

Thyroid Disease Prediction Using Machine Learning

line 1: 2nd Given Name Surname
line 2: dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

Abstract— In the field of healthcare and medicine, machine learning and artificial intelligence have an important result in diagnosing diseases such as thyroid, cancer, heart, and many more as well. In our research analysis, we collect thyroid data to predict that someone has hyperthyroidism, hypothyroidism, or normal. This data contains all types of patient dataset which comes under the above 3 categories. Thyroid disease can occur in anyone, it may be, men, women, or children. It is detected by the blood test but someone it is not possible to detect properly that's why machine learning techniques are going to be implemented to find better results for the treatment of thyroid. According to the analysis, the random forest algorithm performs better and is outstanding for the thyroid treatment.

I. INTRODUCTION

Thyroid disease is a part of endocrinology which is the most common undiagnosed disease according to the World Health Organization. Men have 10% more prevalence than women. This disease is caused by thyroid gland dysfunction. This disease is also affected by pituitary gland failure and malfunctioning of the hypothalamic. Sometimes it is also due to iodine deficiency, goiter, or thyroid nodules. To take diagnosis and care on time are very important otherwise it will cause even death. After various trials, diagnosis is still difficult for doctors. The throat gland looks like a butterfly-shaped gland which is present at the base of the throat. It contains 2 hormones levothyroxine and triiodothyronine which contain functionality of the brain like as body temperature control, heart rate, and blood pressure. These glands are also responsible for digestion and maintaining the moisture of the body. It provides some treatments such as T3, T4, and TSH which stimulate the thyroid hormones.

Machine learning techniques are one of the best solution providers in the field of healthcare. In current days, machine learning is trying to provide the best solutions to the healthcare system, and in the future also, it will predict better results as well so that they can believe blindly in artificial intelligence as well in the upcoming days. To perform the thyroid analysis, it uses the classification technique which predicts whether someone has thyroid or not based on the given symptoms.

Hyperthyroidism is a disorder where the thyroid gland releases the maximum amount of thyroid hormones. It is caused by the increment of hormone levels. Some symptoms of hyperthyroidism are heavy sweating, hair thinning, dry skin, high blood pressure, weight loss, neck enlargement, irregular stomach movements, temperature sensitivity, handshaking, and menstrual cycle shortening are the common signs and of this condition, the thyroid gland is underactive. On the other hand,

hypothyroidism occurs in the body when it declines to produce the thyroid hormones. The main cause of this disorder is thyroid gland injury and inflammation. Some symptoms of the disorder are neck swelling, hand numbness, low heart rate, obesity, temperature sensitivity, hair issues, intestinal problems, dry skin, and heavy menstrual cycles are the common problem. If we will not treat on time then it will directly affect the body and be responsible for mortality.

II. DATA ANALYSIS

A. Data Information:

Understanding the data information is very important for data analysis. To predict thyroid diagnoses, machine learning is providing better results in the healthcare system. We have read the thyroid dataset using the read function which is collected from the Kaggle which is provided by the UCI machine learning repository. This dataset contains 9172 observations and 31 features.

Data Collection

```
In [85]: #Read Thyroid dataset
import pandas as pd
data = pd.read_csv('thyroidDF.csv')
data.head()
```

Out[85]:

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surge
0	29	F	f		f	f	f	f
1	29	F	f		f	f	f	f
2	41	F	f		f	f	f	f
3	38	F	f		f	f	f	f
4	32	F	f		f	f	f	f

5 rows x 31 columns

Figure 1: Data Collection

```
In [86]: #Check dimension
data.shape
```

Out[86]: (9172, 31)

Figure 2: Data Dimension

When we collect data, we don't know which type of data it is. It is in a structured format but other than this we don't know anything initially. We use the info function to understand the deep knowledge of the dataset.

```
In [87]: #information of data
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9172 entries, 0 to 9171
Data columns (total 31 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   age                    9172 non-null   int64
1   sex                    8865 non-null   object
2   on_thyroxine            9172 non-null   object
3   query_on_thyroxine      9172 non-null   object
4   on_antithyroid_meds     9172 non-null   object
5   sick                    9172 non-null   object
6   pregnant                9172 non-null   object
7   thyroid_surgery         9172 non-null   object
8   I131_treatment          9172 non-null   object
9   query_hypothyroid       9172 non-null   object
10  query_hyperthyroid      9172 non-null   object
11  lithium                 9172 non-null   object
12  goitre                  9172 non-null   object
13  tumor                   9172 non-null   object
14  hypopituitary           9172 non-null   object
15  psych                   9172 non-null   object
16  TSH_measured            9172 non-null   object
17  TSH                      8330 non-null   float64
18  T3_measured             9172 non-null   object
19  T3                       6568 non-null   float64
20  TT4_measured            9172 non-null   object
21  TT4                      8730 non-null   float64
22  T4U_measured            9172 non-null   object
23  T4U                      8363 non-null   float64
24  FTI_measured            9172 non-null   object
25  FTI                      8370 non-null   float64
26  TBG_measured            9172 non-null   object
27  TBG                      349 non-null    float64
28  referral_source         9172 non-null   object
29  target                   9172 non-null   object
30  patient_id              9172 non-null   int64
dtypes: float64(6), int64(2), object(23)
memory usage: 2.2+ MB
```

