THYROID DETECTION

Detailed Project Report

By: Nayana J

INTRODUCTION

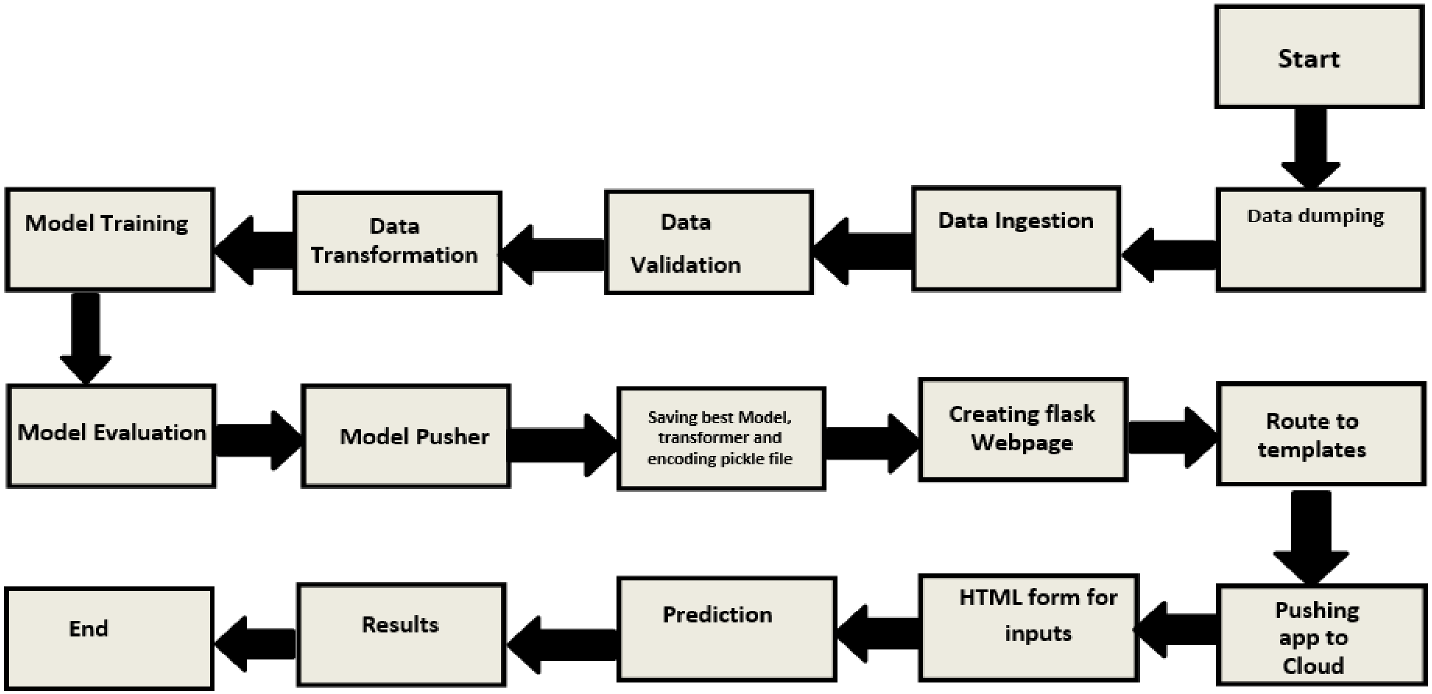
At least a person out of ten is suffered from thyroid disease in India. The disorder of thyroid disease primarily happens in the women having the age of 17–54. The extreme stage of thyroid results in cardiovascular complications, increase in blood pressure, maximizes the cholesterol level, depression, and decreased fertility. The hormones, **total serum thyroxin (T4)** and **total serum triiodothyronine (T3)** are the two active thyroid hormones produced by the thyroid gland to control the metabolism of body. For the functioning of each cell and each tissue and organ in a right way, in overall energy yield and regulation and to generate proteins in the ordnance of body temperature, these hormones are necessary.

Hyperthyroidism and **Hypothyroidism** are the two most common diseases caused by irregular function of thyroid gland. Thyroid disorder can speed up or slow down the metabolism of the body. In the world of rising new technology and innovation, the health care industry is advancing with the role of Artificial Intelligence. Machine learning algorithms can help to detect detection of the disease and to improve the quality of life. This study demonstrates how different classification algorithms can forecast the presence of the disease. Different classification algorithms such as Logistic regression, Random Forest, Decision Tree, Support Vector Machine, XG Boost, KNN have been tested and compared to predict the better outcome of the model.

