EQUITY TRADING USING DEEP REINFORCEMENT LEARNING

Author : Ashok Jana

Introduction

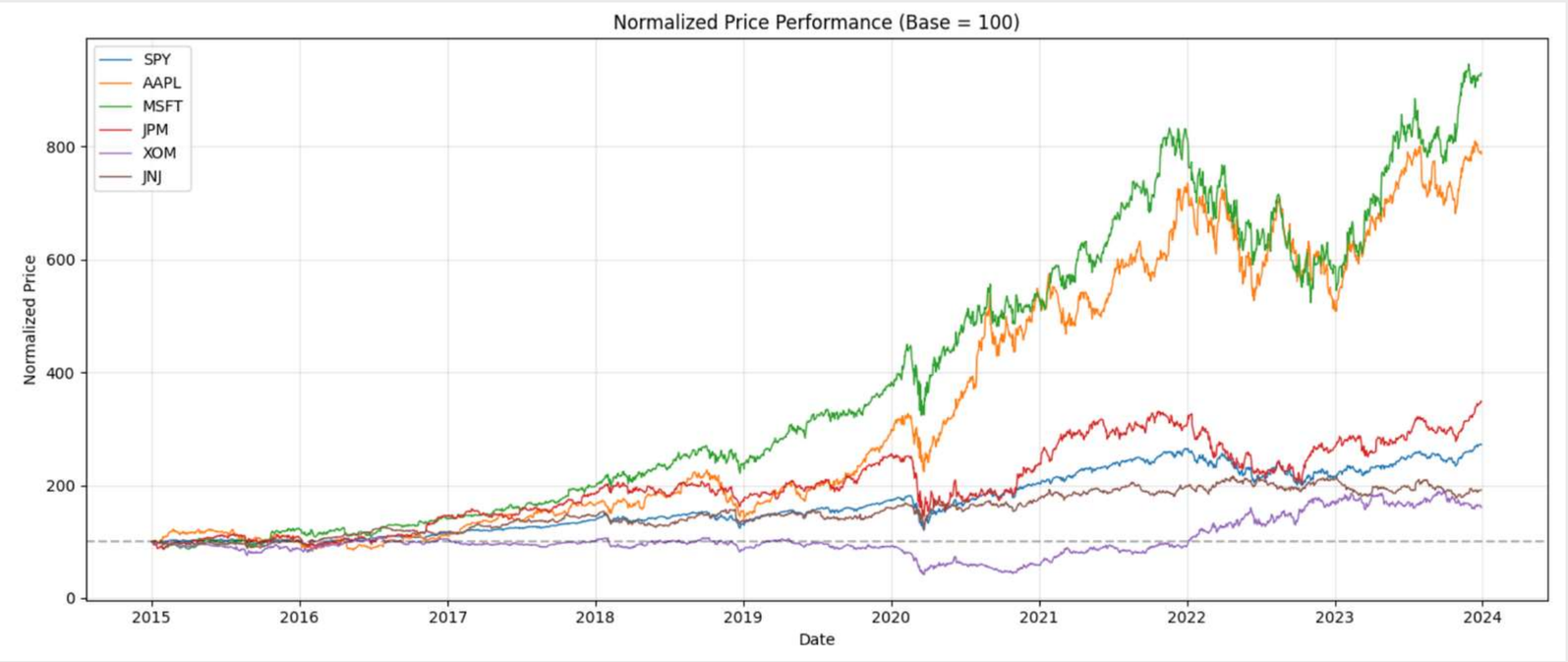
- Stock markets go up, down, and sometimes crash.
- Most people panic during crashes.
- We wanted an intelligent system that stays calm and adapts.
- The focus was smart decision-making, not guessing.

Problem Statement : How can automated trading agent learn to adopt different strategy across different market conditions

DATASET & ASSETS USED

- Downloaded the dataset from yahoo finance
- Assets: SPY, AAPL, MSFT, JPM, XOM, JNJ.
- Includes Bull Market, COVID Crash (2020) and Recovery period.

CLEANING THE DATA



- Make sure all stocks had matching dates to ensure all the dates are aligned properly
- Removed missing or incorrect entries.
- Normalized the price to understand which assert has highest market cap

FEATURE ENGINEERING

- I developed 27 features from raw OHLCV data including momentum, trend, volatility, and volume indicators .
- These features help the AI to understand market mood.

