Thyroid Disease Detection

A B. Tech Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

by

Aman Gupta
(1801CS05)

under the guidance of

Prof. Somanath Tripathy

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY PATNA
PATNA - 800013, BIHAR

CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**Thyroid Disease Detection**" is a bonafide work of **Aman Gupta** (**Roll No. 1801CS05**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Patna under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Prof. Somanath Tripathy**

Assistant/Associate Professor,

May, 2022

Department of Computer Science & Engineering,

Patna.

Indian Institute of Technology Patna, Bihar.

Acknowledgements

I want to express my special thanks to Somanath sir and the entire CSE faculty, who gave me the golden opportunity to do this wonderful project on Thyroid Disease Detection, which also helped me to learn about many things. I am really thankful to them. Secondly, I would also like to thank my friends who helped me a lot to finish my project within the limited time.

THANKS AGAIN TO ALL WHO HELPED ME.

Abstract

The thyroid gland is an essential part of the body that helps in the growth, metabolism, and development of the human body. However, levels and hormones related to the thyroid should be in balance. Irregularities in the levels and hormones can cause hyperthyroidism and hypothyroidism. Here we are going to classify subjects on the basis of different levels of their body in 5 classes (1). Hyperthyroid (2) compensated hypothyroid (2) primary hypothyroid (4). secondary hypothyroid (5) negative. This classification can help anyone to cross verify the reports, or this can be used by labs to save time. Classification is done using Random Forest and KNN. On this dataset, unsupervised learning and Random Forest give us an extremely good result which we are going to see further.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The thyroid gland is a butterfly-shaped gland located in the lower part of the neck, and it consists of two lobes. The thyroid gland plays an essential role in maintaining the human's body metabolism, growth, and development. The thyroid gland maintains the various levels in the body, and some of them are Thyroid Stimulating Hormone (TSH), Free Thyroxine Index (FTI), T3, T4 levels, and many others. Balanced between all should be maintained for the better performance of the thyroid gland. If FTI and T4 are high, then TSH should also be increased. If they are low, TSH should also be down. Otherwise, they may lead to hyperthyroidism or hypothyroidism. In hyperthyroid, the gland is underperforming. It cannot match the body's requirements, whereas, in the case of hypothyroidism, it is overperforming and producing an excess amount of hormone.

1.1 Motivation

Around 42 million people in India suffered from various Thyroid diseases in 2019. The report analysis is done mainly by lab experts or doctors, which is time-consuming. The results of reports depend on multiple factors, including age, pregnancy, various levels (TSH, T4, T3), goiter, and many others. We can improve the time of results with better accuracy

by using a classifier that can give us the results in less time.

1.2 Case Study

(1). Studies have suggested that hypothyroidism is more common in India than in the rest of the world. In India, every 1 out of 2500 subjects suffers from hypothyroidism, whereas on a global average stands around 1 in 4000 subjects. It is mainly due to the lack of early diagnosis, which can help cure hypothyroidism. (2). In a clinic-based study from Mumbai, out of 800 children with thyroid disease, 79% had hypothyroidism. (3). Forty-two million suffer from Thyroid diseases in India.

1.3 Organization of The Report

In chapter 2, we have discussed the prior works that have been done in this field and also covered the problem in existing methods. In chapter 3, we have covered data analysis, balancing, and imputing missing values. In chapter 4, we have discussed unsupervised learning, which we have used before using supervised learning. In chapter 5, there is a brief discussion about the Random forest classifier. I have mentioned the parameters for which random forest is providing the best result for each cluster, apart from which we also have the table where we can see the result for each cluster. Chapter 6 is also similar to chapter 5. Here we have discussed KNN, followed by the results from KNN. Chapter 6 is a brief about the User Interface, how UI is working, and how we can use it, and links are also available there to access the UI. Chapter 7 has a conclusion and future work which can be done in this field.

Chapter 2

Review of Prior Works

Previous studies of thyroid disease predictions have used old classification methods, including Naive Bayes, MLP, and J8 classifiers, that do not yield good results. The accuracy for previous methods is 85.1%, 77.3%, and 91.1% which are not that great. We can improve the results by taking advantage of new techniques, including oversampling the dataset, imputing missing values, and including unsupervised learning before supervised learning.

2.1 Problems in prior works

(1). Use of old classifiers. (2). In disease classification, the dataset is highly imbalanced. To overcome this issue, we should oversample our dataset to yield good results. (3). Dataset may have some values missing, so if we can calculate those before training, which will increase the accuracy further.

