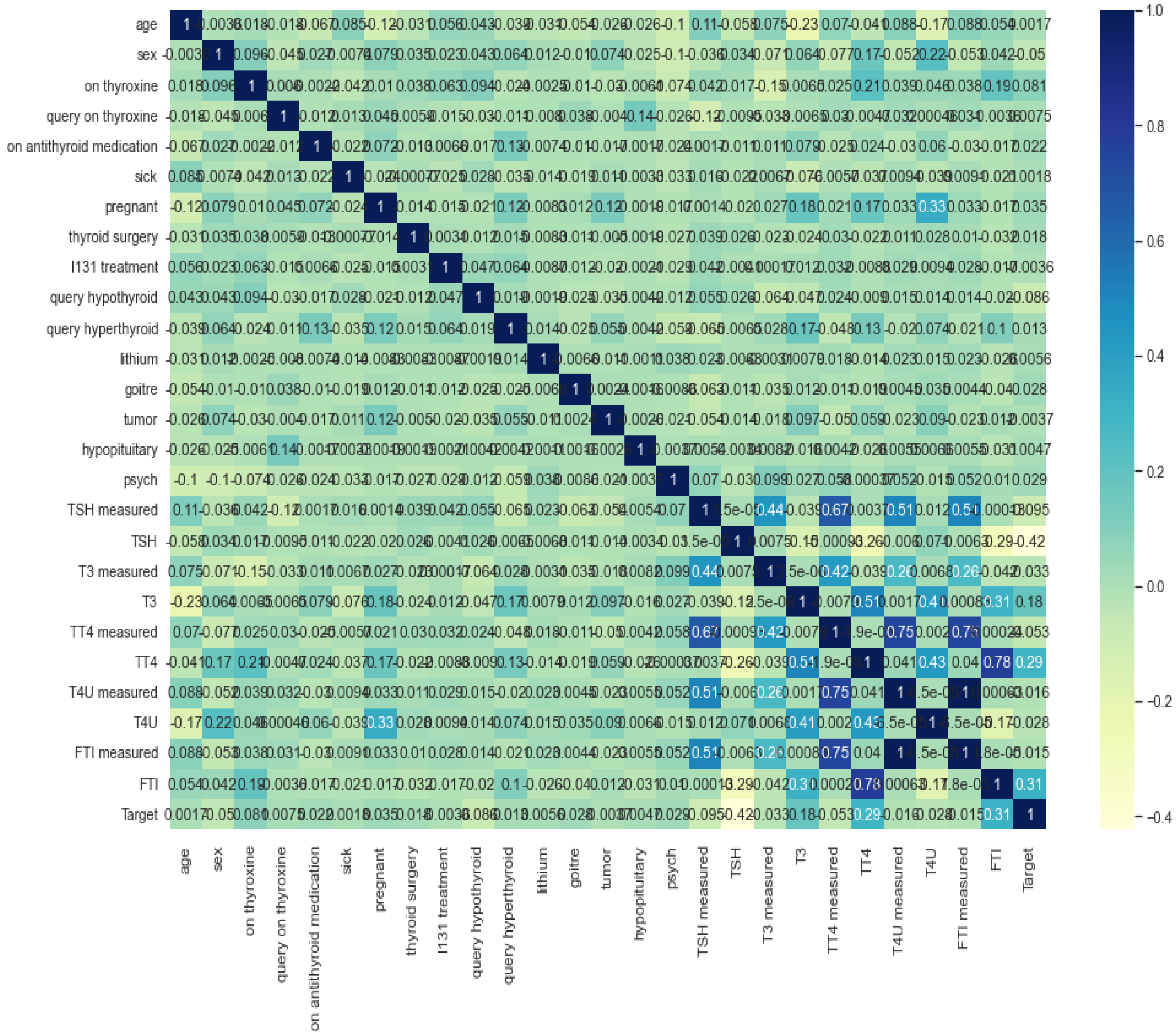


About 96% of patients who has a disease are sick.

