Low Level Design

Stock Price Prediction With Relative Strength Index(RSI)

Written By	Rithin kumar k
Document Version	3
Last Revised Date	05-06-2024

Document Control

Change Record:

Version	Date	Author	comment
0.1	03-06-2024	rithin	Introduction & Architecture defined
0.2	04-06-2024	rithin	Architecture & Architecture Description appended and updated
0.3	05-06-2024	rithin	Unit Test Cases defined and appended

Contents

1. Introduction

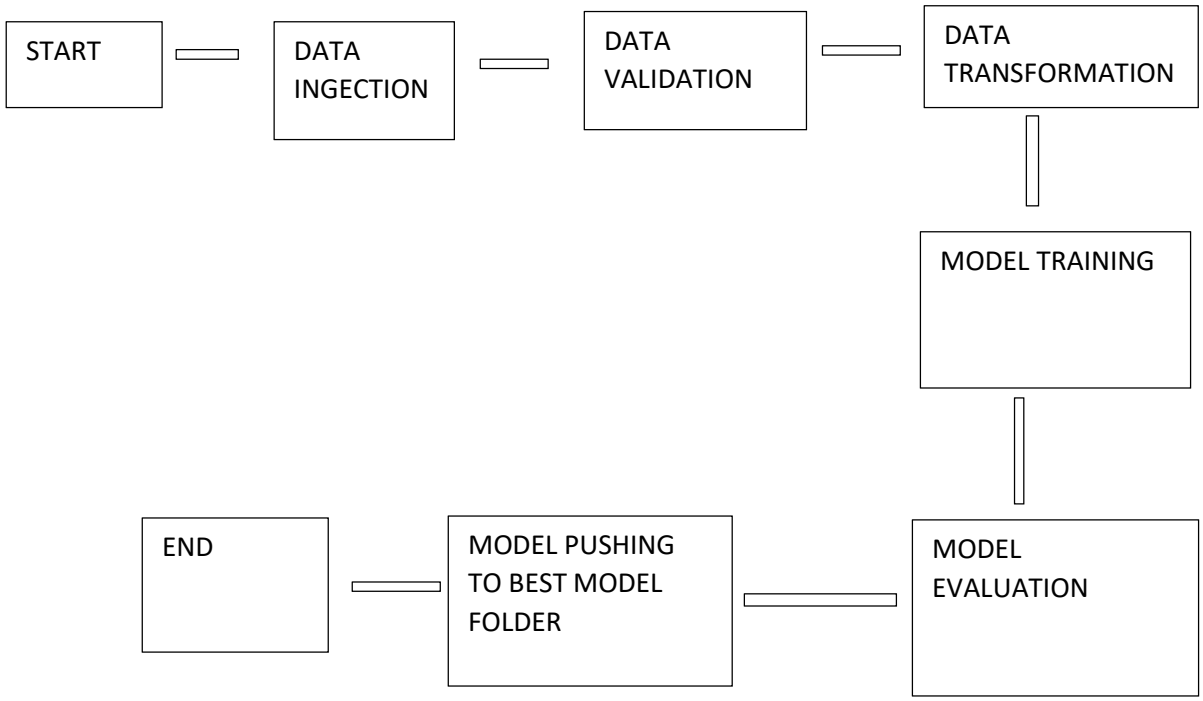
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2 Architecture



3. Architecture Description

3.1. Data Description

Data contains open close, high ,low ,datetime values and data taken from stock broker huge amount of data needed for this problem then only it will work because we need to transform data then we need to filter our condition matching data and sort them.

3.2. Data ingestion

Data injected from Casandra database before that we need to connect with Casandra database and fetch data from table and insert into csv and store inside folder

3.3 . Data validation

Here we are validating our data set is having numerical columns and number of columns present in dataset

3.4. Data transformation

Transform validated data we having only open,close,low,high,datetime values from this we are generate RSI(14), RSI(9), EMA(20) ,EMA(5) values

3.5. Model training

We are using xgboost classifier for prediction and training transformed data on xgboost classifier and trained model saving artifact directory

3.6 Model evaluation

Here evaluating trained model and if this model have good accuracy and compare with previous model accuracy if this model have more accuracy than previous model then insert this model to best model folder

3.7.model pushing

After model evaluation we compare accuracy of current model with previous model and then if trained model has high accuracy than previous model then we push current model to best model folder and remove old model from that folder.

4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should get Submit button to submit the inputs

Verify that output of prediction is displayed after enter submit button	1 application is accessible and user can signup and logged in	User should presented with recommended result when click submit button