	NB Classifier	**MLP Classifier**	**J48 Classifier**
Precision	0.851	0.773	0.957
Recall	0.784	0.813	0.958
F-Measure	0.804	0.784	0.957
ROC Area	0.927	0.866	0.982

Table 2.1 Accuracies of existing methods

2.2 Conclusion

The existing studies use an old classifier. We can see the results that I have mentioned in the TABLE 2.1, and the results are not that accurate. We will use newer approaches that we will discuss further in the upcoming chapters, which will help us to improve the results further. These methods will allow us to get good results so that labs and doctors can use our model for day-to-day report checks.

Chapter 3

Data Analysis & Preprocessing

3.1 Exploratory Data Analysis

We have analyzed the dataset to understand how different levels and parameters vary with the different thyroid disease classes, and various parameters include TSH, FTI, TT4, and many others. Apart from this, we will also plot the variation of age with each thyroid disease class.

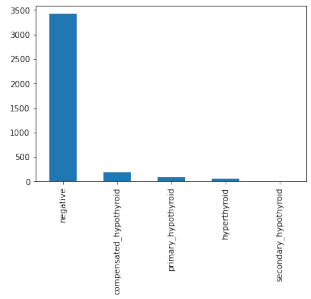

Fig. 3.1 Different Classes

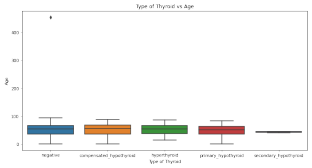

Fig. 3.2 Age vs Thyroid

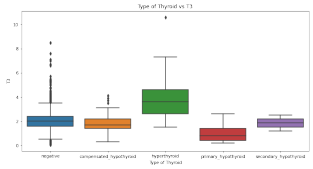

Fig. 3.3 T3 vs Thyroid

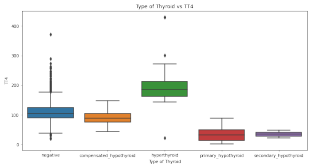

Fig. 3.4 TT4 vs Thyroid

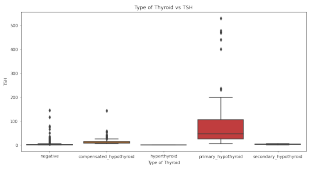

Fig. 3.5 TSH vs Thyroid

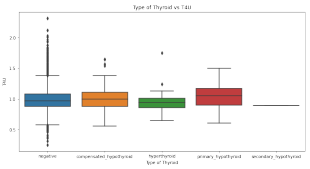

Fig. 3.6 T4U vs Thyroid

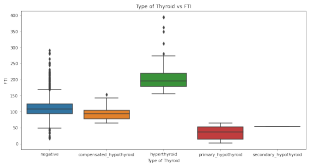

Fig. 3.7 FTI vs Thyroid

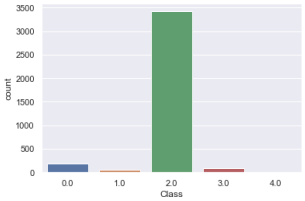

Fig. 3.8 Imabalanced Dataset

3.2 Variation of Parameters

In shown figures, we can see how different parameters change with different thyroid classes. From figure 3.2, we can observe that thyroid is common after around the age of 52. From figure 3.5, we can observe that TSH levels are high in hypothyroidism, whereas they are low in hyperthyroidism. From figure 3.8, we can see that our dataset is highly imbalanced.

3.3 PreProcessing the Dataset

Firstly, we have dropped columns that are not necessary. For example, one column has the value T/F, which means whether TSH is there or not. These kinds of columns are not required. Hence we can drop them.

```
age                          1
sex                        150
on_thyroxine                 0
query_on_thyroxine           0
on_antithyroid_medication    0
sick                         0
pregnant                     0
thyroid_surgery              0
I131_treatment               0
query_hypothyroid            0
query_hyperthyroid           0
lithium                      0
goitre                       0
tumor                        0
hypopituitary                0
psych                        0
TSH                        369
T3                         769
TT4                        231
T4U                        387
FTI                        385
Class                        0
dtype: int64
```

Fig. 3.9 Before Imputing

```
age                          0
sex                          0
on_thyroxine                 0
query_on_thyroxine           0
on_antithyroid_medication    0
sick                         0
pregnant                     0
thyroid_surgery              0
I131_treatment               0
query_hypothyroid            0
query_hyperthyroid           0
lithium                      0
goitre                       0
tumor                        0
hypopituitary                0
psych                        0
TSH                          0
T3                           0
TT4                          0
T4U                          0
FTI                          0
Class                        0
dtype: int64
```

Fig. 3.10 After Imputing

The dataset has some missing values, as shown in figure 3.9, and we have to calculate those to have a better training dataset. To calculate the values, we have used KNNimputer, which we can find in the python library sklearn. After calculating values, we can see there are no missing values, as shown in figure 3.10.