Figure 3: Data Information

The describe function is a very important function that provides the data summary by implementing it on numerical columns.

```
In [88]: #description of data
data.describe()

Out[88]:
```

	age	TSH	T3	TT4	T4U	FTI	TBG
count	9172.000000	8330.000000	6568.000000	8730.000000	8363.000000	8370.000000	349.000000
mean	73.556822	5.218403	1.970829	108.700305	0.978056	113.640748	29.870057
std	1183.978718	24.184008	0.887579	37.522670	0.200380	41.551650	21.080504
min	1.000000	0.005000	0.050000	2.000000	0.170000	1.400000	0.100000
25%	37.000000	0.480000	1.500000	87.000000	0.880000	93.000000	21.000000
50%	55.000000	1.400000	1.900000	104.000000	0.980000	109.000000	26.000000
75%	68.000000	2.700000	2.300000	128.000000	1.086000	128.000000	31.000000
max	85628.000000	530.000000	18.000000	600.000000	2.330000	881.000000	200.000000

Figure 4: Description of Data

Knowing the unique value is very important. It provides which type of data it is, which type of values it is storing, and its count.

```
In [59]: #check unique values in all columns
for i in data.columns:
    print(f"{i} ----- {data[i].nunique()}")

age ----- 100
sex ----- 2
on_thyroxine ----- 2
query_on_thyroxine ----- 2
on_antithyroid_meds ----- 2
sick ----- 2
pregnant ----- 2
thyroid_surgery ----- 2
I131_treatment ----- 2
query_hypothyroid ----- 2
query_hyperthyroid ----- 2
lithium ----- 2
goitre ----- 2
tumor ----- 2
hypopituitary ----- 2
psych ----- 2
TSH_measured ----- 2
TSH ----- 369
T3_measured ----- 2
T3 ----- 85
TT4_measured ----- 2
TT4 ----- 287
T4U_measured ----- 2
T4U ----- 176
FTI_measured ----- 2
FTI ----- 323
TBG_measured ----- 2
TBG ----- 66
referral_source ----- 6
target ----- 32
patient_id ----- 9172
```

Figure 5: Data Uniqueness

There are 31 attributes present such as 'age', 'sex', 'on_thyroxine', 'query_on_thyroxine', 'on_antithyroid_meds', 'sick', 'pregnant', 'thyroid_surgery', 'I131_treatment', 'query_hypothyroid', 'query_hyperthyroid', 'lithium', 'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH_measured', 'TSH', 'T3_measured', 'T3', 'TT4_measured', 'TT4', 'T4U_measured', 'T4U', 'FTI_measured', 'FTI', 'TBG_measured', 'TBG', 'referral_source', 'target', 'patient_id'.

```
In [90]: data.columns

Out[90]: Index(['age', 'sex', 'on_thyroxine', 'query_on_thyroxine',
               'on_antithyroid_meds', 'sick', 'pregnant', 'thyroid_surgery',
               'I131_treatment', 'query_hypothyroid', 'query_hyperthyroid', 'lithium',
               'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH_measured', 'TSH',
               'T3_measured', 'T3', 'TT4_measured', 'TT4', 'T4U_measured', 'T4U',
               'FTI_measured', 'FTI', 'TBG_measured', 'TBG', 'referral_source',
               'target', 'patient_id'],
              dtype='object')
```

Figure 6: Data Columns

B. Data Cleaning and Preprocessing:

Data cleaning and preprocessing is a very important part of cleaning the system and data. We have checked the missing values and some columns contain a high amount of missing data but there is no duplicate data present.

```
In [60]: #check missing values
data.isnull().sum()
```

```
Out[60]: age                0
sex                307
on_thyroxine       0
query_on_thyroxine 0
on_antithyroid_meds 0
sick               0
pregnant          0
thyroid_surgery    0
t13t1_treatment    0
query_hypothyroid  0
query_hyperthyroid 0
lithium           0
goitre            0
tumor             0
hypopituitary     0
psych             0
TSH_measured      0
TSH               842
T3_measured       0
T3                2604
TT4_measured      0
TT4               442
T4U_measured      0
T4U              809
FTI_measured      0
FTI               802
TBG_measured      0
TBG              8823
referral_source   0
target            0
patient_id        0
dtype: int64
```

```
In [61]: #check duplicate data
data.duplicated().sum()
```

```
Out[61]: 0
```

Figure 7: Check Missing and Duplicate Values

We have dropped the columns to perform the data cleaning because those columns contain a high amount of data.

```
In [62]: #perform data cleaning
data.drop(['T3_measured', 'TSH_measured', 'T4U_measured', 'TT4_measured',
          'FTI_measured', 'TBG_measured',
          'referral_source', 'patient_id'], axis=1, inplace=True)
data.head()
```

```
Out[62]:
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surge
0	29	F	f	f	f	f	f	f
1	29	F	f	f	f	f	f	f
2	41	F	f	f	f	f	f	f
3	38	F	f	f	f	f	f	f
4	32	F	f	f	f	f	f	f

5 rows × 9 columns

Figure 8: Data Cleaning

We have assigned some values according to the target column values. We assign negative, hyperthyroid, and hypothyroid according to the target values [A, B, C, D, E, F, G, and H].