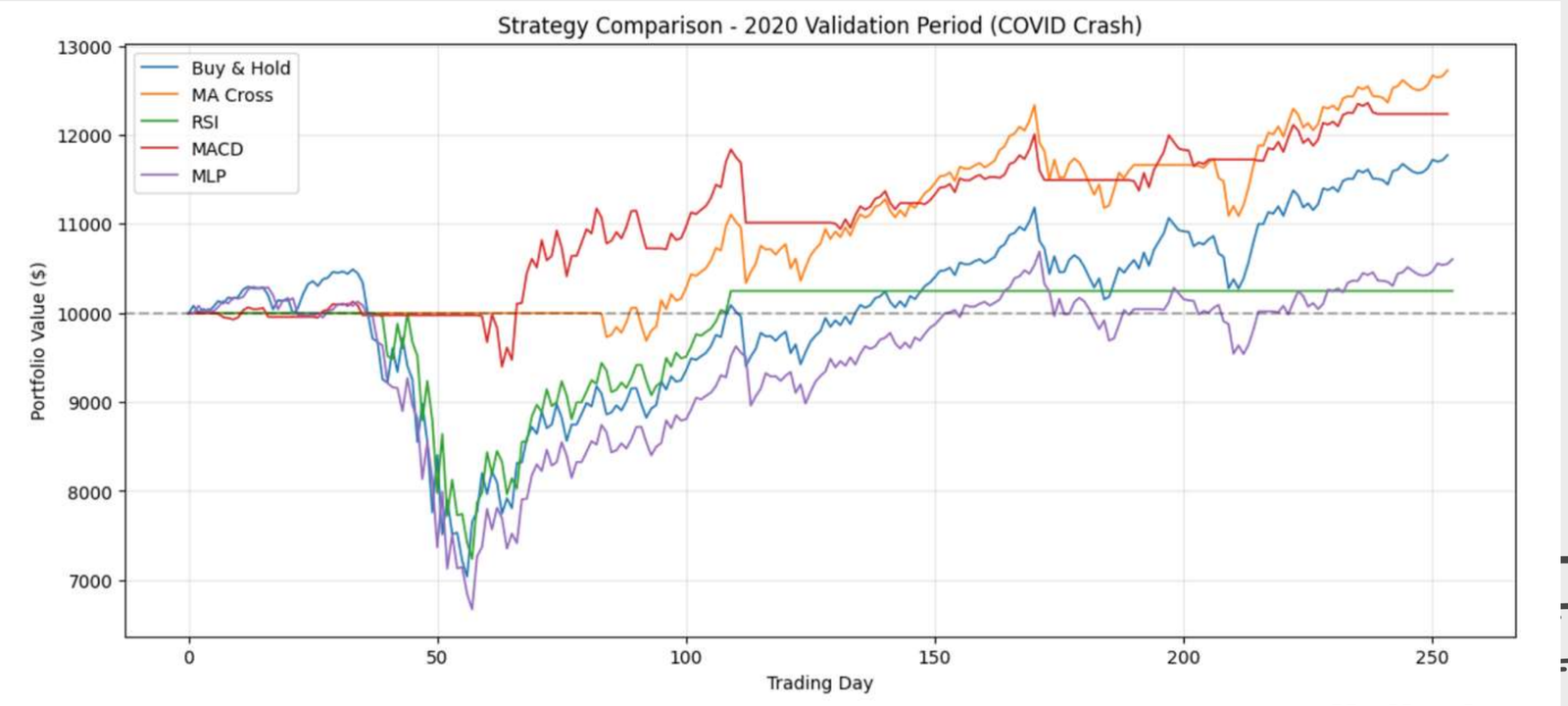
TRADING ENVIRONMENT

- Created a virtual stock market environment.
- Starting money: $10,000.
- The AI could: Buy, Sell or Hold.
- Each trade had a small fee (0.1%) like real markets.
- If portfolio value increases → reward.
- If it loses money → penalty.
- Over time, it learns which actions work best.