5.4. Multivariate Analysis

Correlation is a statistic that measures the degree to which two variables move with each other. A correlation coefficient near 1 indicates a strong relationship between them; a weak correlation indicates the extent to which one variable increases as the other decreases. Correlation among multiple variables can be represented in the form of a matrix. This allows us to see which variables are correlated.

Heat map for the correlation matrix:

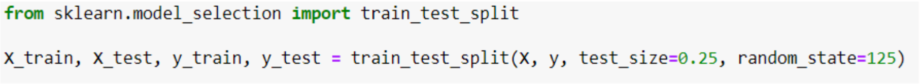


From the above heatmap, it can be seen that the continuous variables are not highly correlated.

6. TRAINING AND TEST DATASET

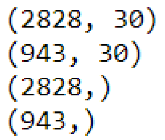
Creating the training dataset with 75% of original data and the remaining 25% as test data.

- **Train Dataset**: Used to fit the machine learning model.
- **Test Dataset**: Used to evaluate the fit machine learning model.

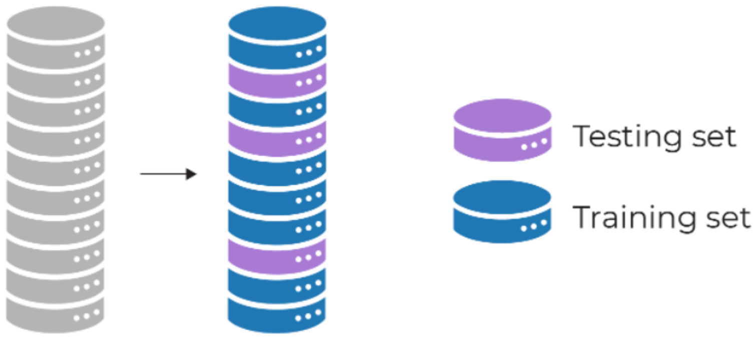


```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=125)
```

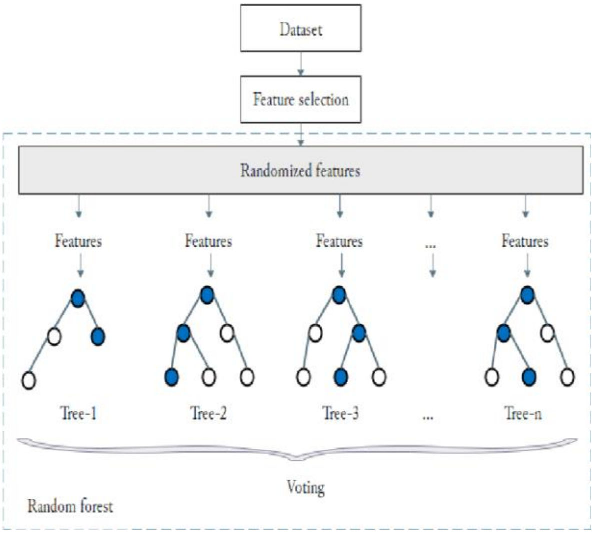
Training and Test dataset



```
(2828, 30)
(943, 30)
(2828,)
(943,)
```



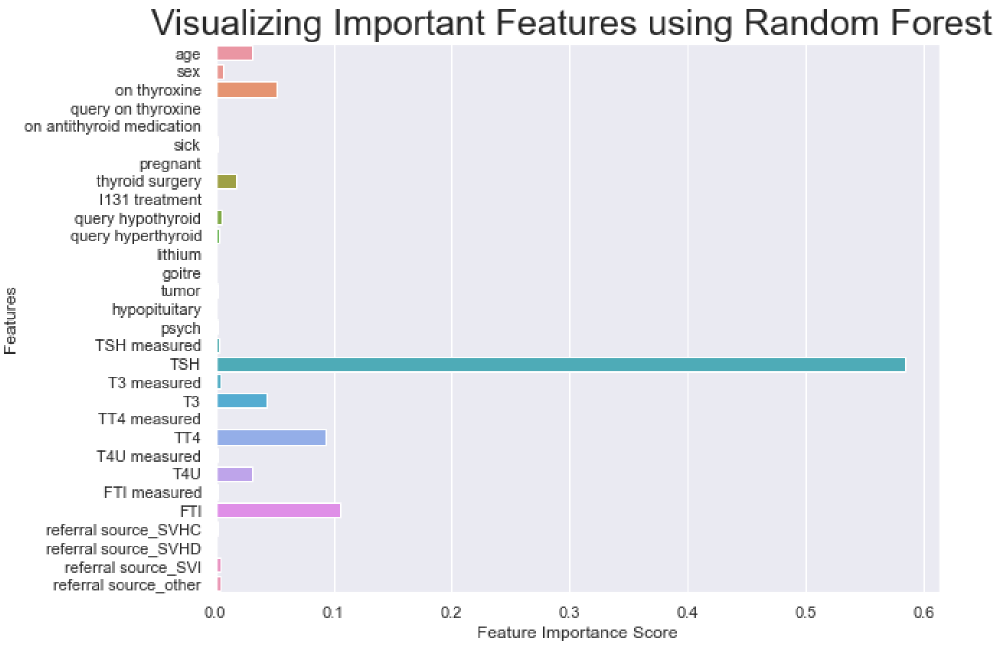
7. FEATURE SELECTION USING RANDOM FOREST