```
In [63]: #assign values
diag = {'-': 'negative', 'A': 'hyperthyroid', 'B': 'hyperthyroid',
        'C': 'hyperthyroid', 'D': 'hyperthyroid', 'E': 'hypothyroid',
        'F': 'hypothyroid', 'G': 'hypothyroid', 'H': 'hypothyroid'}

data['target'] = data['target'].map(diag) # re-mapping
data.dropna(subset=['target'], inplace=True)
data.shape
```

```
Out[63]: (7546, 23)
```

Figure 9: Transform Target Variable

This dataset contains some data whose age is greater than 100 so we have deleted those records from the data otherwise it will come under the outlier.

```
In [65]: #drop rows where patient age greater than 100 as it comes under outliers.
data = data.drop(data[data['age'] > 100].index)
data = data.drop('TBG', axis=1)
data.shape
```

```
Out[65]: (7542, 22)
```

Figure 10: Remove Outliers

There are some columns which need to be filled with some value. We have performed the mean imputation on those columns.

```
In [66]: #fill missing values with mean value
data['TSH'].fillna((data['TSH'].mean()), inplace=True)
data['T3'].fillna((data['T3'].mean()), inplace=True)
data['TT4'].fillna((data['TT4'].mean()), inplace=True)
data['T4U'].fillna((data['T4U'].mean()), inplace=True)
data['FTI'].fillna((data['FTI'].mean()), inplace=True)
```

Figure 11: Mean Imputation

We finally dropped missing rows as well from the data.

```
In [67]: #drop all other rows with missing values
data.dropna(inplace=True)
```

Figure 12: Drop missing values

Label Encoding is also implemented to transform the data into numerical format.

Data Preprocessing

```
In [68]: from sklearn import preprocessing
data1=data.apply(preprocessing.LabelEncoder().fit_transform)
```

Figure 13: Data Preprocessing

C. Data Visualization:

Data Visualization is a task to display charts and graphs on various attributes. On the Thyroid dataset, we have performed the visualization to diagnose the proper treatment. We have plotted the count plot using Seaborn to show the count of each value on the thyroid surgery column.

```
In [77]: sns.countplot(x='thyroid_surgery', data = data)
plt.show()
```

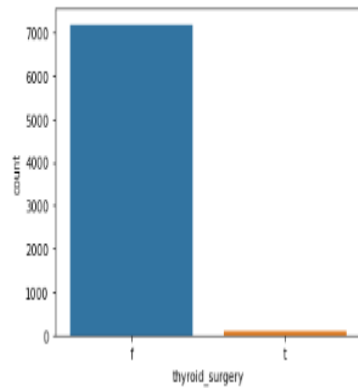


Figure 14: Count of Thyroid Surgery

We have plotted the count plot using Seaborn to show the count of each value on the sex column.

```
In [79]: sns.countplot(x='sex', data = data)
plt.show()
```

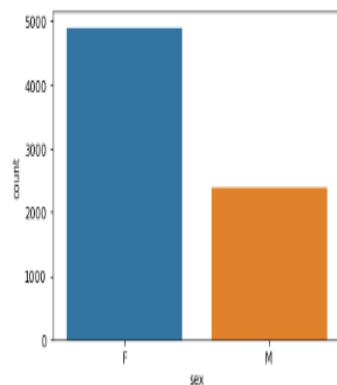


Figure 15: Count of Sex

We have plotted the count plot using Seaborn to show the count of each value on the hyperthyroid column.

```
In [80]: sns.countplot(x='query_hyperthyroid', data = data)
plt.show()
```

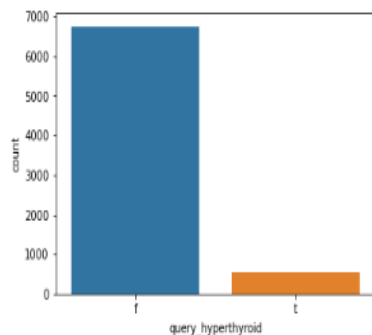


Figure 16: Count of Query Hyperthyroid

We have plotted the count plot using Seaborn to show the count of each value on the hypothyroid column.

```
In [81]: sns.countplot(x='query_hypothyroid', data = data)
plt.show()
```

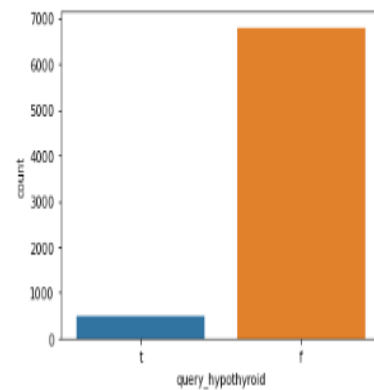


Figure 17: Count of Query Hypothyroid

We have plotted the count plot using Seaborn to show the count of each value on the target column.