OBJECTIVE

The main goal of this project is to predict the risk of hyperthyroid and hypothyroid basedon various factors of individuals. Thyroid disease is a common cause of medical diagnosisand prediction, with an on set that is difficult to forecast in medical research. It will playa decisive role in early detection, accurate identification of the disease and helps the doctors to make proper decisions and better treatment.

ARCHITECTURE


Start

Data dumping

Data Ingestion

Data Validation

Data Transformation

Model Training

Model Evaluation

Model Pusher

Saving best Model, transformer and encoding pickle file

Creating flask Webpage

Route to templates

Pushing app to Cloud

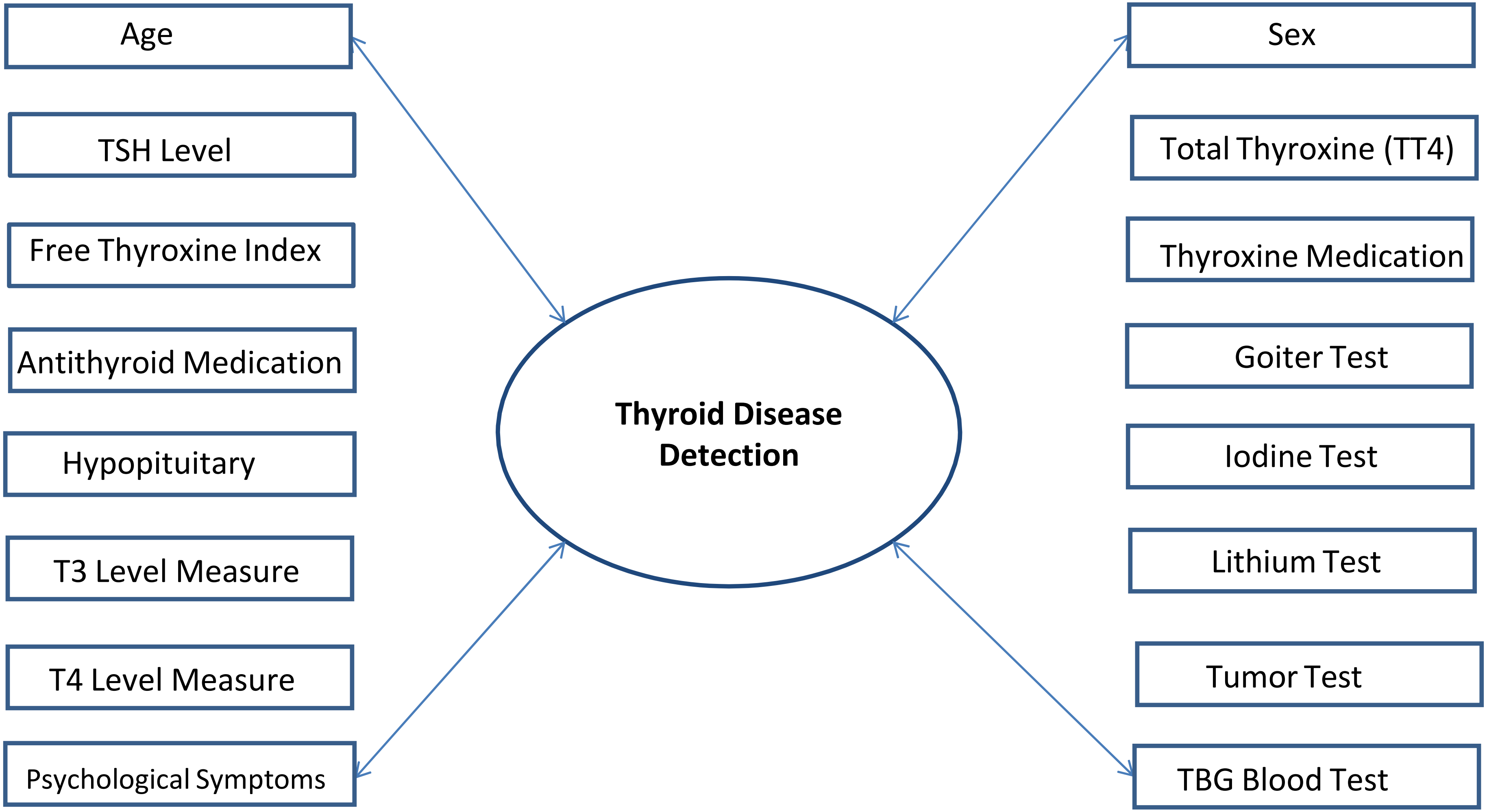
HTML form for inputs

Prediction

Results

End

DATASET



Age

TSH Level

Free Thyroxine Index

Antithyroid Medication

Hypopituitary

T3 Level Measure

T4 Level Measure

Psychological Symptoms

Thyroid Disease Detection

Sex

Total Thyroxine (TT4)