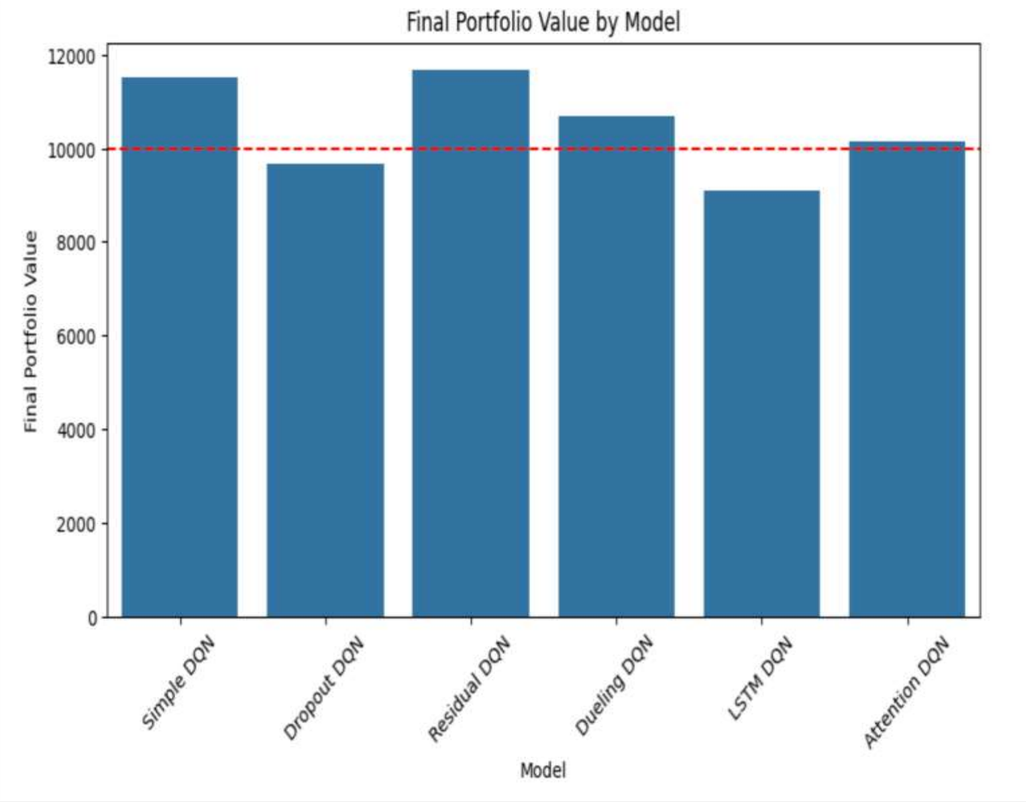
TESTING STRATEGY FINDINGS

Before AI, tested normal strategies:

- Buy & Hold - good money but suffers during crashes.
- Moving Average strategy - reduces crash damage
- RSI and MACD strategies suffering sudden crashes
- Random trading - loses money due to fees.



