Architecture

Thyroid Disease Detection System

Written By	Ashwini Kakde
Document Version	1.0
Last Revised Date	

Document Version Control

Change Record:

Version	Date	Author	Comments
1.0	12-06-2023	Ashwini Kakde	Architecture

Reviews:

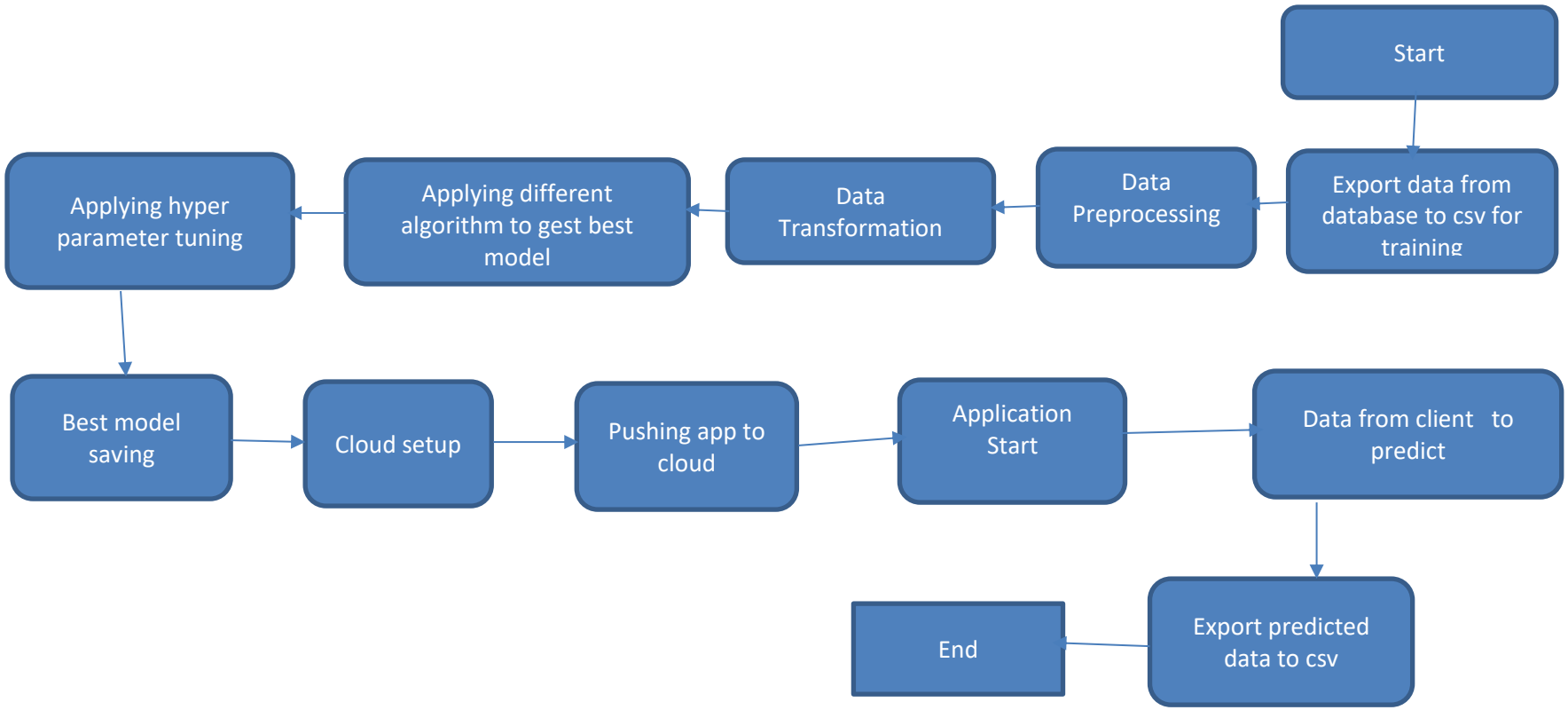
Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

LOW LEVEL DESIGN (LLD)



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 7200 instances present in different batches of data.

2.2 Export Data from database to CSV for Training

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

2.3 Data Preprocessing

We will be exploring our data set here and do EDA if required and perform data preprocessing depending on the data set. We first explore our data set in Jupyter Notebook and decide what pre-processing and Validation we have to do such as imputation of null values, dropping some column, etc and then we have to write separate modules according to our analysis, so that we can implement that for training as well as prediction data.

2.4 Data Transformation

The data transformation phases involves converting all categorical values into numerical format .Missing value imputation for numerical and categorical data .Numerical values are filled with median values and categorical values are filled with most frequent values and imbalanced data is handled by random over sampler.

2.5 Get best model of different applied algorithm models

Here we will train various model on each cluster which we will obtain in Data Clustering, and then will try to get best model of each cluster.

2.6 Hyper parameter Tuning

After selecting best model for each cluster, we will do hyper parameter tuning for each selected model, and try to increase performance of themodels.

2.7 Model Saving

After performing hyper parameter tuning for models, we will save our models so that we can use them for prediction purpose.

2.8 Cloud Setup

Here we will do cloud setup for model deployment. Here we also create our Flask app and user interface and integrate our model with flask app and UI

2.9 Push app to cloud

After doing cloud setup and checking app locally, we will push our app to cloud to start the application.

2.10 Data from client side for prediction purpose

Now our application on cloud is ready for doing prediction. The prediction data which we receive from client side will be exported from DB and further will do same data cleansing process as we have done for training data using modules we will write for training data. Client data will also go along the same process of **Exporting data from DB**, **Data pre-processing**, **and Data clustering** and according to each cluster number we will use our **saved model** for prediction on that cluster.

2.11 Export Prediction to CSV

Finally when we get all the prediction for client data, then our final task is to export prediction to csv file and hand over it to client.