Random forests consist of 4 –12 hundred decision trees, each of them built over a random extraction of the observations from the dataset and a random extraction of the features. Not every tree sees all the features or all the observations, and this guarantees that the trees are de-correlated and therefore less prone to over-fitting. Each tree is also a sequence of yes-no questions based on a single or combination of features.

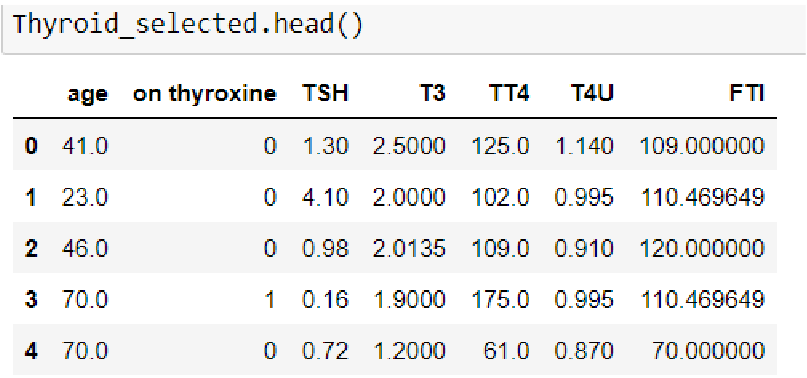
At each node (this is at each question), the tree divides the dataset into 2 buckets, each of them hosting observations that are more similar among themselves and different from the ones in the other bucket. Therefore, the importance of each feature is derived from how “pure” each of the buckets is. For classification, the measure of impurity is either the Gini impurity or the information gain/entropy.

Visualizing Important Features using Random Forest



Selecting the Most Important Features for model building.

