ARCHITECTURE

Thyroid Disease Detection System

Written by - Arkoprovo Ghosh

Document Version Control

Change Record:

Version	Date	Author	Comments
1.0	29-07-2023	Arkoprovo Ghosh	Architecture

Reviews:

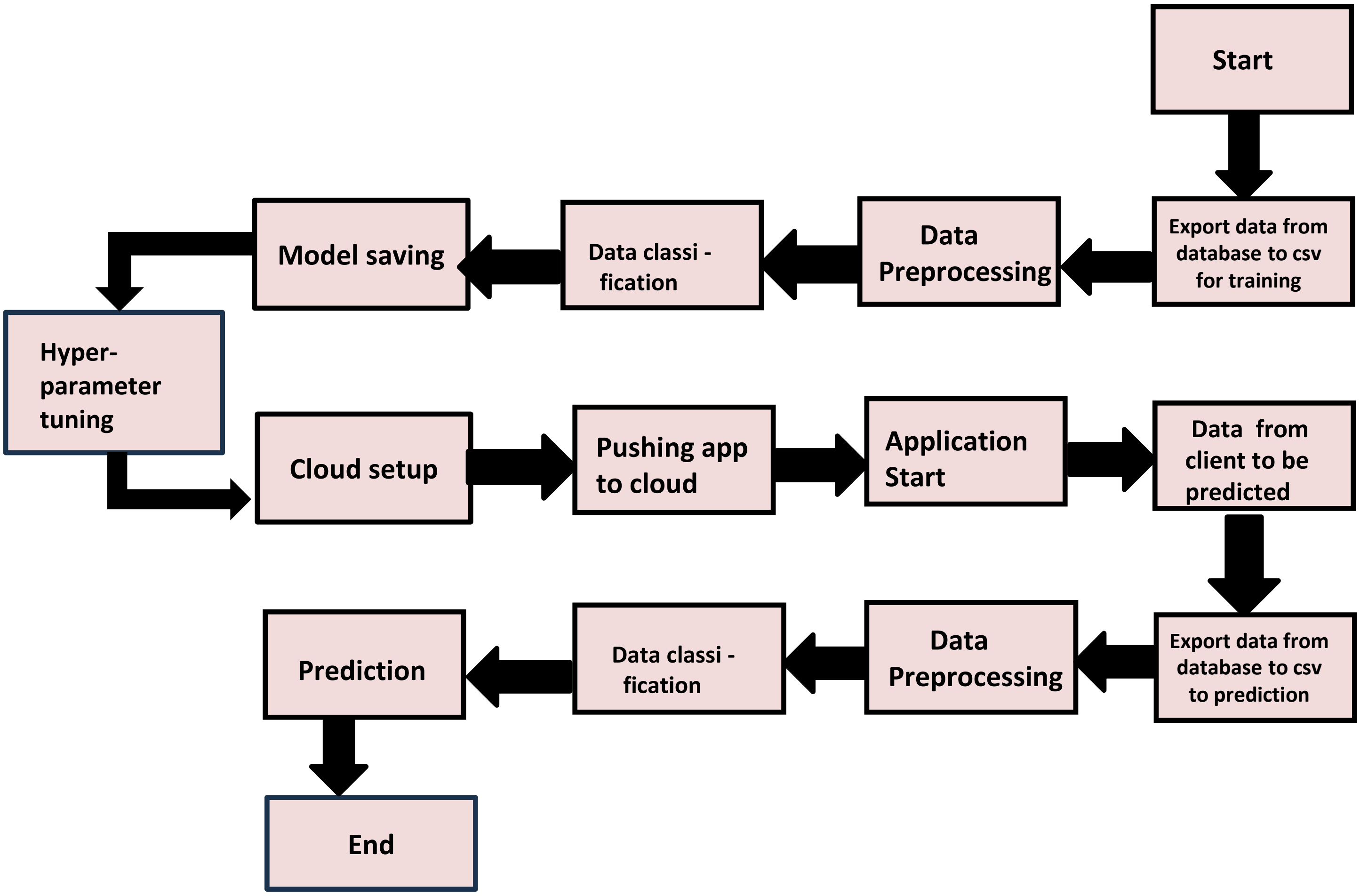
Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

2. Architecture



Architecture Description: Thyroid Disease Prediction

3.1 Data Description:

We will use the Thyroid Disease Data Set from the UCI Machine Learning Repository, which contains 7200 instances in different batches of data.

3.2 Export Data from Database to CSV for Training:

We will export all batches of data from the database into a single CSV file to use it for model training.

3.3 Data Preprocessing:

In this step, we will explore the dataset and perform exploratory data analysis (EDA) if necessary. Based on our analysis, we will conduct data preprocessing tasks like handling null values, dropping unnecessary columns, etc. We will create separate modules for these preprocessing steps to be used during both training and prediction.

3.4 Data Classification:

We will do data classification using the Random Forest technique involves employing an ensemble of decision trees to categorize data into predefined classes or groups. The algorithm builds multiple decision trees on different subsets of the data and combines their predictions to make the final classification. This approach enhances accuracy, reduces overfitting, and handles complex datasets effectively .The trained random forest model will be saved for future predictions.

3.5 Hyperparameter Tuning:

After classification ,we will do hyperparameter tuning to get best results for prediction.

3.5 Model Saving:

After classification, we will save all the trained models so that they can be used for making predictions.

3.6 Cloud Setup:

We will set up the cloud environment for model deployment. This involves creating a Streamlit app and user interface to integrate our trained models. The streamlit app will serve as the backend, while the user interface will interact with the models.

3.7 Push App to Cloud:

Once the cloud setup is complete and the app has been tested locally, we will push the entire application to the cloud to make it publicly accessible.

3.8 Data from Client Side for Prediction Purpose:

With the application on the cloud, we can start receiving prediction data from clients. The data received from the client will go through the same data cleansing process as the training data, utilizing the modules developed earlier. The data will undergo data preprocessing, classification, and will be fed into the appropriate saved models for predictions.

3.9 Prediction displayed:

After undergoing all the above steps, finally the result i.e., for thyroid prediction, is displayed on the application.