Stock Market Prediction

Using Dynamic Programming and Graphs

A Deep Dive into Efficient Stock Forecasting Methods

$5 Trillion

is traded in global stock markets daily, yet 80% of retail investors lose money due to poor timing and lack of predictive insights.

Problem Statement !

Stock Market Challenges:

- Highly volatile price fluctuations
- Traditional models fail in real-time predictions

Objective:

- Efficiently predict stock prices
- Identify optimal buy/sell points
- Maximize profit using Dynamic Programming (DP)
- Leverage Graph Theory for trend analysis


Market
Relevance &
Feasibility

01

Why is this important?

- Global stock market size: **$100+ Trillion**
- Over **50 million** active retail investors in India.
- AI-driven trading is growing at **40% CAGR**

02

Existing Methods & Their Issues

- Technical Analysis → Can be unreliable
- Machine Learning → Needs large training data

03

Who Benefits?

- Retail traders, hedge funds, financial analysts

Case Study

Tesla (TSLA) Stock Analysis

Stock Selected

Tesla Inc. (TSLA)

Time Period

Last 6 months

Observations

Price fluctuations due to market trends, earnings, macroeconomic factors

Algorithmic Approach

Dynamic Programming (DP)

01 Concept:

- Breaks problem into subproblems
- Uses memorization to prevent redundant calculations

02 Application in Stock Prediction:

- Maximizes profit from historical price patterns

Dynamic Programming (DP)

01 The Problem:

- ❑ Given a list of stock prices, find the best time to buy and sell to maximize profit.

02 Why Brute Force Fails?:

- ❑ Checking all buy-sell pairs is too slow ($O(n^2)$) and inefficient for large datasets.

03 Optimized DP Approach O(n):

- ❑ Track **lowest price so far** (best time to buy).
- ❑ Calculate **profit at each step** (if selling today).
- ❑ Update **max profit** whenever a better profit is found.

04 Example:

- ❑ Stock Prices: **[720, 730, 715, 690, 700, 750, 770, 760, 780, 790]**
  ✓ **Buy at 690, Sell at 790 → Max Profit = 100**

Graph Theory

01 Concept:

- Stocks as a directed weighted graph

02 Application in Stock Prediction:

- Uses shortest path algorithms (Bellman-Ford) to find optimal price movements

Dynamic Programming (DP)

01 What is Graph Theory?:

A **graph** models relationships using:

- **Nodes (Vertices):** Stock prices at different times.
- **Edges:** Price transitions over time, often with probabilities.

02 How It Helps in Prediction:

- **Captures Trends:** Analyzes price movements as a network.
- **Forecasting:** Uses historical data to predict future transitions.
- **Identifies Patterns:** Detects cycles and trends in stock behavior.

03 Example:

- **Nodes:** Prices (e.g., $720 → $730 → $715).
- **Edges:** Probability of moving between prices based on past trends.

Graph Representation

- **Nodes:** Stock prices at different times
- **Edges:** Price transitions with probability weights

Dynamic Programming Approach

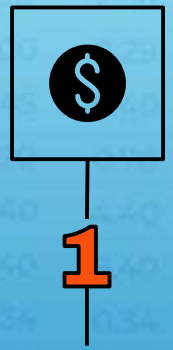
For Profit Maximization

Problem Statement:	Given an array of stock prices, maximize profit by choosing **buy/sell days.**
Example:	Prices: [720, 730, 715, 690, 700, 750, 770, 760, 780, 790]
	Optimal buy: Day 4 at $690
	Optimal sell: Day 10 at $790
	Max Profit: $100
Recurrence Relation:	dp[i]=max(dp[i−1],price[i]−min_price)

Performance Comparison

Algorithm Approach	Time Complexity	Space Complexity	Best Use Case
Dynamic Programming	O(n)	O(1)	Profit optimization
Bellman-Ford (Graph)	O(V × E)	O(V + E)	Trend prediction
Dijkstra's Algorithm	O((V + E) log V)	O(V + E)	Faster but no negative weights

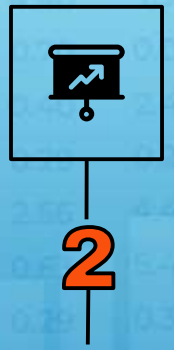
Real-World Applications



1

Retail investors

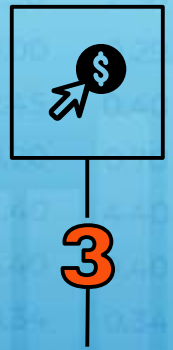
Maximize small-scale trading profits



2

Hedge funds

Improve portfolio risk management



3

Trading Firms

Automate high-frequency trading

"The stock market is a device for transferring money from the impatient to the patient"

— Warren Buffett

Thanks!

- ✓ Be patient. 🧘‍♂️
- 🎯 Be strategic. 📊
- 🔍 Predict wisely. 📈🤖
- 💰 Profit massively. 🚀💸