Create A Data Subset with Only the Most Important Features



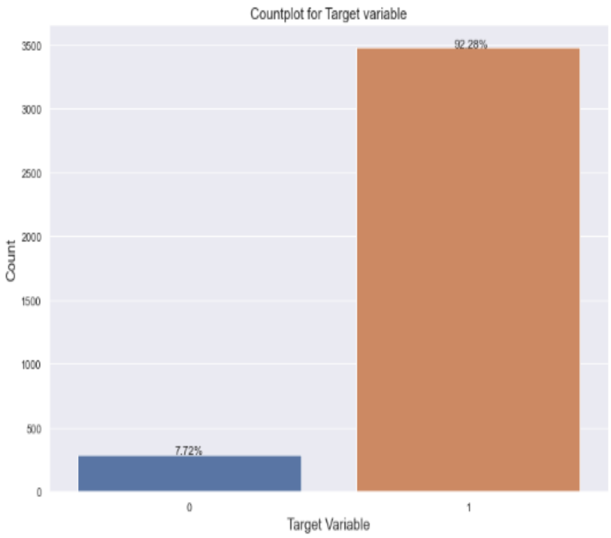
```
Thyroid_selected.head()
```

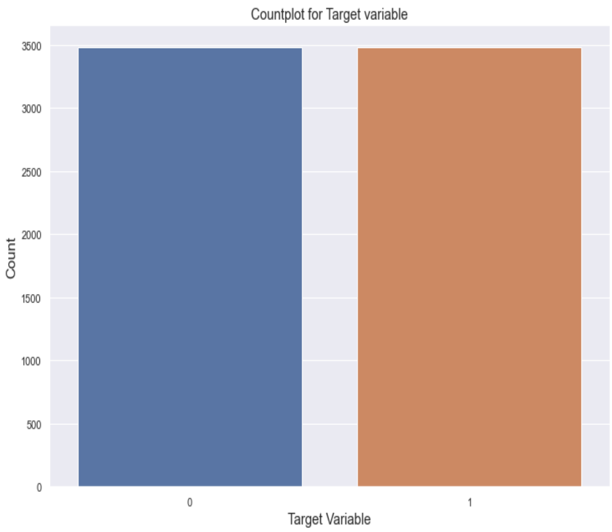
	age	on thyroxine	TSH	T3	TT4	T4U	FTI
0	41.0	0	1.30	2.5000	125.0	1.140	109.000000
1	23.0	0	4.10	2.0000	102.0	0.995	110.469649
2	46.0	0	0.98	2.0135	109.0	0.910	120.000000
3	70.0	1	0.16	1.9000	175.0	0.995	110.469649
4	70.0	0	0.72	1.2000	61.0	0.870	70.000000

8. HANDLING IMBALANCED DATA USING SMOTE

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together.

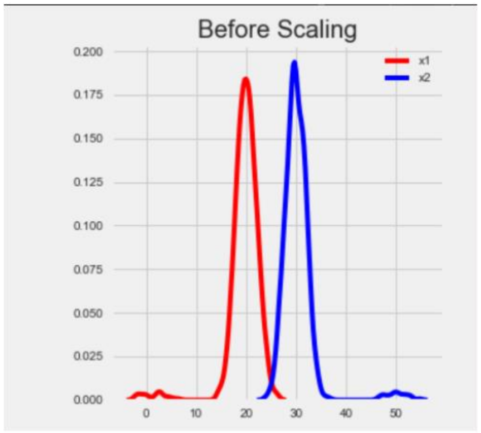
Imbalanced Target variable **Target variable after applying SMOTE**

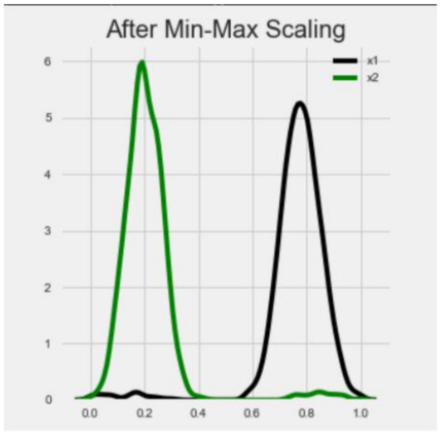




9. FEATURE SELECTION USING MINMAX SCALER

There are many ways of data scaling, where the minimum feature is made equal to zero and the maximum feature equal to one. MinMax Scaler shrinks the data within the given range, usually from 0 to 1. It transforms data by scaling features to a given range. It scales the values to a specific value range without changing the shape of the original distribution.





10. MODEL

LOGISTIC REGRESSION

Logistic regression is another powerful supervised ML algorithm used for binary classification problems (when the target is categorical). It is a predictive analytic technique that is based on the probability idea. The goal of Logistic Regression is to discover a link between characteristics and the likelihood of a specific outcome.

The Logistic Regression utilizes a more sophisticated cost function, which is known as the "Sigmoid function" or "logistic function" instead of a linear function.

K-NEAREST NEIGHBORS

K-Nearest Neighbours, or KNN for short, is one of the simplest machine learning algorithms and is used in a wide array of institutions. KNN is a non-parametric, lazy learning algorithm. When we say a technique is non-parametric, it means that it does not make any assumptions about the underlying data. Being a lazy learning algorithm implies that there is little to no training phase. Therefore, we can immediately classify new data points as they present themselves.

SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is a supervised and linear Machine Learning algorithm most commonly used for solving classification problems and is also referred to as Support Vector Classification. SVM also supports the kernel method also called the kernel SVM which allows us to tackle non-linearity.

The objective of SVM is to draw a line that best separates the two classes of data points. SVM generates a line that can cleanly separate the two classes. How clean, you may ask. There are many possible ways of drawing a line that separates the two classes, however, in SVM, it is determined by the margins and the support vectors. The SVM then generates a hyperplane that has the maximum margin. In the case of more than 2 features and multiple dimensions, the line is replaced by a hyperplane that separates multidimensional spaces.

DECISION TREE

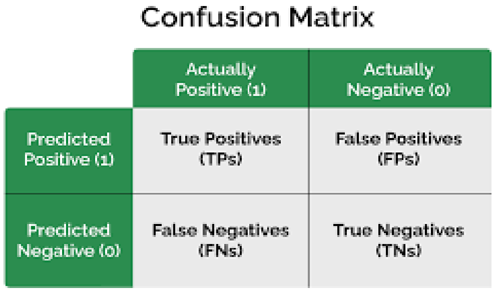
A Decision Tree is a non-parametric supervised learning method. It builds a regression model in the form of a tree structure. It breaks down a data set into smaller subsets, which is called splitting.

The final result is a tree with a decision and leaf nodes. A decision node has two or more branches. The leaf node represents a class or decision. The topmost decision node in a tree that corresponds to the best predictor is called the 'root node'. The decision tree is built using different criteria like Gini index, and entropy.

To build the decision tree, we used the criterion of 'entropy'. Entropy is one of the criteria used to build the decision tree. It calculates the homogeneity of the sample. The entropy is zero if the sample is completely homogeneous, and it is equal to 1 if the sample is equally divided.

11. MODEL EVALUATION

Confusion matrix



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

The confusion matrix is a very popular measure used while solving classification problems. It can be applied to binary classification as well as for multiclass classification problems.

TP (True Positive):

TP represents the number of patients who have been properly classified to have thyroid disease.

TN (True Negative):

TN represents the number of correctly classified patients who are healthy.

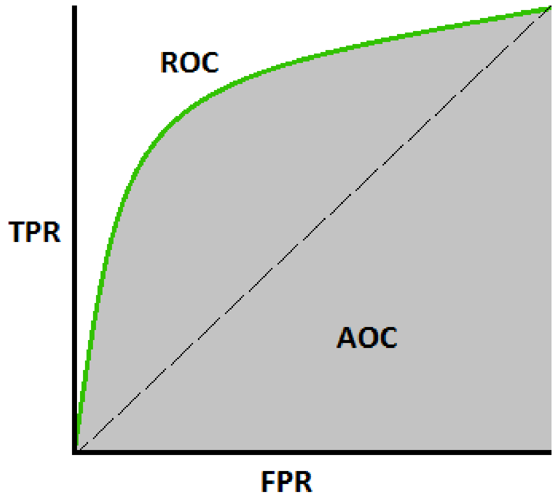
FP (False Positive):

FP represents the number of misclassified patients with the disease but who are healthy. FP is also known as a *Type I error*.

FN (False Negative):

FN represents the number of patients misclassified as healthy but who are suffering from the disease. FN is also known as a *Type II error*.

AUC - ROC curve:



AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

Performance metrics of an algorithm are accuracy, precision, recall, and F1 score, which are calculated based on the above-stated TP, TN, FP, and FN.

Accuracy:

The accuracy of an algorithm is represented as the ratio of correctly classified patients (TP+TN) to the total number of patients (TP+TN+FP+FN).

Accuracy=(TP+TN) / (TP+FP+FN+TN)

Precision:

The precision of an algorithm is represented as the ratio of correctly classified patients with the disease (*TP*) to the total patients predicted to have the disease (*TP*+*FP*).

Precision=TP / (TP+FP)

Recall:

The metric is defined as the ratio of correctly classified diseased patients (*TP*) divided by the total number of patients who have the disease.

