Architecture

Thyroid Disease Detection System

Written By	Nayana J
Document Version	1.0
Last Revised Date	

Document Version Control

Change Record:

Version	Date	Author	Comments
1.0	16-03-2024	Nayana J	Architecture

Reviews:

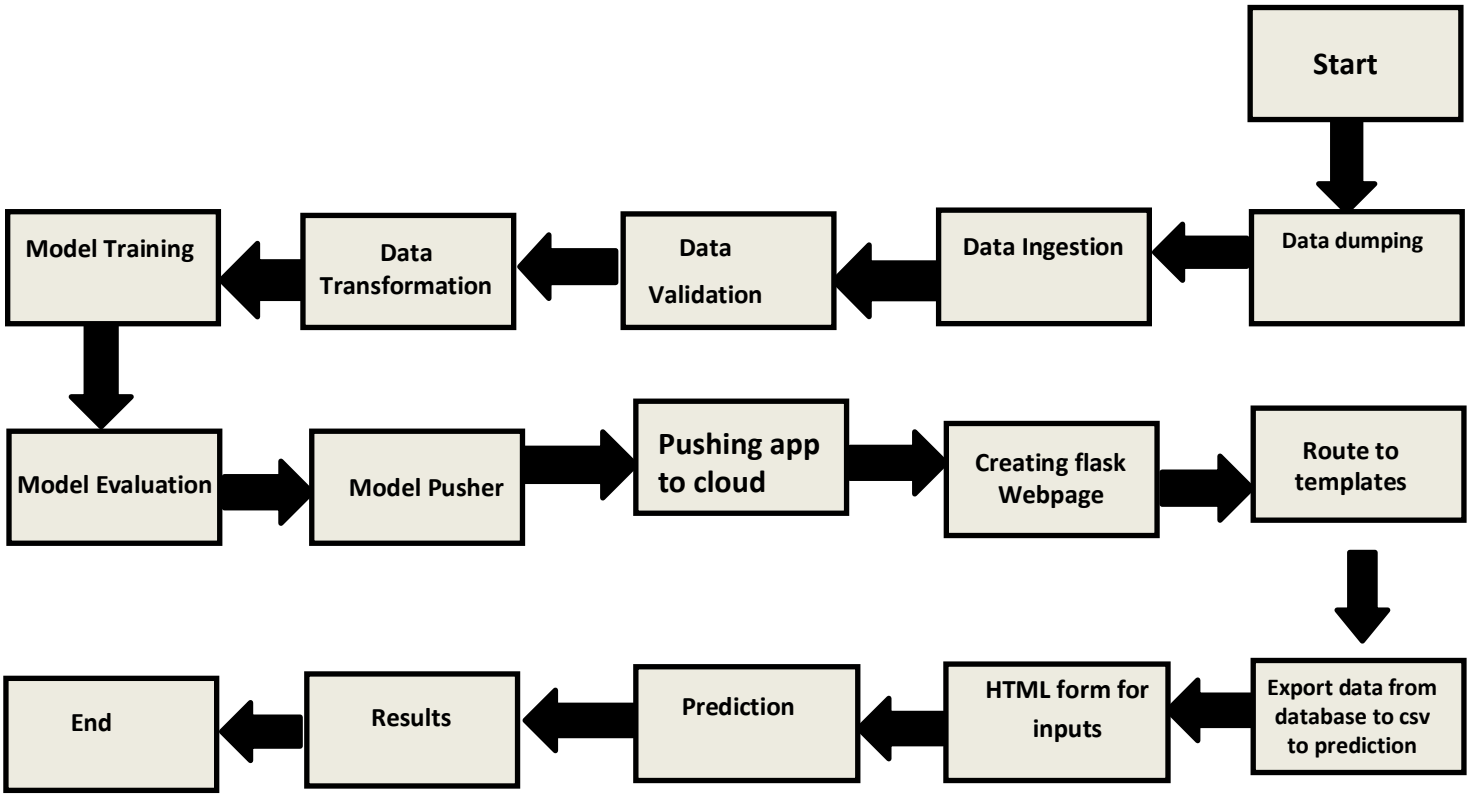
Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Architecture



2. Architecture Description

2.1 Data Description

We will be using Thyroid Disease Data Set present in UCI Machine Learning Repository. This Data set is satisfying our data requirement. Total 9069 records present.

2.2 Export Data from database to CSV for Training

Here we will be exporting all batches of data from the database into one csv filefor training.

2.3 Data Validation

Here I used different cleaning of data and setting the dataset cleaned for further operations. Here I also did the splitting of data in train and test file and send it to artifacts folder.

2.4 Data Transformation

In this section I have done the feature selection and transformation of the columns such as Ordinal Encoder, One Hot Encoding, KNNImputer and Min Max Scaler and imputed on splitted data and created the object and sent it to artifacts folder.

2.5 Model Trainer

Decision Tree Classifier is selected as it gave the best accuracy after fine tuning the Hyperparameter and giving best accuracy for max_depth=8. Also creating the object and saving it to the artifacts folder.

2.6 Model Evaluation

Here I evaluated the model accuracy and checked if the model has already had any previous object that is giving better accuracy in compared to the current object. The best model is to be saved in the artifacts folder.

2.7 Model Pusher

Here the best accurate model object is to be sent to the saved_model folder with the naming convention as the latest file object has highest numeric.

2.8 Flask Webpage

Here I created a flask webpage to take the inputs from the user and showing the result with the help of the predictions. Here two HTML templates were designed.

2.10 Using GitHub actions to create CI/CD pipeline

Here code is written to interact with the Azure and connect GitHub to Azure to run continuous deployment.