Rohit Chitte

Architecture

# Contents

## Dataset Overview

## Logging

## DB operations

## Dataset Overview

Objective of our project is predicting the trend of prices of stocks or indexes based on past data. Knowing the principle of garbage in garbage out, it is extremely important to accumulate data which is reliable and clean. Hence, we have used Yahoo Finance as a source of data for building our project. Prediction is done on past prices hence price data is target column from dataset. Dataset used for prediction is prices data of past 5 years of that stock. As part of future potential requirements we have preserved Date and Volume columns. Data is scraped in .csv file format

E.g

Graphical user interface, text

Description automatically generated

Scraping Source : [Yahoo Finance - Stock Market Live, Quotes, Business & Finance News](https://finance.yahoo.com/)

## Logging

Logging is able to log every activity done by the user.

● The System identifies at what step logging required

● The System is able to log each and every system flow.

● System is never hung even after using so many loggings. Because of logging it became very easy to debug the project. Because by looking at logs we can find where system was interrupted.

## Database

We have efficiently used the functionalities of remote DB to store the data used for training purposes.

Data scraped is efficiently uploaded to remote DB. As DB connection and uploading is time consuming process. We have used threading procedure to reduce the total execution time.

To know the tables uploaded to remote DB. We have introduced mechanisms to store the table names and key space names.

Using keyspace names and table names one can easily fetch the data from remote tables.

Input Schema for uploading data to remote DB is

f"INSERT INTO {keyspace}.{table\_name} (Date,Price,Volume) VALUES{i[**0**]**,** i[**1**]**,** i[**2**]}"

command to fetch data from remote DB

self.session.execute(f"select \* from {keyspace}.{table\_name};").current\_rows

## Exceptional Scenarios.

As we know every project needs to be maintained, so is the case with this project. As Web scraping is used, which is based on target website and its elements. In case of any update from YAHOO FINANCE in website design elements. There is possibility of unexpected exceptions.

## Text Cases.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No.** | **Stock Name** | **Exceution time in (sec)** | **Pass/Fail.** |
| 1 | Bajafinsv | 74.9 | Pass |
| 2. | Tata motors | 120 | Pass |
| 3. | Banknifty | 77.9 | Pass |

## Prediction and training mechanism

Original Data Transformed Data

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Price** | | 110 | | 109 | | 112 | | 113 | | 111 | | 118 | | 108 | | |  |  | | --- | --- | | **X** | **Y** | | 110, 109 | 112 | | 109, 112 | 113 | | 112, 113 | 111 | | 113,111 | 108 | | 111,108 | 108 | |

In this example Y depends on past to prices, but in actual training data and testing data. Y is predicted based on past 100 prices. Next point is reading data using pandas from given path and using data and price column. Load ‘scaler.pkl’ file to load the scaler used during training phase.

Passing this data to scaler to normalize it followed by transforming data to format analogous to above example. Then final data for prediction is prepared and then fed to next function for prediction. In this way time series prediction model is trained.