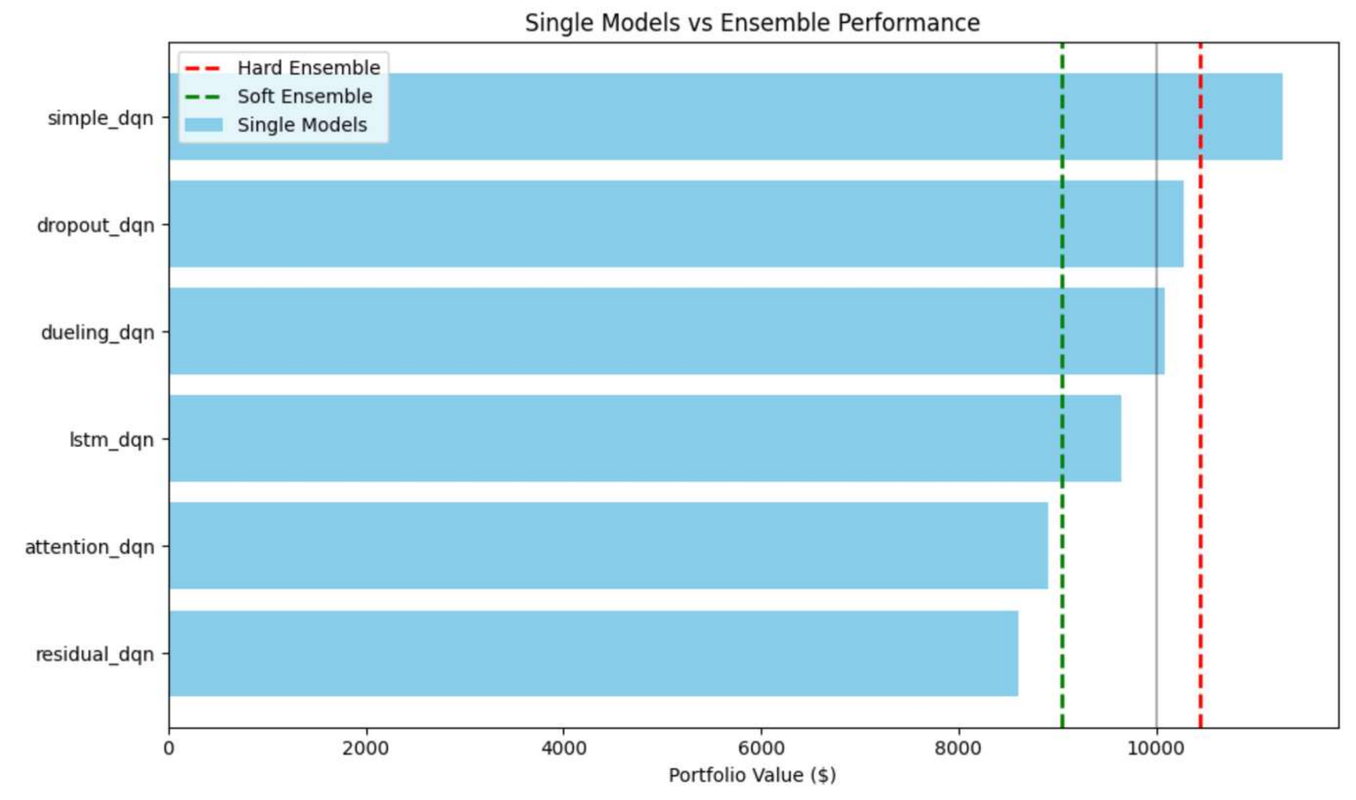
MODEL BUILDING



Trained 6 different AI models.

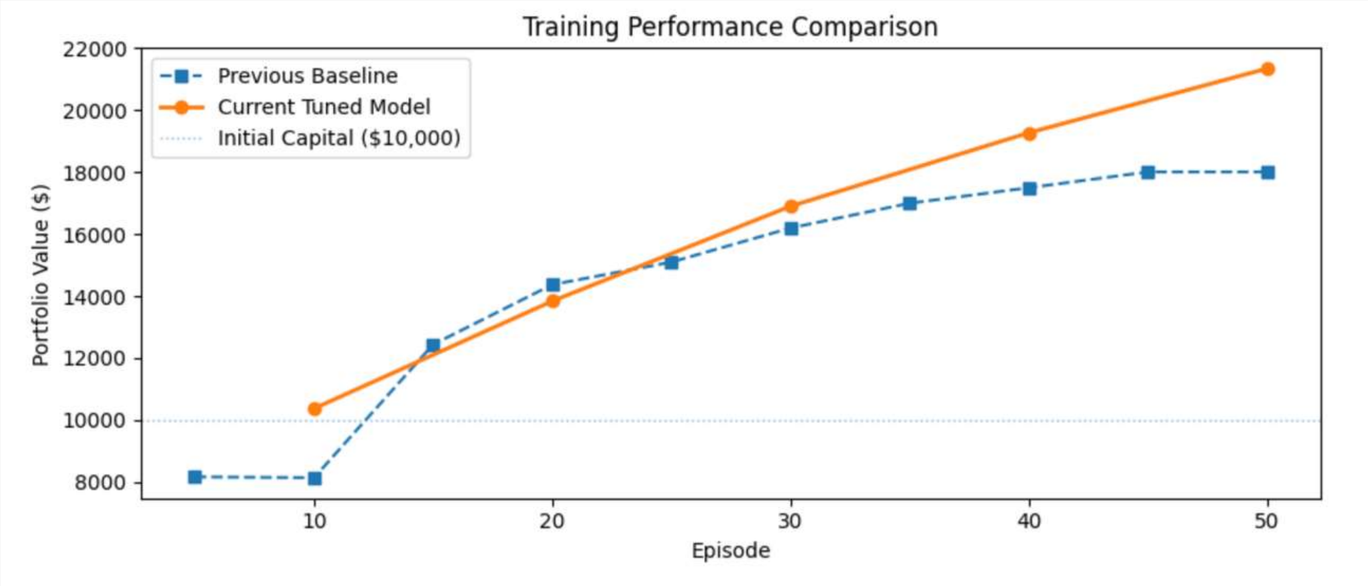
- Simple DQN
- Dropout DQN
- Residual DQN
- Dueling DQN
- LSTM DQN
- Attention DQN
  - At first, the model acted randomly having more loses
  - Slowly it started finding better buy/sell timing.
  - Simple DQN ,and Residual DQN models performed well in training.

ENSEMBLE TRAINING



- Hard Voting: Majority decision wins
- Soft Voting : Averages model confidence scores
- Stacking: Another model decides which to trust.

HYPERPARAMETER TUNING



Hyperparameter tuning performed on Simple DQN Model by adjusting learning rate 0.001 to 0.0001 , epsilon delay and batch size

Training Value improved:
$18,010 → $21,347.

Validation improved:
$11,277 → $12,641 (+12%).

ENSEMBLE OPTIMIZATION

- In ensemble optimization I weighted top-2 ensemble I selected **Simple DQN and Dropout DQN** based on validation portfolio performance. Their Q-values were combined using weighted averaging, where the stronger model was given a higher weight. The final action was selected using the highest combined Q-value, and this ensemble strategy improved the validation portfolio value to **$12,589**, outperforming individual models

CONCLUSION

- Successfully built an intelligent trading system.
- Hyperparameter tuning gave biggest improvement.
- Best result achieved using a weighted AI combination.
- The project proves AI can learn disciplined trading behavior.

THANK YOU