3.4 Balancing the Dataset

As seen in the EDA section, our dataset is highly imbalanced, and we should balance our dataset to have good results on the new data that we will classify.

For this work, we have used RandomOverSampler, which we can find in the python library imblearn.oversampling. Another alternative for RandomOverSampler is SMOTENC, and it was not providing us with the good results, so we have used RandomOverSampler.

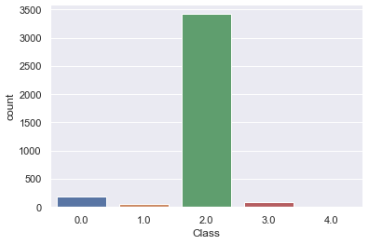

Fig. 3.11 Before Balancing

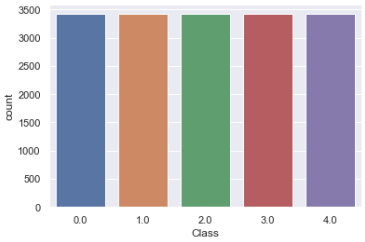

Fig. 3.12 After Balancing

In figure 3.12, we can see our balanced dataset.

Chapter 4

Clustering

Clustering is unsupervised learning. Here we try to divide the population of data points into groups such that data points in the same group are more similar than the data points in other groups. There are two types of clustering (1). Soft Clustering: Here, we have a probability of data points belonging to any particular group. (2). Hard Clustering: Here, our data will only belong to one group, which means either any data point will belong to the group or it will be in the other group. There are mainly two clustering algorithms (1). KMeans Clustering (2) Hierarchical Clustering. In this work, we have used KMeans Clustering.

4.1 How do we find optimal clusters?

We have used an elbow plot, and it is a method that is used to find the optimal number of clusters based on the heuristic distances. We try to find the knee or elbow of the curve, which we plot on the basis of WCSS stands for Within-Cluster-Sum-of-Squares (WCSS). We choose the knee or elbow point because this means adding another cluster will decrease the WCSS value by a significant amount. The intuition is that with an increase in the number of clusters, we will have a better fit, but however increase them too much can lead to overfitting, which can reduce the accuracy by a significant amount. Elbow point gives

us that intuition for the optimal number of clusters.