Thyroxine Medication

Goiter Test

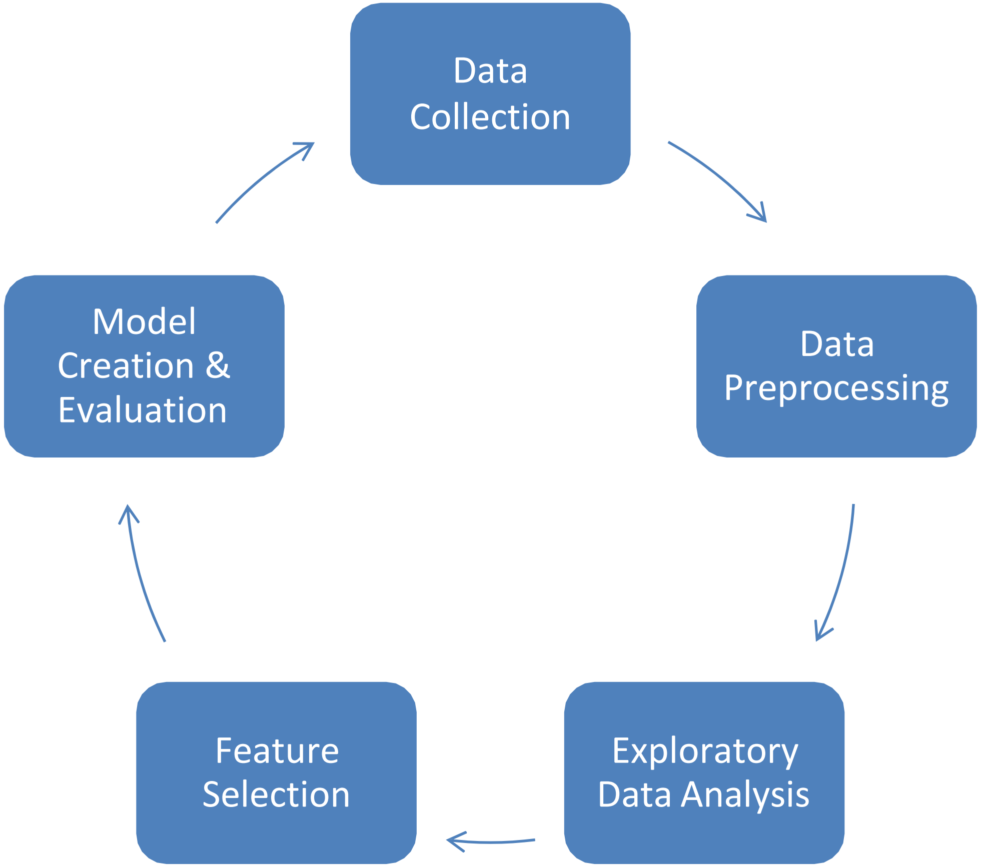
Iodine Test

Lithium Test

Tumor Test

TBG Blood Test

DATA ANALYSIS STEP



Data Collection

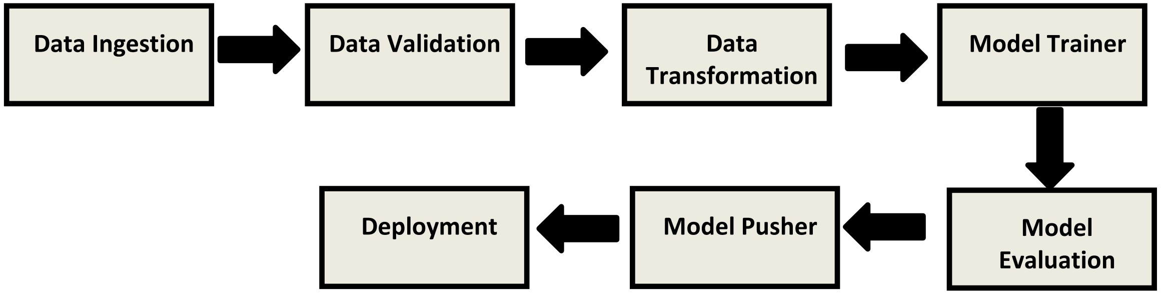
Data Preprocessing

Exploratory Data Analysis

Feature Selection

Model Creation & Evaluation

MODEL TRAINING AND EVALUATION



Data Ingestion

Data Validation

Data Transformation

Model Trainer

Model Evaluation

Model Pusher

Deployment

Data Ingestion

- Thyroid Disease Data Set from UCI Machine Learning Repository
- Data ingested to MongoDB and imported to ingest the data for use.
- For Data Set: https://archive.ics.uci.edu/ml/datasets/thyroid+disease

Data Validation

- Checking if the required number of columns ae present or not.
- If the column contains null values more than the threshold, we just remove it.
- Create a yaml file to report the error.
- Done train test split and save it inside the artifacts folder.

Data Transformation

- Missing values handling by Simple imputation (Used KNN Imputer).
- Categorical features handled by ordinal encoding and label encoding.
- Feature scaling done by MinMax Scaler method.
- Applied Column Transformation to create a pipeline.
- Saved the transformed object to artifacts folder.

Model Trainer

- Various classification algorithms like Random Forest, XG Boost, KNN etc tested.
- Random Forest, XGBoost and KNN all were given better results.
  Decision Tree was chosen for the final model training and testing.
- Hyper parameter tuning was performed.
- Model performance evaluated based on accuracy, confusion matrix, classification report.

Model Evaluation

- To evaluate the performance of the model by comparing the model accuracy with any previously saved object file.
- If the model performs better, then save the latest object file with latest numeric file name.

Model Pusher

- To check if there any new entry of model object happened
- If there is any new model object detected save it to saved_model folder.

Decision Tree Classifier Model

INTRODUCTION

A Decision Tree Classifier is a machine learning algorithm used for classification problems. It works by recursively splitting the input data into subsets based on the values of the features. The goal is to create a tree-like structure that represents the decision-making process of the algorithm. At each node of the tree, the algorithm selects a feature that splits the data into two or more subsets, based on a certain criterion, such as entropy or Gini