```
In [76]: sns.countplot(x='target', data = data)
plt.show()
```

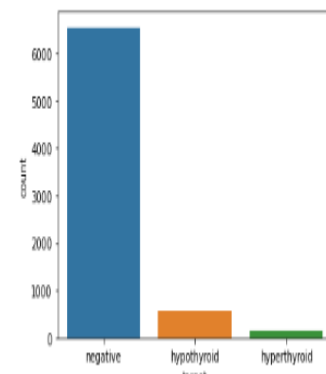


Figure 18: Count of Target Variable

The correlation matrix is a matrix that provides the correlation between the features and we represent it through a heat map.

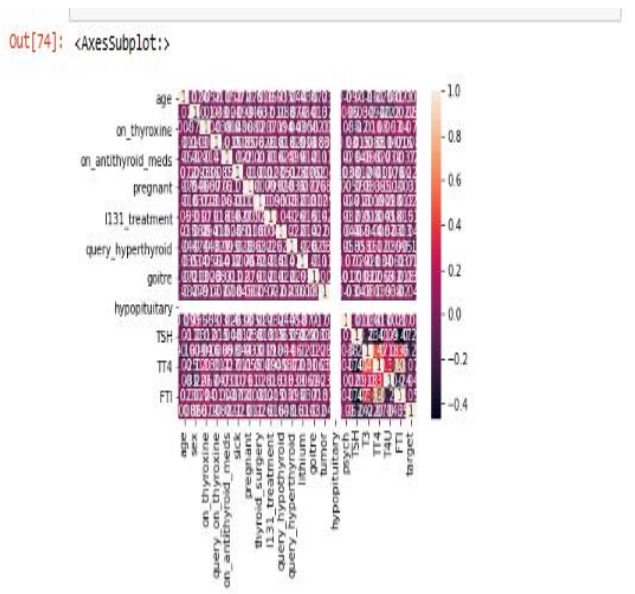


Figure 19: Heat Map

The line chart is a graph that helps to show the variation though in between the 2 features.

We have graphed the relationship between thyroid target and age through a line chart.

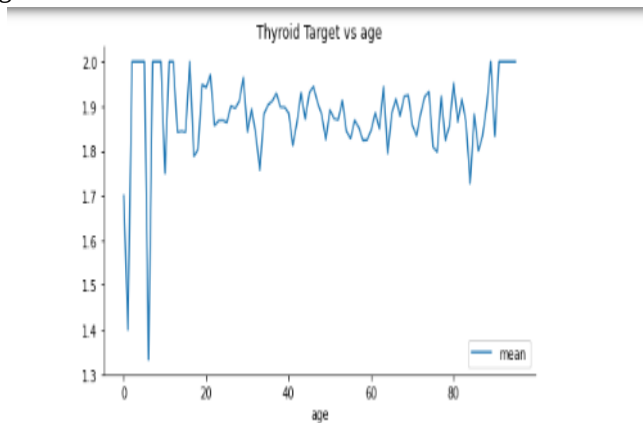


Figure 20: Figure 1: Line Chart between Thyroid Target and Age

We have graphed the relationship between thyroid target and sex through a line chart.

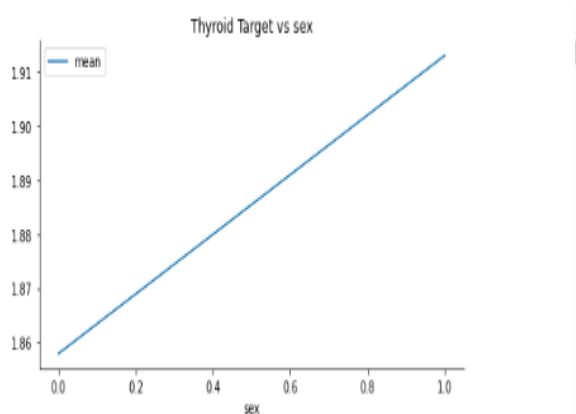


Figure 21: Figure 1: Line Chart between Thyroid Target and Sex

We have graphed the relationship between thyroid target and thyroxine through a line chart.

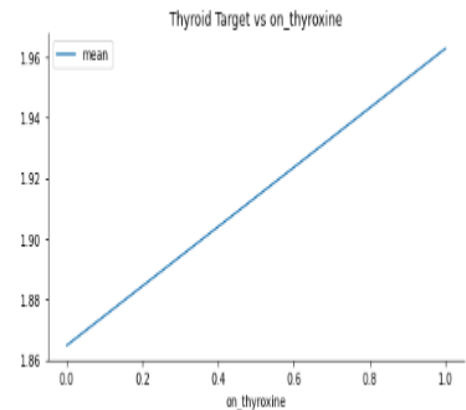


Figure 22: Figure 1: Line Chart between Thyroid Target and Thyroxine

We have graphed the relationship between thyroid target and query on thyroxine through a line chart.

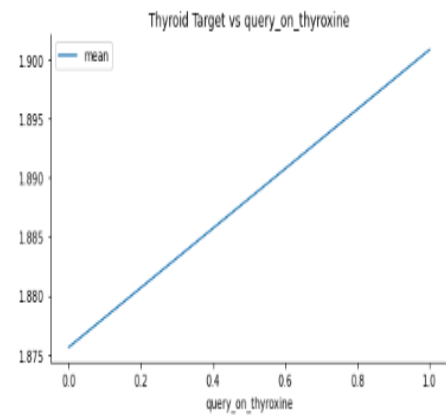


Figure 23: Figure 1: Line Chart between Thyroid Target and Query on thyroxine

We have graphed the relationship between thyroid targets and antithyroid meds through a line chart.

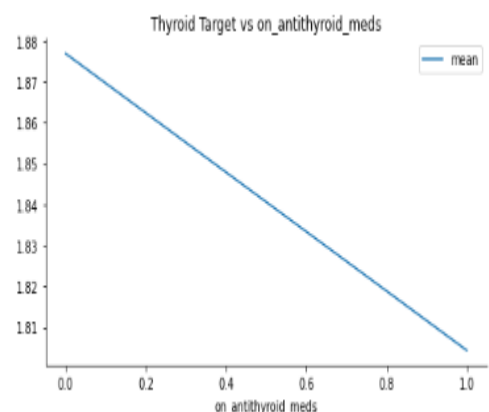


Figure 24: Figure 1: Line Chart between Thyroid Target and Antithyroid meds

We have graphed the relationship between thyroid target and sickness through a line chart.

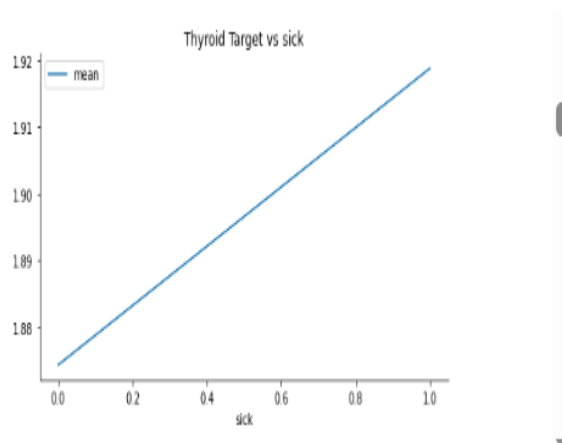


Figure 25: Figure 1: Line Chart between Thyroid Target and Sick

We have graphed the relationship between thyroid target and pregnancy through a line chart.