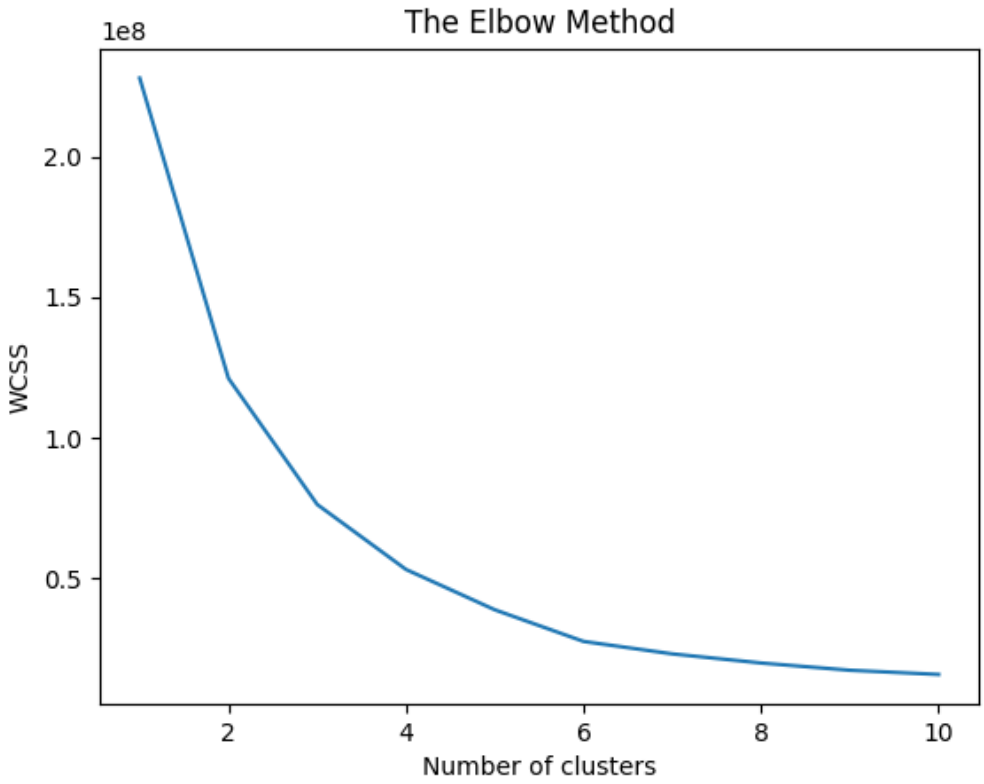

Fig. 4.1 Clustering vs WCSS

4.2 Clusters for our Dataset

In figure 6.1, we can see the elbow at 4 clusters. Using elbow point, we have clustered the dataset into four groups by using the KMeans function, which is available in the python library sklearn.cluster.

Chapter 5

Random Forest

Random forest is an ensembled supervised machine learning algorithm, and it is widely used in classification and regression problems. In a random forest, we have multiple trees. For each tree, we fetch some data points, and some feature from our dataset to train it. We repeat this step multiple times until n_estimators trees are formed. For final prediction, we use the voting mechanism or probabilities, whichever class has maximum votes or whichever has the highest probability that is going to be our predicted class. Fig 7.1 shows how random forest works.

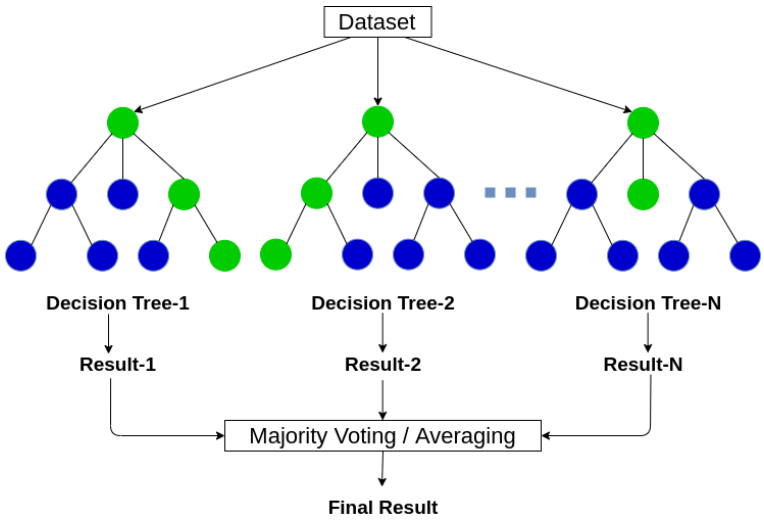

Fig. 5.1 Random Forest

5.1 Parameters for Random Forest

n_estimators: Number of trees in the forest.

criterion: Function measure the quantity of a split. Two types of criterion are supported (1). gini (2). entropy

max_depth: This determines the max depth of the tree.

max_features: Number of features that need to be considered while splitting. We have checked the algorithm on various parameters, which are given in figure 7.2.

```
# initializing with different combination of parameters
param_grid = {"n_estimators": [10, 50, 100, 130, 200, 300],
              "criterion": ['gini', 'entropy'],
              "max_depth": range(2, 6, 1),
              "max_features": ['auto', 'log2']}
```

Fig. 5.2 Parameters of Random Forest

5.2 Results from Random forest

Random Forest best params for Cluster 1: {'criterion': 'entropy', 'max_depth': 5, 'max_features': 'auto', 'n_estimators': 200}.

Random Forest best params for Cluster 2: {'criterion': 'gini', 'max_depth': 5, 'max_features': 'log2', 'n_estimators': 100}.

Random Forest best params for Cluster 3: {'criterion': 'gini', 'max_depth': 2, 'max_features': 'auto', 'n_estimators': 10}.

Random Forest best params for Cluster 4: {'criterion': 'gini', 'max_depth': 5, 'max_features': 'auto', 'n_estimators': 100}.

