Architecture

Thyroid Disease Detection System

Written By	Komal Madiwal
Document Version	1.0
Last Revised Date	

Document Version Control

Change Record:

Version	Date	Author	Comments
1.0	20-01-2024	Komal Madiwal	Architecture

Reviews:

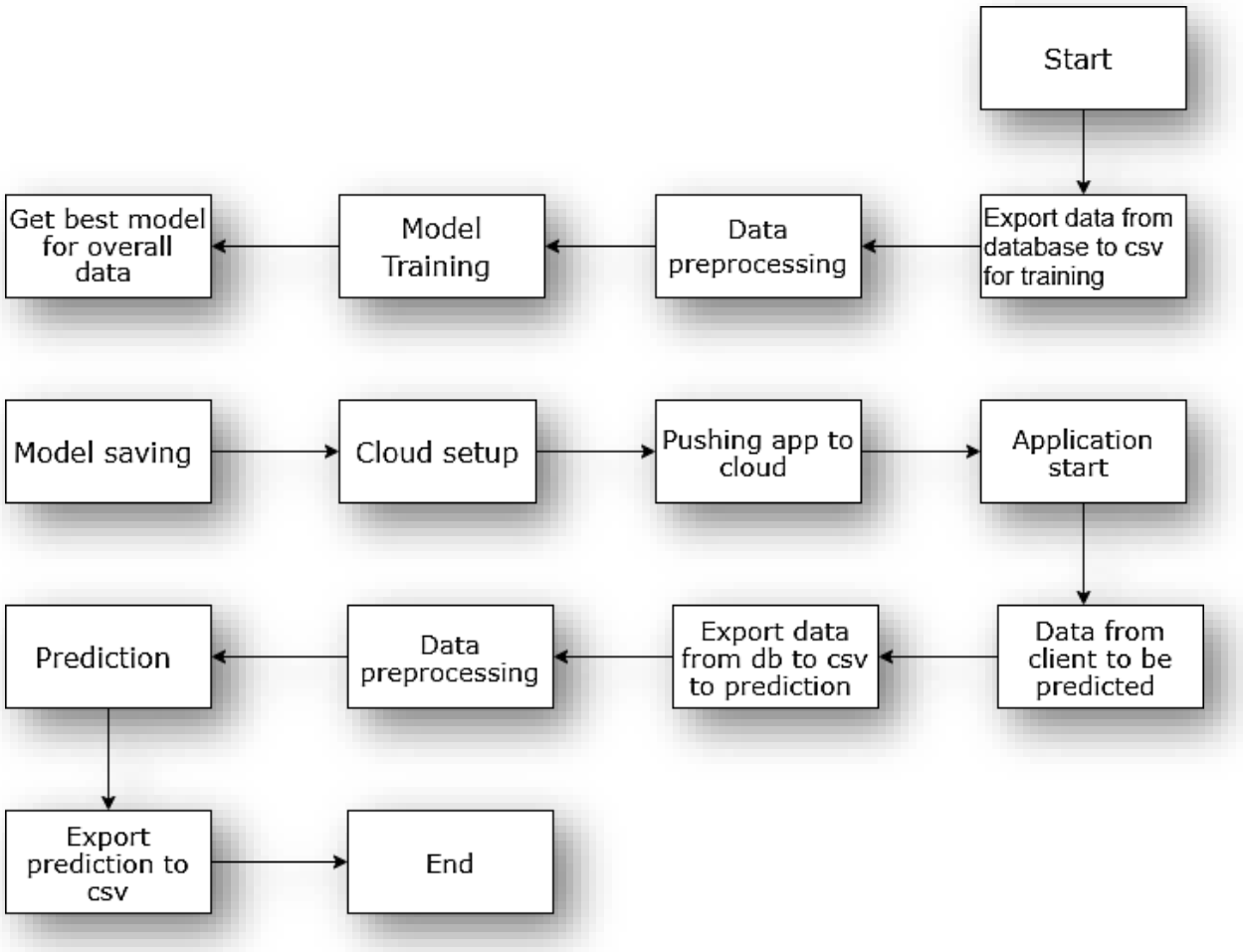
Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1.Architecture



2.Architecture Description

2.1Data 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.

2.2Export Data from database to CSV for Training

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

2.3Data 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 Model Training

Various machine learning models will be trained on the preprocessed data for subsequent use in prediction.

2.5 Get Best Model for Overall Data

Different models will be evaluated, and the best-performing model for the overall dataset will be selected.

2.6 Model Saving

After identifying the best model, we'll save it for future use in prediction.

2.7 Cloud Setup

Cloud setup will be performed for model deployment. Additionally, a Flask app and user interface will be created, integrating the model with the Flask app and UI.

2.8 Push App to Cloud

After local testing, the application will be pushed to the cloud to make it accessible for users.

2.9 Data from Client Side for Prediction

Prediction data from clients will be exported from the database and undergo the same data cleansing process as the training data. This involves data preprocessing using modules created for training data.

2.10 Export Prediction to CSV

Once predictions for client data are complete, the final step is to export the predictions to a CSV file for delivery to the client.

2.11 User Interface

For single user input prediction, we will make a separate UI which will take all inputs from a single user and give back the prediction there only. So that any single user can also use this system for single prediction.