Stock Trading Prediction and Analysis Using AI/ML Techniques

1. Introduction

In the ever-evolving world of finance, the ability to predict stock market trends and make informed trading decisions is of paramount importance. This project explores the use of Artificial Intelligence (AI) and Machine Learning (ML) techniques to analyze and predict stock price movements. Specifically, we focus on historical stock data of Apple Inc. (AAPL) from January 1, 2020, to January 1, 2023. By leveraging financial indicators such as the 50-day Moving Average (MA) and the 14-day Relative Strength Index (RSI), we aim to identify potential trading opportunities and provide insights into stock price trends.

2. Data Acquisition

The first step in our analysis involves acquiring historical stock data. We utilize the yfinance library to download data for Apple Inc. (AAPL) over a specified period.

Libraries

- import yfinance as yf: Imports the yfinance library, which is used to download historical market data from Yahoo Finance.
- import pandas as pd: Imports the pandas library, used for data manipulation and analysis.
- import matplotlib.pyplot as plt: Imports the pyplot module from the matplotlib library, which is used for plotting and visualizing data.

3. Data Processing

To facilitate analysis, we calculate two key financial indicators: the 50-day Moving Average (MA50) and the 14-day Relative Strength Index (RSI).

50-Day Moving Average (MA50): The 50-day Moving Average smooths out price data to identify trends over a specified period. It is calculated by averaging the closing prices over the last 50 days.

data = yf.download('AAPL', start='2020-01-01', end='2023-01-01') :

Relative Strength Index (RSI):

The RSI is a momentum oscillator that measures the speed and change of price movements. It oscillates between 0 and 100, with typical overbought and oversold thresholds at 70 and 30, respectively.

```
# Define the RSI computation function
def compute_rsi(data, window):
    diff = data.diff(1).dropna()  # Get the difference in price from the previous day
```

```
    gain = (diff.where(diff > 0, 0)).rolling(window=window).mean()  # Calculate the rolling average of gains
    loss = (-diff.where(diff < 0, 0)).rolling(window=window).mean()  # Calculate the rolling average of losses
    rs = gain / loss  # Relative strength
    rsi = 100 - (100 / (1 + rs))  # Relative strength index
    return rsi

# Calculate the 14-day RSI
data['RSI'] = compute_rsi(data['Close'], 14)
```

4. Data Visualization

To visualize the data and the calculated indicators, we create subplots using the matplotlib library. The first subplot displays the closing prices along with the 50-day Moving Average, and the second subplot shows the 14-day RSI.

```
# Create subplots for closing prices and RSI
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(14, 14), sharex=True)

# Plot the closing prices and 50-day moving average on the first subplot
ax1.plot(data['Close'], label='Closing Price')
ax1.plot(data['MA50'], label='50-Day Moving Average', color='orange')
ax1.set_title('AAPL Closing Prices and 50-Day Moving Average (2020-2023)')
ax1.set_ylabel('Price')
ax1.legend()

# Plot the RSI on the second subplot
ax2.plot(data['RSI'], label='14-Day RSI', color='purple')
ax2.axhline(30, linestyle='--', alpha=0.5, color='red')
ax2.axhline(70, linestyle='--', alpha=0.5, color='green')
```

```
ax2.set_title('AAPL 14-Day RSI (2020-2023)')

ax2.set_xlabel('Date')

ax2.set_ylabel('RSI')

ax2.legend()


# Show the plots

plt.show()
```

Explanation of Visualization:

The first subplot displays the closing prices (blue line) and the 50-day moving average (orange line). The moving average helps to identify the overall trend of the stock price.

The second subplot shows the 14-day RSI (purple line) with two horizontal dashed lines at the 30 and 70 levels, indicating oversold and overbought conditions, respectively.

5. Analysis and Insights

By examining the visualizations:

- **MA50 Analysis:** The 50-day moving average provides a smoothed trend line of the stock price, allowing for better identification of trends. When the stock price crosses above the MA50, it may indicate a potential buy signal, and when it crosses below, it may indicate a potential sell signal.
- **RSI Analysis:** The RSI helps in identifying overbought and oversold conditions. When the RSI is above 70, the stock is considered overbought, and a price correction may be expected. When the RSI is below 30, the stock is considered oversold, indicating a potential buying opportunity.

6. Scope in Future

Future work can involve:

1. **Incorporating Additional Indicators:** Including more technical indicators like Moving Average Convergence Divergence (MACD), Bollinger Bands, and others to enhance the analysis.
2. **Machine Learning Models:** Applying machine learning models such as Random Forest, Support Vector Machines, or neural networks to predict stock price movements.
3. **Backtesting Trading Strategies:** Implementing and backtesting various trading strategies to evaluate their effectiveness using historical data.
4. **Real-Time Data Analysis:** Extending the analysis to real-time data to make live trading decisions.

This project serves as a stepping stone towards a comprehensive AI/ML-based trading system, providing a solid foundation for further exploration and development.

7. Conclusion

This project demonstrates the application of AI/ML techniques to stock trading prediction and analysis. By leveraging historical stock data and key financial indicators such as the 50-day Moving Average and the 14-day RSI, we can gain valuable insights into stock price trends and potential trading opportunities. While this analysis provides a foundational understanding, further enhancements using more sophisticated machine learning models and additional financial indicators can improve prediction accuracy and trading strategies.