Recall=TP / (TP+FN)

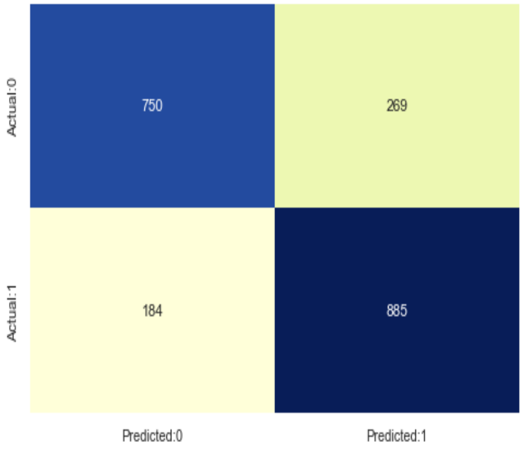
The perception behind the recall is how many patients have been classified as having the disease. The recall is also called sensitivity.

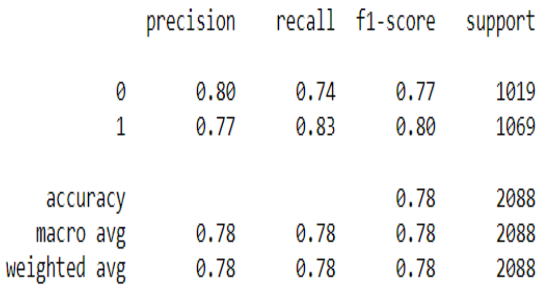
F1 score:

The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

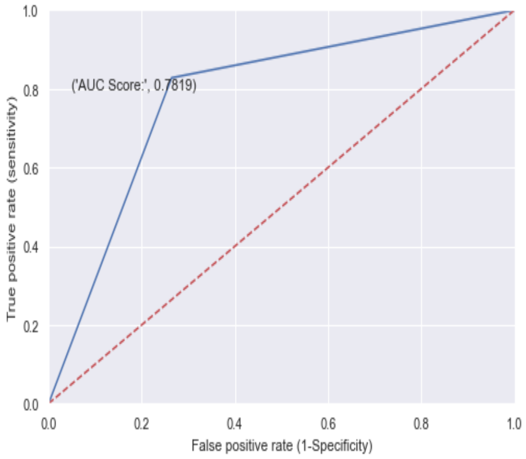
Comparing Model Results:

Logistic Regression

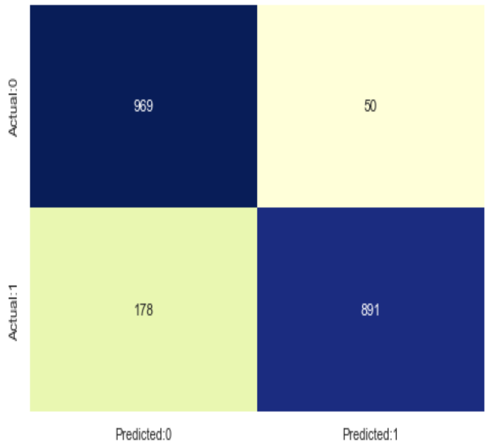


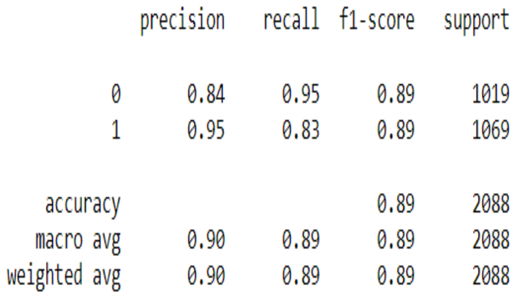


```
              precision    recall  f1-score   support

           0       0.80      0.74      0.77      1019
           1       0.77      0.83      0.80      1069

    accuracy                           0.78      2088
   macro avg       0.78      0.78      0.78      2088
weighted avg       0.78      0.78      0.78      2088
```

