Low Level Design (LLD)

Stock investment predictions System

Revision Number: 1
Last updated on 11/06/2024

Document Version Control

Date Issued	Version	Description	Author
7th Jan 2024	1.1	First Draft	Sudip Joshi
20th Feb 2024	1.2	Added Workflow chart	Sudip Joshi
20th March 2024	1.3	Added Exception Scenarios Overall, Constraints	Sudip Joshi
21st Apr2024	1.4	Added KPIs	Sudip Joshi
26th May 2024	1.5	Added user I/O flowchart	Sudip Joshi
11th June 2024	1.6	model diagrams	Sudip Joshi

Contents

Abstract

The goal of this project is to develop a system that can automatically sort news articles into predefined categories using Natural Language Processing (NLP) theory & techniques. This project can be beneficial for news aggregators, content management systems, or any platform that deals with large volumes of news articles and needs to organize them efficiently and arrange in structure way.

The ultimate goal of NLP is to enable computers to understand interpret, and respond to human language in a way that is both meaningful and useful. NLP Libraries combines computational linguistics—rule-based modeling of human language—with statistical, machine learning

1 Introduction

1.1 Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the Deep EHR System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

Creating a stock price investment app involves a range of objectives to ensure it meets user needs, provides accurate information, and offers a user-friendly experience. Here are the key objectives for developing such an appEHRs are a vital part of health IT and can:

- Real-Time Stock Prices**:** Provide up-to-date stock prices with minimal latency.

- Market News**:** Integrate news feeds to provide the latest financial news and analysis.

Comprehensive Stock Information

- **Detailed Stock Data:** Offer detailed information about each stock, including historical data, company profiles, financials, and analyst ratings.
- **Sector and Industry Data:** Provide insights into sector and industry performance.
- Portfolio Management

- **Portfolio Tracking:** Allow users to create and manage multiple portfolios, tracking their investments' performance.
- **Investment Analysis:** Offer tools to analyze portfolio performance, including returns, volatility, and risk assessment.
- Research and Education

- **Educational Content:** Provide educational resources such as articles, videos, and tutorials to help users learn about investing and financial markets.
- **Research Tools:** Include screeners and other tools to help users find investment opportunities based on various criteria.

This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

1.2 Scope

Defining the scope of a stock price investment app project involves outlining the features, functionalities, deliverables, and boundaries of the project. This helps ensure clarity and sets expectations for what the project will accomplish. Here's a detailed scope for the stock price investment app: • Integrate APIs to provide real-time stock prices and market data. Include detailed stock information such as historical prices, company profiles, financial reports, and analyst ratings. Provide sector and industry performance data..

1.3 Constraints

We will only be selecting a stock records.

1.4 Risks

As this is a financial data if any prediction is wrong then it will cause loss of money.

1.5 Out of Scope

Old data specific activities, capabilities, and items that are out of scope for the project.