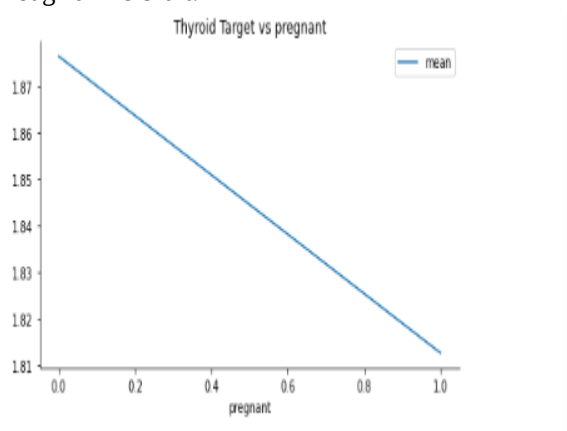


Figure 26: Figure 1: Line Chart between Thyroid Target and Pregnant

We have graphed the relationship between thyroid target and thyroid surgery through a line chart.

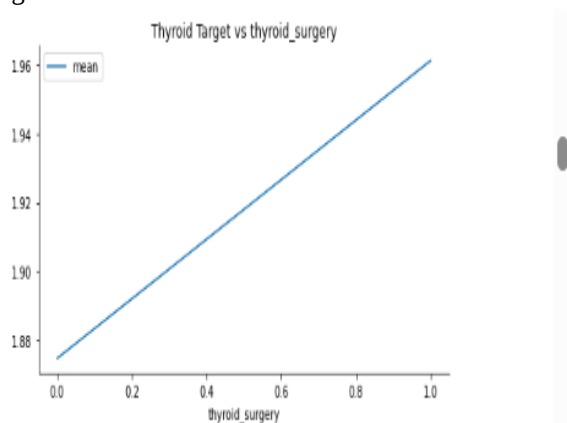


Figure 27: Figure 1: Line Chart between Thyroid Target and Thyroid Surgery

We have graphed the relationship between thyroid target and I131 treatment through a line chart.

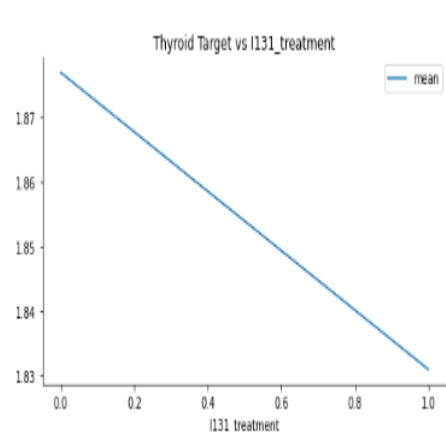


Figure 28: Figure 1: Line Chart between Thyroid Target and I131 Treatment

We have graphed the relationship between thyroid target and query hypothyroid through a line chart.

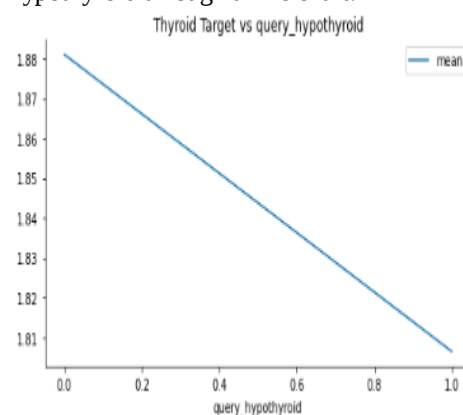


Figure 29: Figure 1: Line Chart between Thyroid Target and Query Hypothyroid

We have graphed the relationship between thyroid target and query hyperthyroid through a line chart.

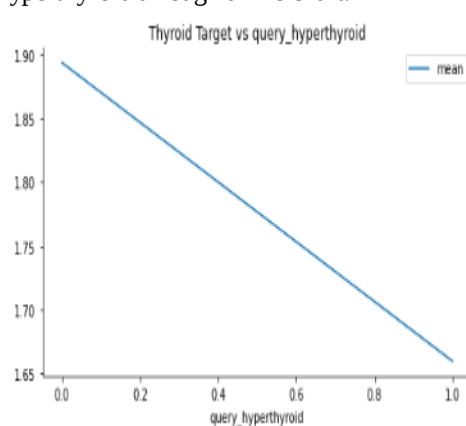


Figure 30: Line Chart between Thyroid Target and Query Hyperthyroid

We have graphed the relationship between thyroid target and lithium through a line chart.

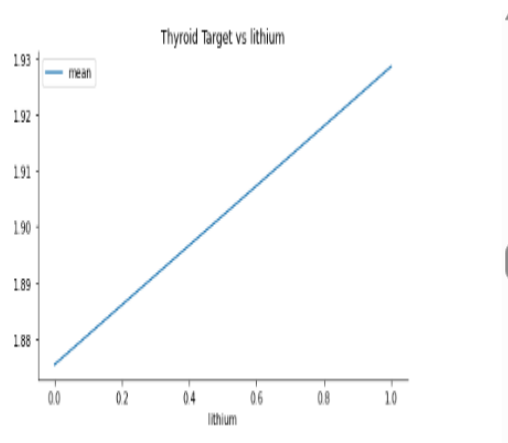


Figure 31: Line Chart between Thyroid Target and Lithium

We have graphed the relationship between thyroid target and goiter through a line chart.

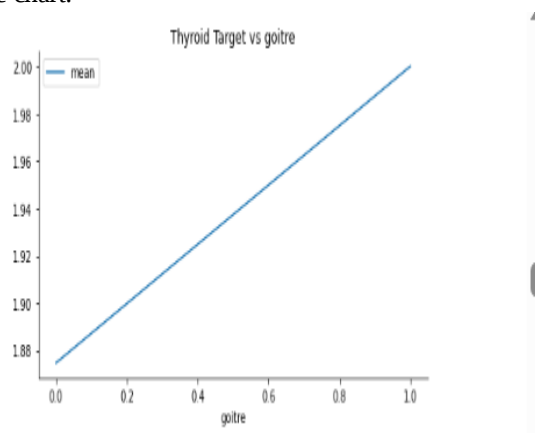


Figure 32: Line Chart between Thyroid Target and Gottle

We have graphed the relationship between thyroid target and tumor through a line chart.

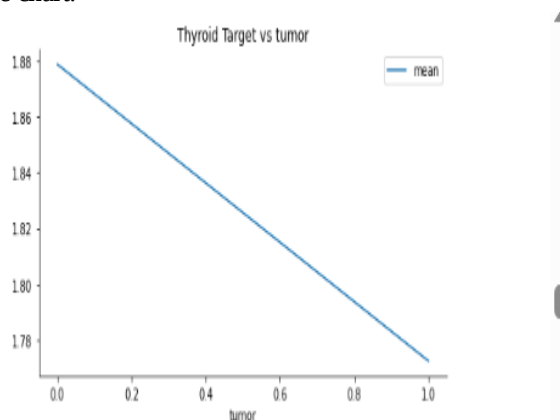


Figure 33: Line Chart between Thyroid Target and Tumor

We have graphed the relationship between thyroid target and hypopituitary through a line chart.

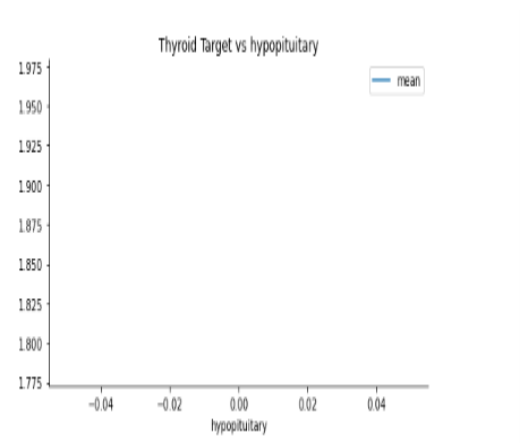