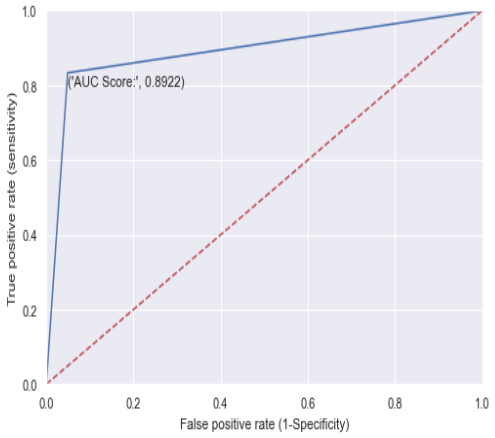


K-Nearest Neighbour

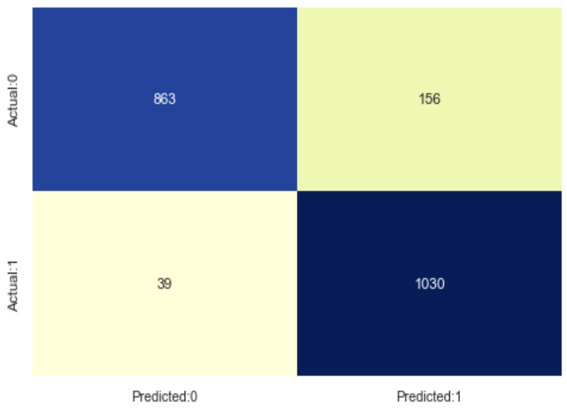


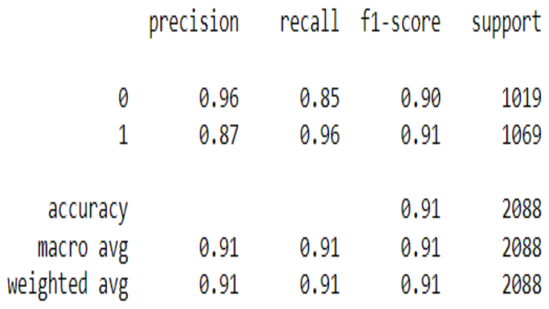


```
              precision    recall  f1-score   support

           0       0.84      0.95      0.89      1019
           1       0.95      0.83      0.89      1069

    accuracy                           0.89      2088
   macro avg       0.90      0.89      0.89      2088
weighted avg       0.90      0.89      0.89      2088
```

