Architecture

Thyroid Disease Detection System

Written By	Dheerendra Kushwaha
Document Version	1.0
Last Revised Date	

Document Version Control

Change Record:

Version	Date	Author	Comments
1.0	10-03-2025	Dheerendra Kushwaha	Architecture

Reviews:

Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

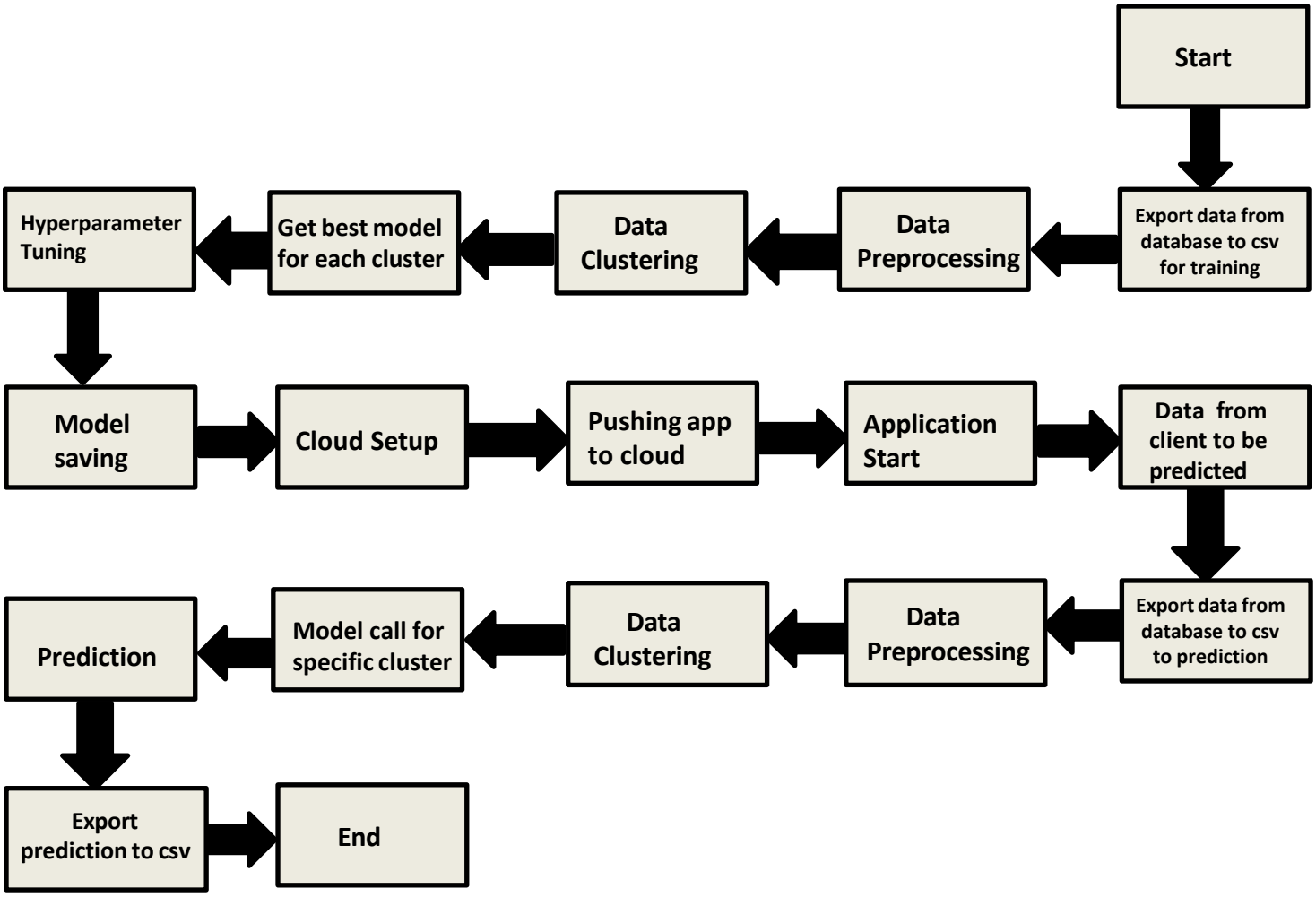
Contents

Document Version control----------2

1 Architecture----------4

2 Architecture Description---------- 5
2.1 Data Description ----------5
2.2 Export Data from DB to CSV for training---------- 5
2.3 Data Preprocessing----------5
2.4 Data Clustering----------5
2.5 Get best model of each cluster ----------5
2.6 Hyperparameter Tuning----------5
2.7 Model saving----------6
2.8 Cloud setup----------6
2.9 Push App to cloud----------6
2.10 Data from client side for prediction ---------- 6
2.11 Export prediction to CSV----------6

2. Architecture



3. Architecture Description

3.1 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.

3.2 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.

3.3 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.

3.4 Data Clustering

K-Means algorithm will be used to create clusters in the pre-processed data. The optimum number of clusters is selected by plotting the elbow plot. The idea behind clustering is to implement different algorithms to train data in different clusters. The K-means model is trained over pre-processed data and the model is saved for further use in prediction

3.5 Get best model of each cluster

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.

3.6 Hyperparameter Tuning

After selecting best model for each cluster, we will do hyperparameter tuning for each selected model, and try to increase performance of the models.

3.7 Model Saving

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

3.8 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

3.9 Push app to cloud

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

3.10 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**, **Data clustering** and according to each cluster number we will use our **saved model** for prediction on that cluster.

3.11 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.