impurity. The algorithm then recursively applies this process to each subset, creating a tree structure that represents a sequence of decisions that lead to the final classification. The advantage of using a Decision Tree Classifier is that it is easy to interpret and visualize, making it a popular choice for data analysis and decision-making. However, it can be prone to overfitting, meaning that it can become too complex and specialized to the training data, which can reduce its accuracy on new, unseen data. Therefore, techniques such as pruning or ensembling with other models can be used to improve its performance.

Reason to use DecisonTree Classifier model:

- Easy to interpret and visualize.
- Decision trees can be trained very quickly, especially on small to medium-sized datasets.
- Hypertuning gave the best result in DecisionTree Classifier.

Prediction Result

- The model gave the training accuracy of 93.24% and test accuracy of 89.14%

Model Deployment

- The final model is deployed on Azure using Flask framework.

- Implemented batch prediction to generate predictions at scale in a fast and efficient manner.
- Used GitHub Actions for CI/CD implementation.

FREQUENTLY ASKED QUESTIONS

Q1) What is the source of data?

The data for training is obtained from famous machine learning repository.

UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/thyroid+disease

Q2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q3) What's the complete flow you followed in this Project?

Refer slide 7th, 8th and 9th for better understanding.

Q4) After the File validation what do you do with incompatible file or files which didn't passthe validation?

Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q5) How logs are managed?

We are using different logs as per the steps and each logs are printed in the logs folder. Anyone can refer to check the flow of the code.

Q 6) What techniques were you using for data pre-processing?

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.

Q 7) How training was done or what models were used?

- Data preprocessing is done, and data is splitted into train and test file.
- Used that data for model transformation with the help of ColumnTransformer.
- Did model training with the help of decision tree with max_depth=8.

Q 8) How Prediction was done?

- Flask webpage was created for user interface to provide details to the given form.
- Prediction page will show the predicted result on the basis of user inputs.

THANK YOU