Moving Averages

The 100-day and 200-day moving averages are common technical indicators used in financial analysis to smooth out price data and identify trends over a specified period of time. Here's how they are calculated:

1. 100-Day Moving Average (100-day MA):

The 100-day moving average is calculated by taking the average of a stock's closing prices over the last 100 trading days.

Mathematically, it is computed as the sum of the closing prices for the past 100 days divided by 100.

Formula:

100-day MA = (∑Closing Pirces for Last 100Days)/100

Purpose:

The 100-day moving average is used to smooth out short-term fluctuations in a stock's price. It provides a longer-term view of the stock's trend.

2. 200-Day Moving Average (200-day MA):

The 200-day moving average is calculated by taking the average of a stock's closing prices over the last 200 trading days.

Mathematically, it is computed as the sum of the closing prices for the past 200 days divided by 200.

Formula:

200-day MA = (∑Closing Prices for Last 200 Days)/200

Purpose:

The 200-day moving average is an even longer-term indicator. It provides a smoothed view of a stock's trend over a more extended period compared to the 100-day moving average.

Data

This data is used to train the LSTM Machine Learning Algorithm. The data is collected from yahoo finance website using yfinance library.

The data contains 6 attributes those are Open, High, Low, Close, Adj Close, Volume. Here I have used only the Close column to predict the stock prices

1. Setting Start and End Dates:

startdate and enddate are defined to specify the time range for which the stock data will be retrieved. In this case, it's from January 1, 2010, to December 31, 2019.

2. User Input for Stock Ticker:

user_input = st.text_input('Enter Stock Ticker') prompts the user to input a stock ticker.

3. Fetching Stock Data:

data = pdr.get_data_yahoo(user_input, start=startdate, end=enddate) attempts to retrieve historical stock data for the entered stock ticker within the specified date range.

4. Describing Data:

If data is successfully fetched, it displays summary statistics using st.write(data.describe()).

5. Displaying Data:

The training and testing data are displayed side by side using st.columns(2) for a two-column layout.

In the left column, it shows the training data and in the right column, it shows the testing data.

Comparison

In this module we can compare the stock prices of two dates of any company that are listed on yahoo finance website.

1. User Input for Stock Ticker and Dates:

st.subheader('Comparison between two specific dates') sets a subheader for the user input section.

user_input = st.text_input('Enter Stock Ticker') prompts the user to input a stock ticker.

startdate = st.date_input("Enter start date (YYYY-MM-DD): ", min_value=min_date, max_value=max_date) and enddate = st.date_input("Enter end date (YYYY-MM-DD): ", min_value=min_date, max_value=max_date) allow the user to input start and end dates for the stock price analysis.

2. Fetching Stock Price Data:

The get_data() function is defined to fetch stock data for the specified stock ticker and date range.

It uses the yf.download() function to retrieve the data and extracts the closing prices.

3. Displaying Stock Price Information:

It displays the closing price of the specified stock on the entered start and end dates.

It also provides textual information about the price difference.

4. Fetching and Plotting Stock Prices:

If valid user input is provided, it attempts to fetch historical stock data using pdr.get_data_yahoo().

If data is successfully retrieved, it plots the closing prices using st.line_chart(data.Close).

Predict Future Prices

In this module we can predict two days stock prices in advance and plot the interactive graph using st.plotly_chart(fig).

It utilizes key libraries like streamlit, yfinance, pandas, matplotlib, and plotly.graph_objects to create an interactive interface. Users begin by inputting a stock ticker. The app fetches historical data from Yahoo Finance, focusing on the period from January 1, 2023, to the present date. It then calculates essential indicators: the 100-day moving average (100-day MA) and the 200-day moving average (200-day MA). These smoothed averages provide valuable insights into the stock's trend over their respective time frames.

The app also displays the acquired stock data, along with the calculated moving averages. This visual representation gives users a clear overview of the stock's historical performance. Additionally, it predicts the stock prices for the next two days based on the latest data. These projections are presented in a concise tabular format, with dates aligned to the corresponding prices. The app enhances user experience through interactive plotting with Plotly. It creates a dynamic line chart showcasing the closing prices, 100-day MA, and 200-day MA. This graphical representation offers a more intuitive understanding of the stock's behavior. In case of any unexpected errors, the app handles them gracefully by providing an informative error message. Overall, this Streamlit app provides a user-friendly platform for exploring and predicting future stock prices, offering valuable insights for investors and traders.

Stock News

It displays the top 10 stock news from yahoo finance. The news is webscrapped from the yahoo finance website using BeautifulSoap library.

1. Web Scraping:

The code starts by attempting to scrape the latest stock news from Yahoo Finance. It sends a GET request to the Yahoo Finance URL and stores the response.

2. Parsing HTML Content:

The HTML content of the response is then parsed using BeautifulSoup. This allows the code to easily navigate and extract specific elements from the webpage.

3. Finding News Headlines:

The code uses BeautifulSoup to find all the <h3> elements with the class 'Mb(5px)'. These elements typically contain the headlines of the latest news articles on Yahoo Finance.

4. Displaying News Headlines:

After extracting the news headlines, the code uses Streamlit to display them. It first prints the heading "Top Stock News" using st.write("## Top Stock News"). Then, it iterates through the list of headlines and displays them with their corresponding index.