Architecture

Thyroid Disease Detection

Revision Number: 1.1

Last date of revision: 01/02/2021

Sai Subhasish Rout

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 4th Feb 2023 | 1.1 | First Draft | Sai Subahsish Rout |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_gjdgxs)

[**Abstract** 4](#_30j0zll)

[1](#_1fob9te) Model training/validation work flow 5

[2](#_4d34og8) Architecture 7

[2.1 Batch Prediction 7](#_2s8eyo1)

[2.2 Real-time Prediction 7](#_17dp8vu)

[3](#_26in1rg) Architecture Description 8

3.1 Data Description

3.2 Data Ingestion

3.3 Data Validation

3.4 Data Transformation

3.5 Model Trainer

3.6 Model Pusher

3.7 Model Evaluation

3.8 Deployment

4User I/O workflow 12

5Key Performance Indicator (KPI) 13

# Abstract

There are types of thyroids and we need to identify if the patient has thyroid or not. If it is a positive case then what type of thyroid the person is suffering from. We need to build a model which will be used by hospitals. In the first case of spatialization the model will predict if the person is suffering from thyroid or not. If the result will come positive, then the treatment will be on fast-track. The doctors will start treating the patients and identify the patient is suffering from hypo-thyroid or hyper-thyroid. If the result will come negative then the patient will be sent to a junior doctor and the junior doctors by using their own expertise they will decide that if the model has done correct prediction or not. Based the prediction comes true then the doctor release the patient. By seeing the readings if doctor analysed that there may be a chance of thyroid then patient sent to the senior doctors

# Model training/validation workflow



1. **Architecture**

**2.1 Batch Prediction**

Data Transformation

Lab test data

Data validation

Start

Application Start

Batch data from user

Pushing app to cloud

Model Building

End

Displaying output

Label Decoder

Predicting output

**2.2 Real-time prediction**

Lab test data

Lab test data

Lab test data

Start

Data from User Interface

Application Start

Pushing app to cloud

Model Building

End

Displaying output

Label Decoder

Predicting output

# Architecture Description

## 3.1 Data Description

Thyroid Detection dataset contains 3772 records and 29 features & one columns for label.

## Data Ingestion

In data ingestion step we are retrieving data from database and storing it to feature store folder, from where ml pipeline starts. Then we are splitting our data set to train and test file with a test size 0.2.

## Data Validation

In data validation we are checking the data-drift, datatypes of features, number of features as per base data, missing value percentage and preparing a report. And dropping the missing value columns below threshold 0.2. Along with that we are dropping unnecessary features in the dataset then saving a train & test file which will be used in data transformation.

* 1. **Data Transformation**

In the Transformation Process, we will convert we are converting categorical data to numerical, missing value, handling outliers, splitting data to train and test, label encoding, handling imbalanced data.

* 1. **Model Trainer**

After clusters are created, we will find the best model for each cluster. For each cluster, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the AUC scores for models and select the model with the best score. Similarly, the models will be selected for each cluster. All the models for every cluster will be saved for use in Recommendation.

## Model Evaluation

In model evaluation we evaluate the newly trained model’s accuracy with old model’s accuracy and keep the new model to saved models folder if it’s accuracy comes greater than previous model.

## Model Pusher

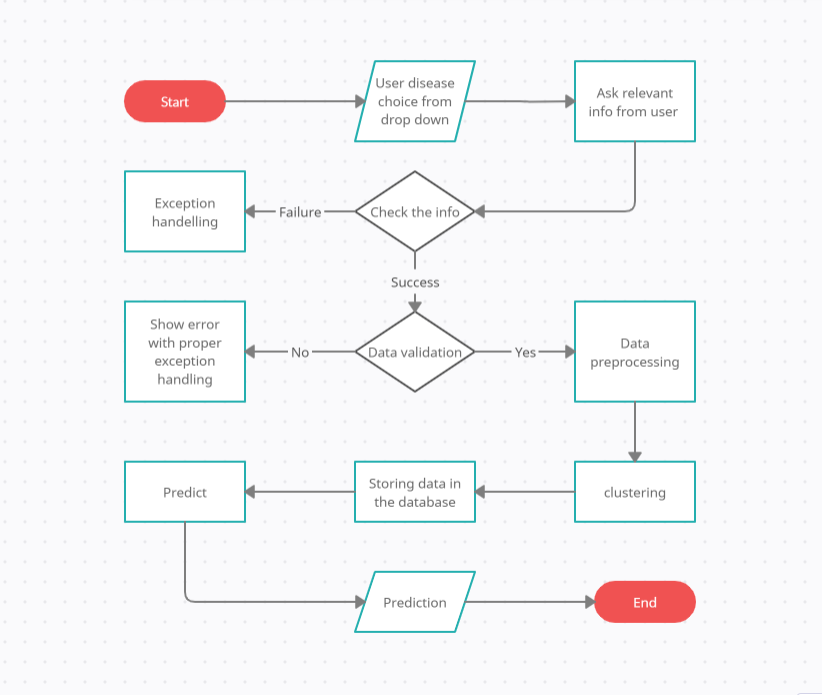
In model pusher component we are pushing the model with greater accuracy to production.

## Deployment

In deployment process we are first dockerizing our project by removing the dependencies. Then we are saving it to ECR by testing in local. And pushing it to EC2 instance. Using Airflow we are accessing the application.

And for real-time prediction we are creating an User Interface through which authorities can enter the values of a patient to predict the outcome.

# User I/O workflow

****

# Key performance indicators (KPI)

* Time and workload reduction using the EHR model.
* Comparison of accuracy of model prediction and doctor’s prediction.
* Number of times a patient visits the hospital.
* Time between symptom onset and detection of illness/visit to hospital.
* Immunity of patient (based on previous illnesses).
* Vaccines the patient has taken.
* Length of stays in hospital.