Training accuracy using these parameters is 99.7%. Accuracy for test data has been shown in the TABLE.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**
Precision	0.9825	0.9940	0.9876	0.9841
Recall	0.9322	0.9222	0.8	0.9506
F-Measure	0.9570	0.9422	0.7333	0.9701
AUC	0.9974	0.9990	.9990	0.9822

Table 5.1 Accuracy of Random Forest on each cluster

Chapter 6

KNN - K Nearest Neighbors

It is a supervised machine learning algorithm, and this algorithm works on the assumption that similar things will exist in close proximity. We choose the closest K points to our data points, and we will check the label of all K neighbors, and whichever has the majority that will win. Fig 8.1 shows how KNN works.

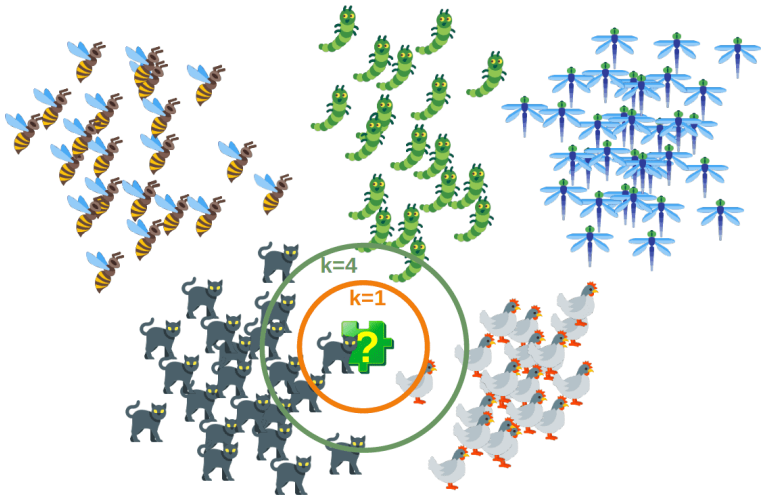

Fig. 6.1 KNN

6.1 Parameters for KNN

n_neighbours: Number of neighbors to determine the class.

algorithm: 3 types of the algorithm are available (1). ball_tree (2). KD_tree (3). brute.

p: p stands for distance used. Two types of distance are there (1). Manhattan distance (2) Euclidean distance.

leaf_size: it is used when we use ball or kd tree. It can help us to speed up the process and optimize the tree.

```
#initializing with different combination of parameters
param_grid_knn = {
    'algorithm' : ['ball_tree', 'kd_tree', 'brute'],
    'leaf_size' : [10,17,24,28,30,35],
    'n_neighbors':[4,5,8,10,11],
    'p':[1,2]
}
```

Fig. 6.2 Parameters of KNN

6.2 Results from KNN

KNN best params for Cluster 1: {'algorithm': 'ball_tree', 'leaf_size': 17, 'n_neighbors': 4, 'p': 1}.

KNN best params for Cluster 2: {'algorithm': 'ball_tree', 'leaf_size': 10, 'n_neighbors': 4, 'p': 1}.

KNN best params for Cluster 3: {'algorithm': 'ball_tree', 'leaf_size': 10, 'n_neighbors': 4, 'p': 1}.

KNN best params for Cluster 4: {'algorithm': 'kd_tree', 'leaf_size': 10, 'n_neighbors': 5, 'p': 1}.

Training accuracy using these parameters is 98.9%. Accuracy for test data has been shown in the TABLE.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**
Precision	0.9777	0.9919	0.9506	0.9657
Recall	0.7391	0.8875	0.3974	0.3012
F-Measure	0.7365	0.9133	0.2653	0.3277
ROC Area	0.9970	0.9744	.0.9831	0.9558

Table 6.1 Accuracy of KNN on each cluster

Chapter 7

User Interface

For interaction purposes, I have developed a UI that can be either used by labs or by a normal person to verify the reports.

Tech Stack: HTML, Flask and CSS

Deployed on HEROKU

Fig. 7.1 Flask

Fig. 7.2 HTML & CSS

HTML: It has been used to design the front-end of the application, which we are going to view on the webpage. HTML provides us with the basic structure of the webpage.

CSS: It has been used to give style to our webpage.

Flask: It has been used to connect the front-end with the backend. With the flask, we are using our ML models, which we saved after training on the dataset.

Heroku: It is a deployment service that is available for free.

7.1 How is UI working?

First, we are taking the input from the user in the format of CSV or form. After taking the input, we calculate the cluster number for each datapoint. Based on the cluster number, we will call different saved models.