Figure 34: Line Chart between Thyroid Target and Hypopituitary

We have graphed the relationship between thyroid target and psych through a line chart.

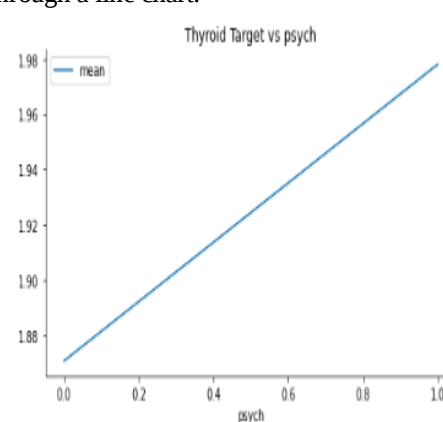


Figure 35: Line Chart between Thyroid Target and Psych

We have graphed the relationship between thyroid target and TSH through a line chart.

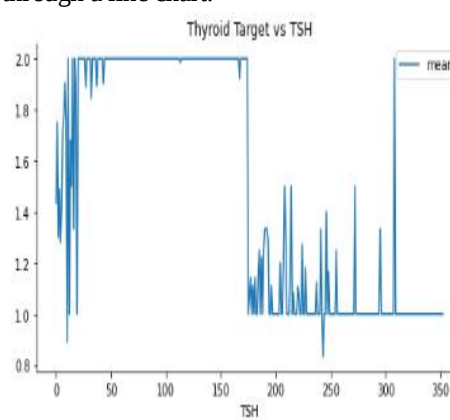


Figure 36: Line Chart between Thyroid Target and TSH

We have graphed the relationship between the thyroid target and T3 through a line chart.

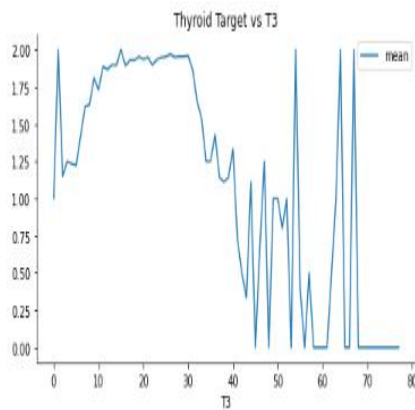


Figure 37: Line Chart between Thyroid Target and T3

We have graphed the relationship between thyroid target and TT4 through a line chart.

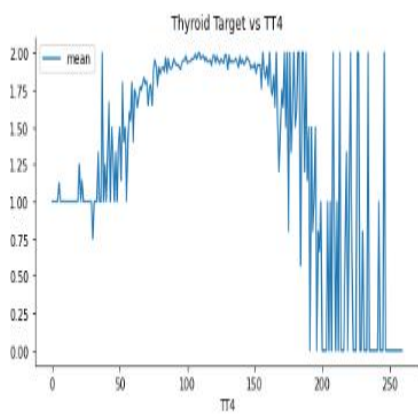


Figure 38: Line Chart between Thyroid Target and TT4

We have graphed the relationship between the thyroid target and T4U through a line chart.

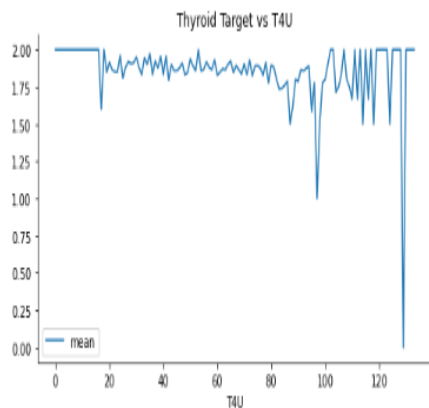


Figure 39: Line Chart between Thyroid Target and T4U

We have graphed the relationship between thyroid target and FTI through a line chart.

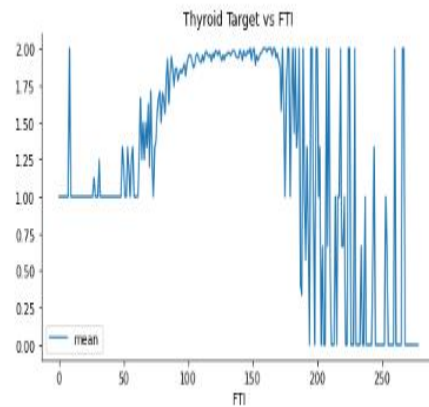


Figure 40: Line Chart between Thyroid Target and FTI

III. MACHINE LEARNING TECHNIQUES

To implement the techniques, the target is considered a dependent variable. The other 23 features are considered as independent variables. We assign 20% of the data for testing purposes.

A. Support Vector Machine:

Support Vector is also known as a discriminative classifier. It uses the kernel function to handle nonlinear relationships efficiently. It is less interpretable. It is sensitive when we need to select the hyperparameters. It is expensive for large datasets and sometimes not scalable for huge datasets.

Support Vector Machine