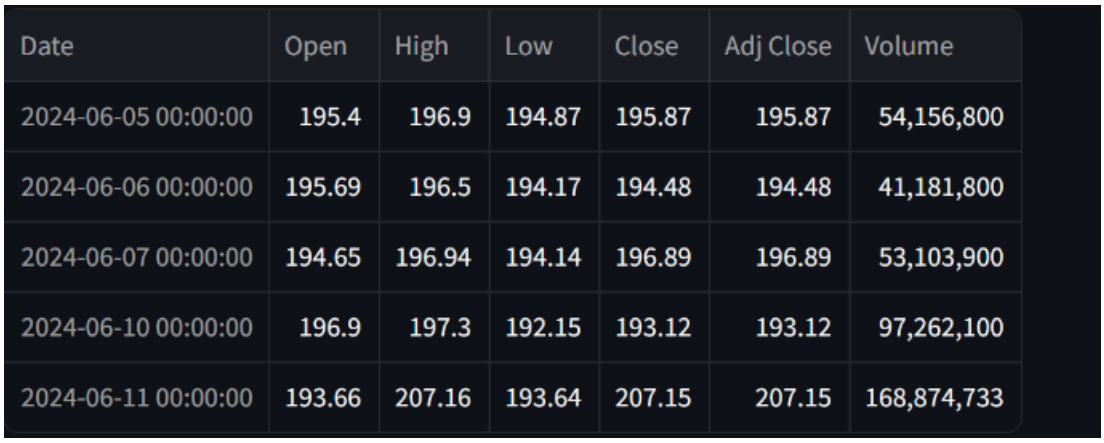
2 Technical specifications

2.1 Dataset

Date	high	Low
2024-06-05 00:00:00	195.4	196.9
2024-06-06 00:00:00	195.69	196.5
2024-06-07 00:00:00	194.65	196.94
2024-06-10 00:00:00	196.9	197.3
2024-06-11 00:00:00	193.66	207.16

2.1.1 Stock dataset overview

Consists of 7 different tables, As follows

- 

Date	Open	High	Low	Close	Adj Close	Volume
2024-06-05 00:00:00	195.4	196.9	194.87	195.87	195.87	54,156,800
2024-06-06 00:00:00	195.69	196.5	194.17	194.48	194.48	41,181,800
2024-06-07 00:00:00	194.65	196.94	194.14	196.89	196.89	53,103,900
2024-06-10 00:00:00	196.9	197.3	192.15	193.12	193.12	97,262,100
2024-06-11 00:00:00	193.66	207.16	193.64	207.15	207.15	168,874,733

2.1.2 Input schema

Feature name	Datatype	Size	Null/Required
Duration (Month)	int	3	Required
Symbol			

2.2 Technical Specifications

Architecture

Overall Architecture:

- **Client-Server Architecture:** The app will use a client-server model, where the client-side (mobile and web) communicates with the server-side to fetch and update data.
- **Microservices Architecture:** The backend will be divided into microservices for modularity, scalability, and maintainability.

2. Frontend

Technologies:

- **Desktop Application**
- **Web:** Streamlit Python framework for app development.

UI/UX Design:

- **Design Tools:** Figma or Adobe XD for designing the user interface and user experience.
- **Components:** Reusable and responsive UI components with libraries like Material-UI or Ant Design.

State Management:

- **Web:** Redux or Context API.

3. Backend

Technologies:

- **Programming Language:** Node.js (JavaScript/TypeScript) or Python (Django/Flask).
- **Frameworks:** Express.js (for Node.js) or Django/Flask (for Python).

Database:

- **Primary Database:** PostgreSQL or MySQL for relational data.
- **NoSQL Database:** MongoDB for unstructured data.

Front end Development:

- **Streamlit framework**

4. Data Integration

APIs for Real-Time Data:

- **Financial Data Provider:** Alpha Vantage, IEX Cloud, or Yahoo Finance API for real-time stock prices and market data.
- **News API:** Integration with news APIs like NewsAPI.org for market news.

5. Security

Authentication and Authorization:

- **User Authentication:** JWT (JSON Web Tokens) for secure user authentication.
- **Authorization:** Role-based access control (RBAC).

Data Protection:

- **Encryption:** TLS/SSL for data in transit, AES-256 for data at rest.
- **Secure Storage:** Use secure methods for storing sensitive data, such as encrypted databases.