We can see the application's architecture in figure 7.3.

7.2 Working of UI for lab prediction

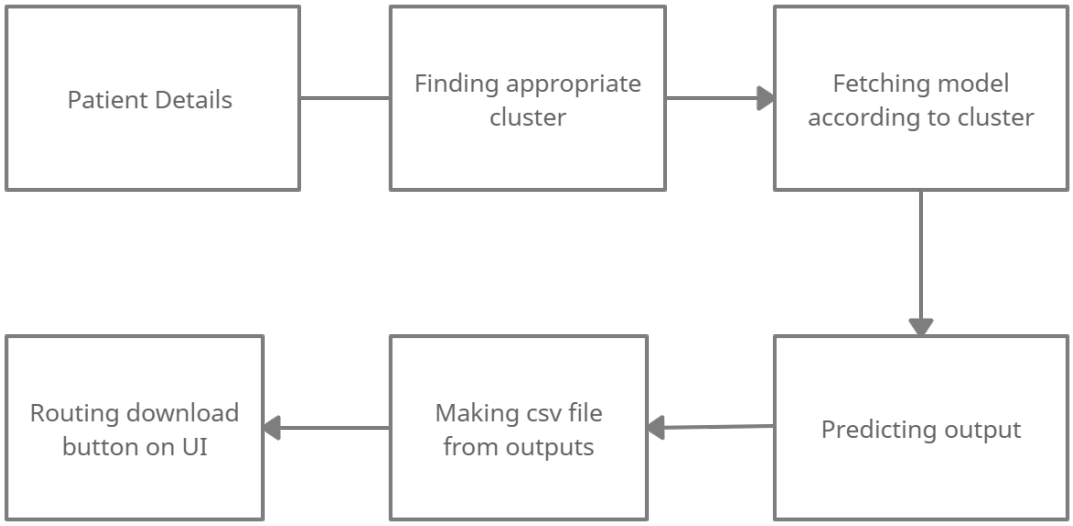

Fig. 7.3 Application's Architecture

7.3 Working of UI for one prediction

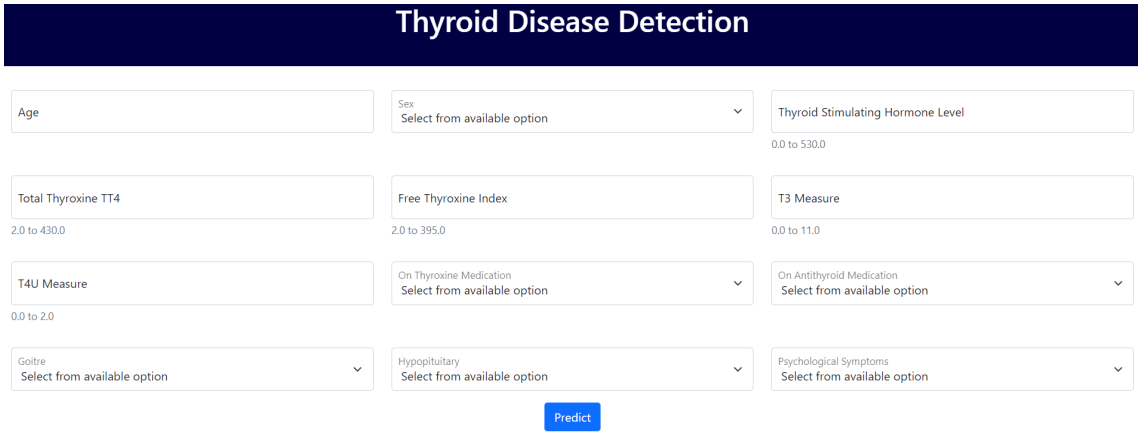

Fig. 7.4 Form for patient details

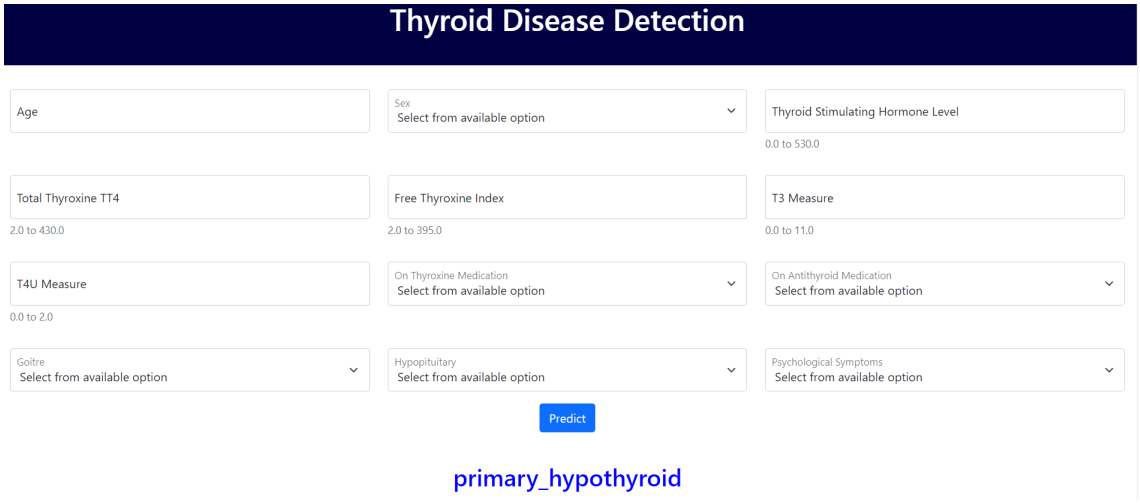

Fig. 7.5 Result for the patient

(1). Fill out the form as shown in figure 7.4, (2). Click on the predict button, and after prediction, the result is visible at the bottom of the screen as shown in figure 7.5.