```
In [39]: from sklearn.svm import SVC
classifier1 = SVC()
classifier1.fit(x_train, y_train)
y_pred1 = classifier1.predict(x_test)

In [40]: from sklearn.metrics import confusion_matrix, accuracy_score
cm1 = confusion_matrix(y_test, y_pred1)
acc1 = accuracy_score(y_test, y_pred1)
print("Accuracy Score:- ", acc1*100)
print("Confusion Matrix : \n", cm1)

Accuracy Score:- 96.91569568197396
Confusion Matrix :
[[ 20  0 20]
 [ 0 90 17]
 [ 2  6 1304]]

In [42]: classifier1.score(x_test, y_test)

Out[42]: 0.9691569568197396
```

Figure 41: Performance of Support Vector Machine

B. Random Forest:

Random forest is a supervised model and familiar for ensemble learning which combines the weak learner and converts into a strong learner. It can handle the non-linear relationships as well. It contains multiple decision trees. It is also famous for its high

performance and less prone to overfitting. But it is expensive if we train multiple decision trees.

Random Forest

```
In [48]: from sklearn.ensemble import RandomForestClassifier
classifier3= RandomForestClassifier()
classifier3.fit(x_train , y_train)
y_pred3 = classifier3.predict(x_test)

In [49]: from sklearn.metrics import confusion_matrix,accuracy_score
cm3 = confusion_matrix(y_test, y_pred3)
acc3=accuracy_score(y_test,y_pred3)
print("Accuracy Score:- ",(acc3*100))
print ("Confusion Matrix : \n", cm3)