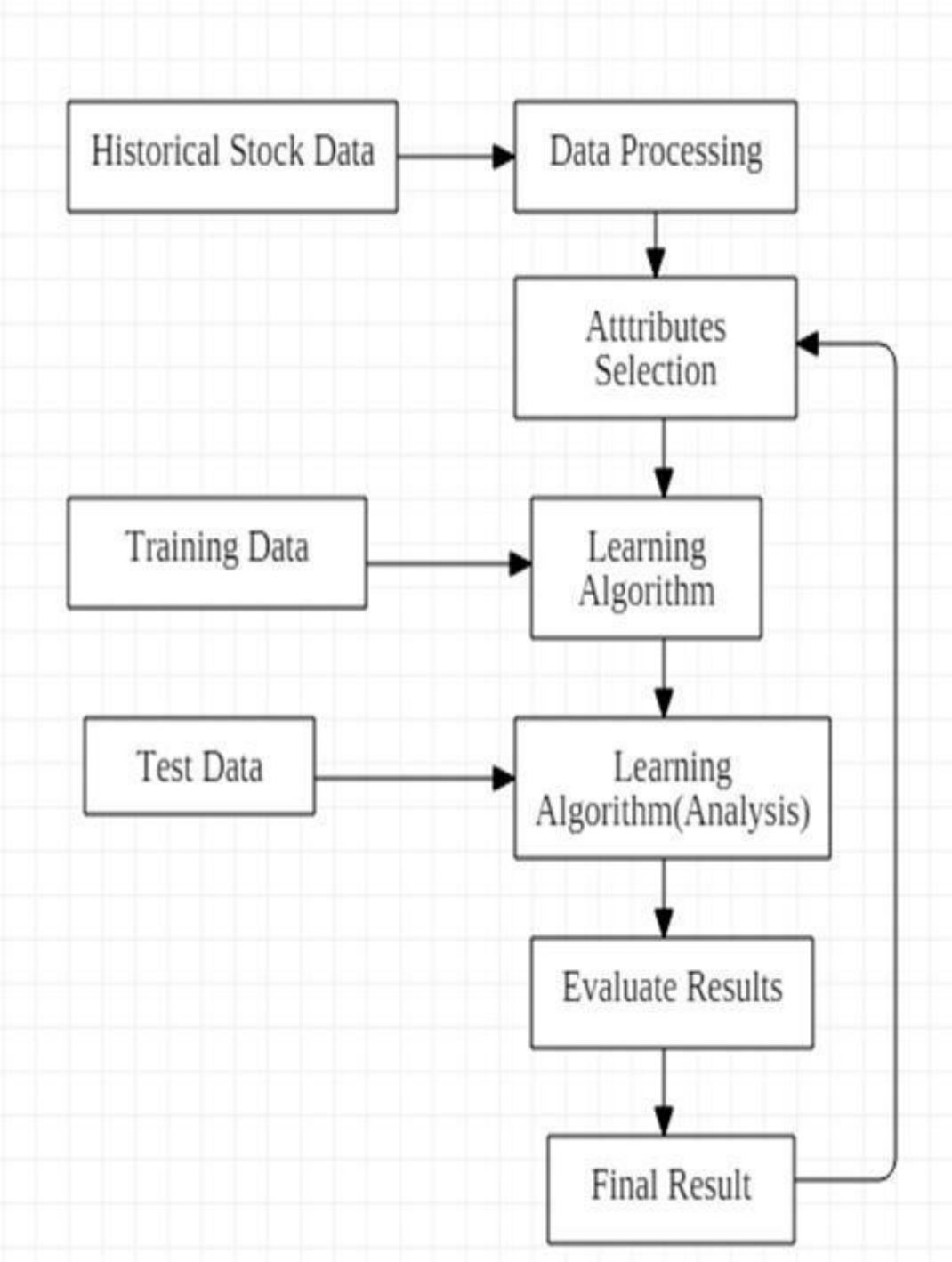
3 Technology stack

Front End	Streamlit
Backend	Python
Database	MongoDB/MySql/ Source :Kaagle

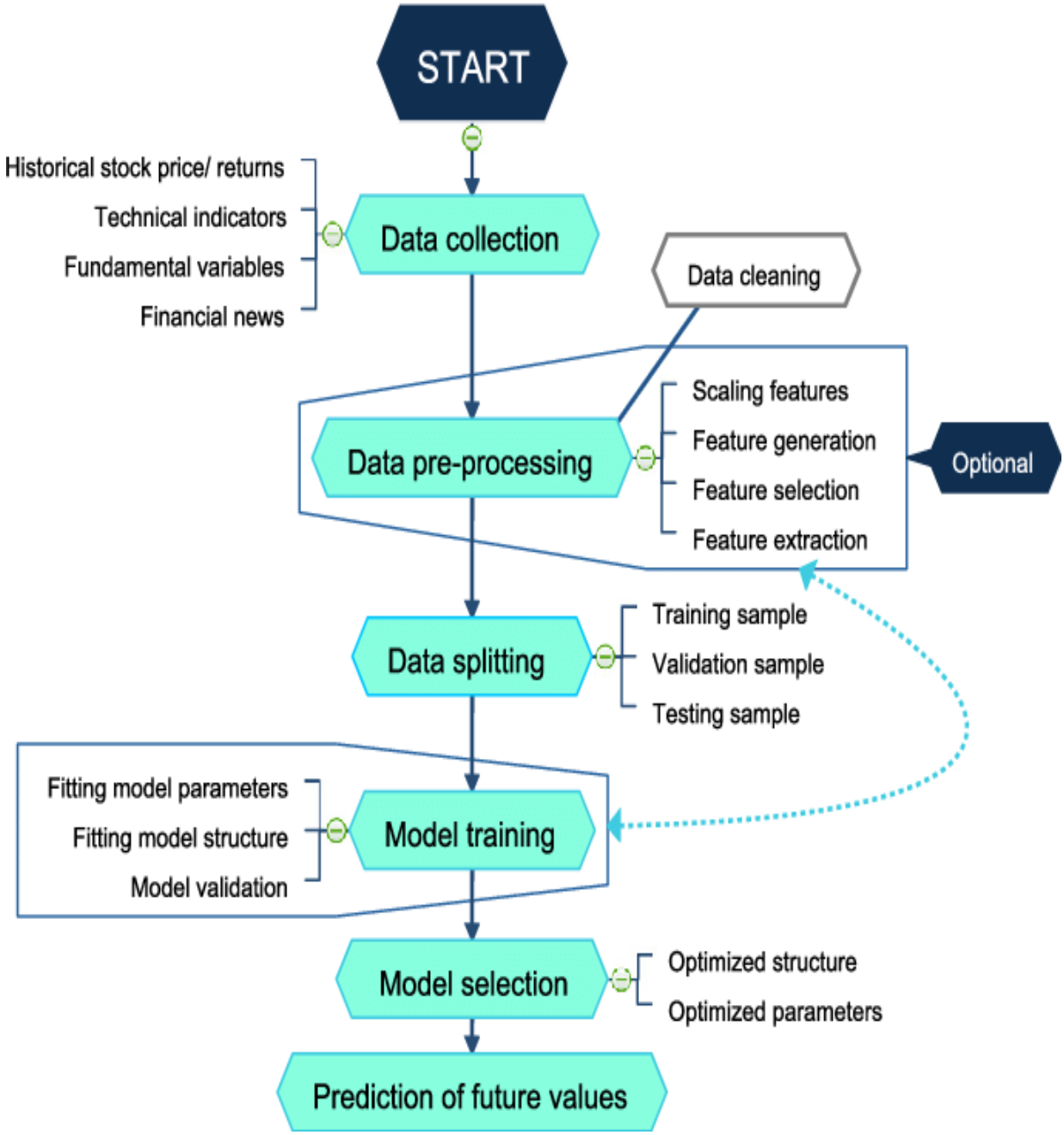
4 Proposed Solution

To develop a comprehensive stock price investment app, we will follow a structured approach that encompasses all aspects of the project, from initial planning and design to development, deployment, and maintenance. Here's a detailed plan outlining the proposed solution:

5 Model training/validation workflow



6 User I/O workflow


START
Historical stock price/ returns
Technical indicators
Fundamental variables
Financial news
Data collection
Data cleaning
Scaling features
Feature generation
Feature selection
Feature extraction
Optional
Data pre-processing
Training sample
Validation sample
Testing sample
Data splitting
Fitting model parameters
Fitting model structure
Model validation
Model training
Optimized structure
Optimized parameters
Model selection
Prediction of future values

7 Exceptional scenarios

Step	Exception	Mitigation	Module
8th Jan 2024	1.1	First Draft	Sudip Joshi
20th May 2024	1.2	Added Workflow chart	Sudip Joshi

8 Test cases

Test case	Steps to perform test case	Module	Pass/Fail

9 Key performance indicators (KPI)

Daily Active Users (DAU):

- Description: The number of unique users who interact with the app on a daily basis.
- Objective: Measure the app's stickiness and user retention.

Monthly Active Users (MAU):

- Description: The number of unique users who interact with the app on a monthly basis.
- Objective**:** Track overall user engagement and growth trends.

Session Duration:

- Description: The average amount of time users spend in the app per session.
- Objective: Assess how engaging and useful the app content is for users..