7.4 Working of UI for lab prediction

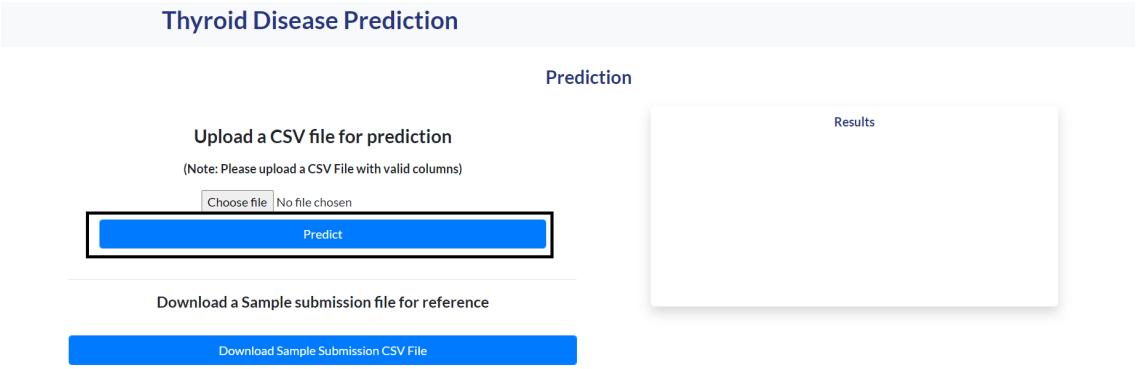

Fig. 7.6 Button to download Sample CSV

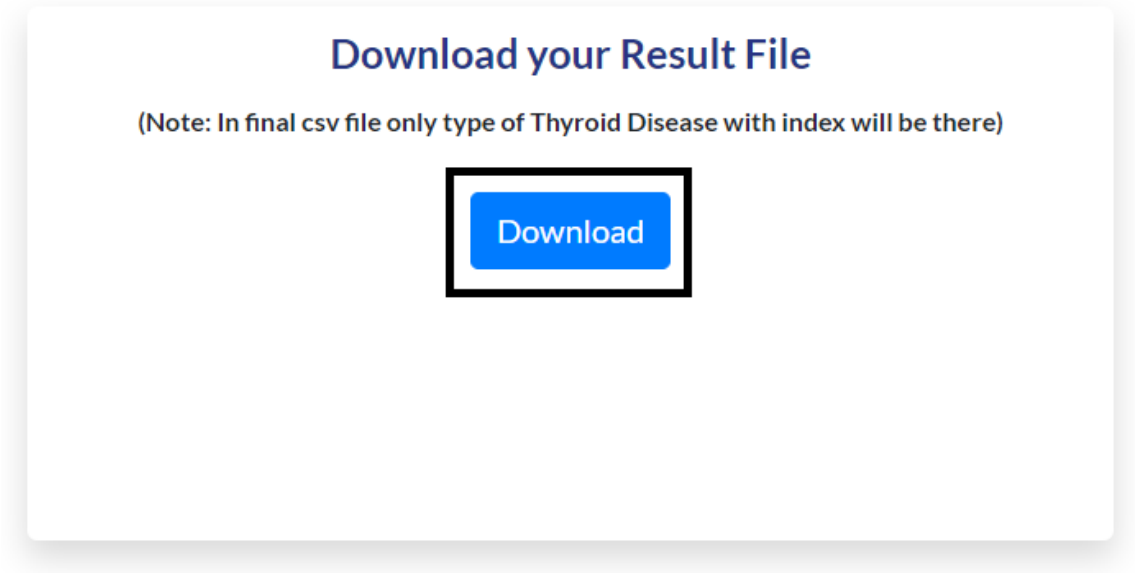

Fig. 7.7 Download button for Results

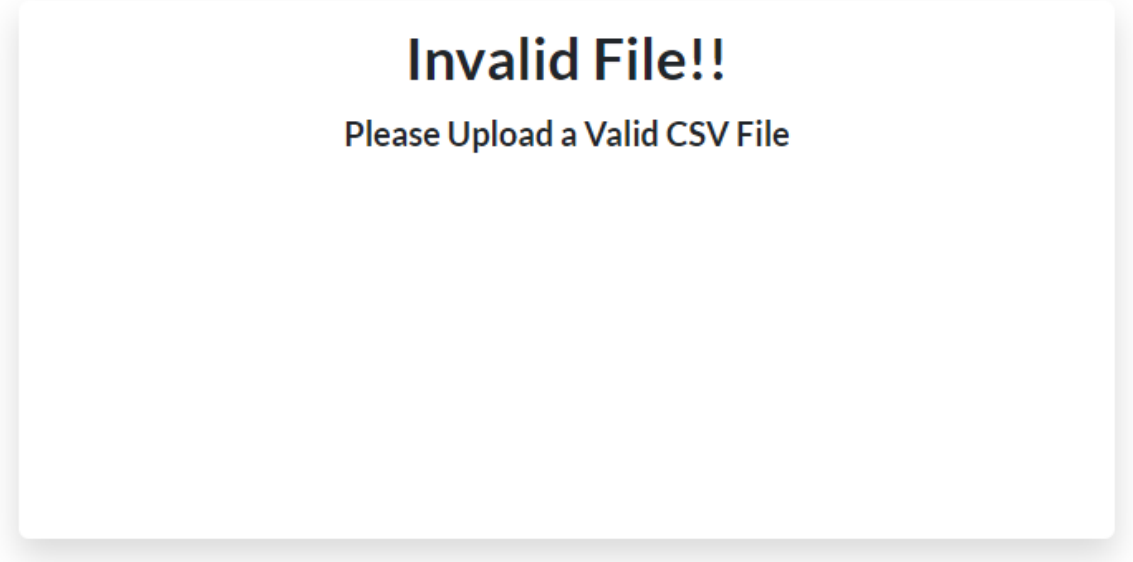

Fig. 7.8 Error if input CSV is not valid

(1). Input format is a CSV with all the patient details. For reference sample, CSV can be downloaded by clicking on the sample CSV file as shown in fig 7.6. (2). For prediction, we can upload a CSV file that consists of patient details, and after clicking on predict, we can see an option to download the results, as shown in fig 7.7. If the input file is not the valid file, then we can see the error as shown in fig 7.8.

7.5 Link of the Deployed Apps

Both apps have been deployed on Heroku link can be found here:

https://thyroid-one-prediction.herokuapp.com/

https://thyroid-lab-prediction.herokuapp.com/

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Using semi-supervised learning before supervised have helped us to achieve good accuracies, which are better than previous. Results of the current study have been mentioned in TABLE 8.1. We can observe that Random forest is performing better than KNN in every cluster. Apart from this, a UI has been deployed that anyone can use. For the final score, I have taken the weighted averages.

	Random Forest	**KNN**
Precision	0.9874	0.9805
Recall	0.9298	0.6476
F-Measure	0.9502	0.7175
ROC Area	0.9947	0.9786

Table 8.1 Random Forest vs KNN

8.2 Future Work

We can also test our dataset on algorithms such as XG boost, and we can also give a shot of CNN for classification, which may provide us with more accurate results. Apart from data classification, we can also include Ultrasound image classification on thyroid gland,

which can give information about the early stage of hypothyroidism and hypothyroidism.