Accuracy Score:- 99.04043865661411
Confusion Matrix :
[[ 31  0  9]
 [ 0 107  0]
 [ 3  2 1307]]

In [50]: classifier3.score(x_test, y_test)

Out[50]: 0.9904043865661412
```

Figure 42: Performance of Random Forest

C. Logistic Regression:

Logistic regression is a linear model which performs binary and multi-classification. It provides a linear relationship between the variables. To predict the relationship, a coefficient is helpful for this. It is famous for its good performance and robustness. It is less prone to overfitting. It is scalable for large datasets. We only need to tune very less hyperparameters.

Logistic Regression

```
In [51]: from sklearn.linear_model import LogisticRegression
classifier2 = LogisticRegression(random_state = 0)
classifier2.fit(x_train, y_train)
y_pred2 = classifier2.predict(x_test)

C:\Users\SR\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
on
n_iter_i = _check_optimize_result(

In [52]: from sklearn.metrics import confusion_matrix,accuracy_score
cm2 = confusion_matrix(y_test, y_pred2)
acc2 = accuracy_score(y_test,y_pred2)
print("Accuracy Score:- ",(acc2*100))
print ("Confusion Matrix : \n", cm2)

Accuracy Score:- 96.36737491432487
Confusion Matrix :
[[ 20  0 20]
 [ 0 85 22]
 [ 2  9 1301]]

In [53]: classifier2.score(x_test, y_test)

Out[53]: 0.9636737491432488
```

Figure 43: Performance of Logistic Regression

IV. CONCLUSION

Thyroid disease is affecting the population of the world and the death rate is also increasing. According to the medical reports. Some imbalances are there to predict hypothyroidism and hyperthyroidism that's why machine learning techniques are implemented to perform the prediction and compare the accuracy score for the implementation in real-world healthcare. According to the research analysis on the thyroid dataset, we identified that the accuracy score of random forest is 99%. The accuracy score of logistic regression is 96%. The accuracy score of the support vector machine is 96%. So we can say that the random forest model provides better accurate results and helps in decision making.

V. FUTURE SCOPE

In the future, we can use k-fold cross-validation for superior performance. We can also recommend the use of the neural network over machine learning techniques to predict thyroid disease for early-stage recovery. In the preprocessing steps, we can use the SMOTE method to perform oversampling. The main drawback is we use a small dataset so in the future, we have to train large datasets for better effective results. One of the aspects we have to focus on is to predict hypothyroidism.

REFERENCES

1. Chaganti, R., Rustam, F., De La Torre Díez, I., Mazón, J.L.V., Rodríguez, C.L. and Ashraf, I., 2022. Thyroid disease prediction using selective features and machine learning techniques. *Cancers*, 14(16), p.3914.
2. Chaubey, G., Bisen, D., Arjaria, S. and Yadav, V., 2021. Thyroid disease prediction using machine learning approaches. *National Academy Science Letters*, 44(3), pp.233-238.
3. Aversano, L., Bernardi, M.L., Cimitile, M., Iammarino, M., Macchia, P.E., Nettore, I.C. and Verdone, C., 2021. Thyroid disease treatment prediction with machine learning approaches. *Procedia Computer Science*, 192, pp.1031-1040.
4. Sonuç, E., 2021, July. Thyroid disease classification using machine learning algorithms. In *Journal of Physics: Conference Series* (Vol. 1963, No. 1, p. 012140). IOP Publishing.
5. Guleria, K., Sharma, S., Kumar, S. and Tiwari, S., 2022. Early prediction of hypothyroidism and multiclass classification using predictive machine learning and deep learning. *Measurement: Sensors*, 24, p.100482.
6. Razia, S., Siva Kumar, P. and Rao, A.S., 2020. Machine learning techniques for thyroid disease diagnosis: a systematic review. *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI*, pp.203-212.
7. Lee, K.S. and Park, H., 2022. Machine learning on thyroid disease: a review. *Frontiers in Bioscience-Landmark*, 27(3), p.101.
8. Liu, W., Wang, S., Ye, Z., Xu, P., Xia, X. and Guo, M., 2022. Prediction of lung metastases in thyroid cancer using machine learning based on SEER database. *Cancer Medicine*, 11(12), pp.2503-2515.
9. Alyas, T., Hamid, M., Alissa, K., Faiz, T., Tabassum, N. and Ahmad, A., 2022. Empirical method for thyroid disease classification using a machine learning approach. *BioMed Research International*, 2022.
10. Asif, M.A.A.R., Nishat, M.M., Faisal, F., Shikder, M.F., Udoy, M.H., Dip, R.R. and Ahsan, R., 2020, December. Computer aided diagnosis of thyroid disease using machine learning algorithms. In *2020 11th International Conference on Electrical and Computer Engineering (ICECE)* (pp. 222-225). IEEE.
11. Hu, M., Asami, C., Iwakura, H., Nakajima, Y., Sema, R., Kikuchi, T., Miyata, T., Sakamaki, K., Kudo, T., Yamada, M. and Akamizu, T., 2022. Development and preliminary validation of a machine learning system for thyroid dysfunction diagnosis based on routine laboratory tests. *Communications Medicine*, 2(1), p.9.