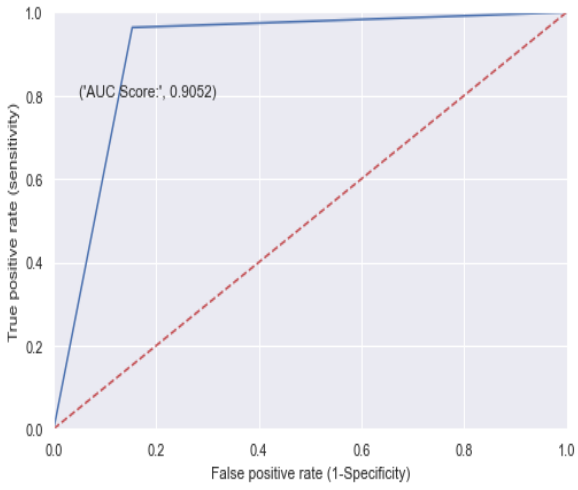


Support Vector Machine

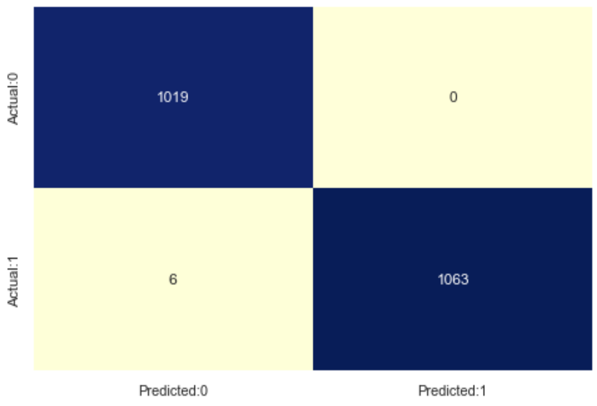


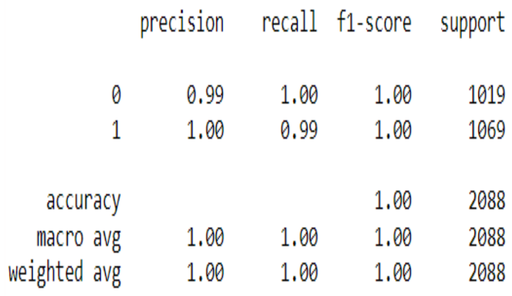


```
              precision    recall  f1-score   support

           0       0.96      0.85      0.90      1019
           1       0.87      0.96      0.91      1069

    accuracy                           0.91      2088
   macro avg       0.91      0.91      0.91      2088
weighted avg       0.91      0.91      0.91      2088
```

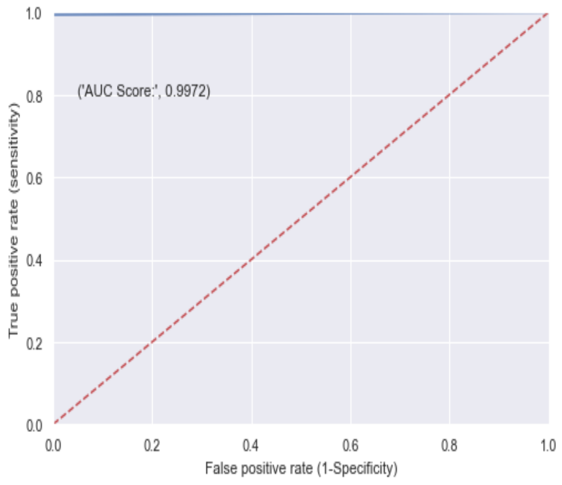


Decision Tree

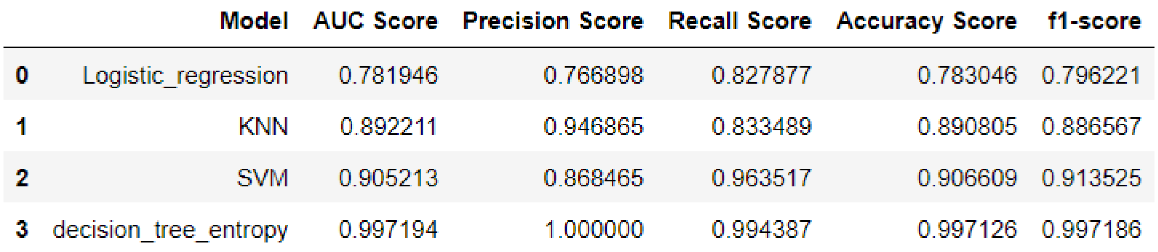




```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      1019
           1       1.00      0.99      1.00      1069

    accuracy                           1.00      2088
   macro avg       1.00      1.00      1.00      2088
weighted avg       1.00      1.00      1.00      2088
```



12. RESULT TABULATION:



	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_regression	0.781946	0.766898	0.827877	0.783046	0.796221
1	KNN	0.892211	0.946865	0.833489	0.890805	0.886567
2	SVM	0.905213	0.868465	0.963517	0.906609	0.913525
3	decision_tree_entropy	0.997194	1.000000	0.994387	0.997126	0.997186

13. Conclusion:

The supervised classification learning algorithms named in the above table have been implemented on the given dataset. The performance of the models was evaluated using the AUC score, accuracy, precision, recall, and f1-score.

The above table shows that the **Decision Tree** has the highest values for most of the performance measures like AUC Score, Recall, f1-score, and accuracy. Therefore, it can be concluded that the Decision Tree can be used to predict the existence of Hypothyroidism in patients.

14. REFERENCE

1. https://en.wikipedia.org (Theoretical information)

2. https://scikit-learn.org (Python packages and code help)

3. https://www.downtoearth.org.in (Thyroid information)

4. https://www.ncbi.nlm.nih.gov (Thyroid disease understanding)

5. https://towardsdatascience.com (Theory and visualization)

6. https://seaborn.pydata.org (Data visualization)

7. https://www.geeksforgeeks.org (Theoretical information)

8. https://www.analyticsvidhya.com (Theoretical information)

9. https://images.google.com (concept related images)