Maximizing Investment Returns with Deep Reinforcement Learning for Stock Trading.

Mid-Term Report

Group → G41

Mentor name → Tejas Gatthoria

Team Members

1. Lakshay Bhadana 210020068
2. Sujeet Mehta 21D070076

How to analyze and predict stock prices?

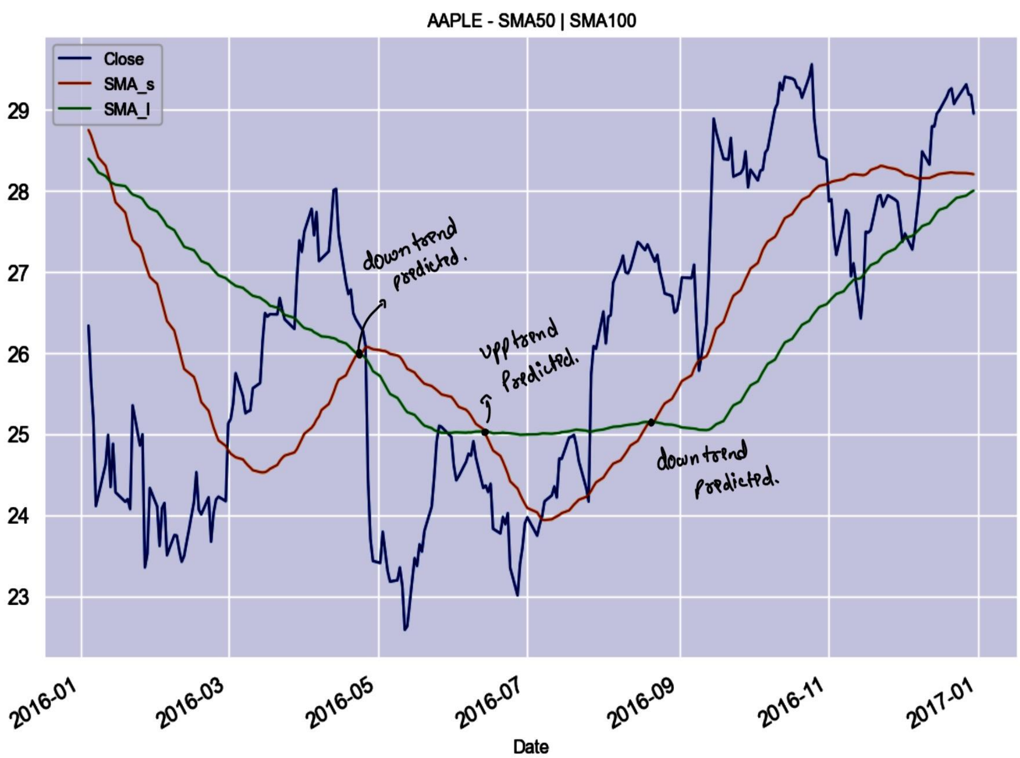
= There exists two main approaches to analyze and predict stock prices which are **technical analysis** and **fundamental analysis** . The technical analysis looks into the past data of the market only to predict the future. On the other hand, the fundamental analysis takes into account other information such as the economic status, news, financial reports, meeting notes of the discussion between CEOs.

Before we explore RL-based algorithms to optimize stock prices, let's first take a closer look at traditional algorithms.

Traditional Algorithms.

1.**Simple Moving Average**

= Simple moving averages (SMAs) use a simple arithmetic average of prices over some timespan.By calculating the moving average, the impacts of random, short-term fluctuations on the price of a stock over a specified time frame are mitigated.The longer the period for the moving average, the greater the lag .("lag" refers to the delay or slow responsiveness).



Upward momentum is confirmed with a **bullish crossover**, which occurs when a short-term moving average crosses above a longer-term moving average. Conversely, **downward momentum** is confirmed with a **bearish crossover**, which occurs when a short-term moving average crosses below a longer-term moving average.

2.**Mean-Variance optimisation**

= This Classic portfolio optimization technique aims to maximize the expected return of a Portfolio while minimizing its Variance. It involves optimizing the allocation of assets to achieve the desired risk-reward-trade-off.

The expected return is a probability expressing the estimated return of the investment in the security.

MEAN → RETURN
&
VARIANCE → RISK

3.**Risk Parity**

= The goal of risk parity investing is to earn the optimal level of return at the targeted risk level.This strategy can perform well in challenging & unpredictable market conditions.

4. **Genetic Algorithm**

= Genetic algorithms are a form of a brute-force type attack to the problem. The only differences being using the concept of evolution to speed up the process. Here is how the genetic algorithm works.

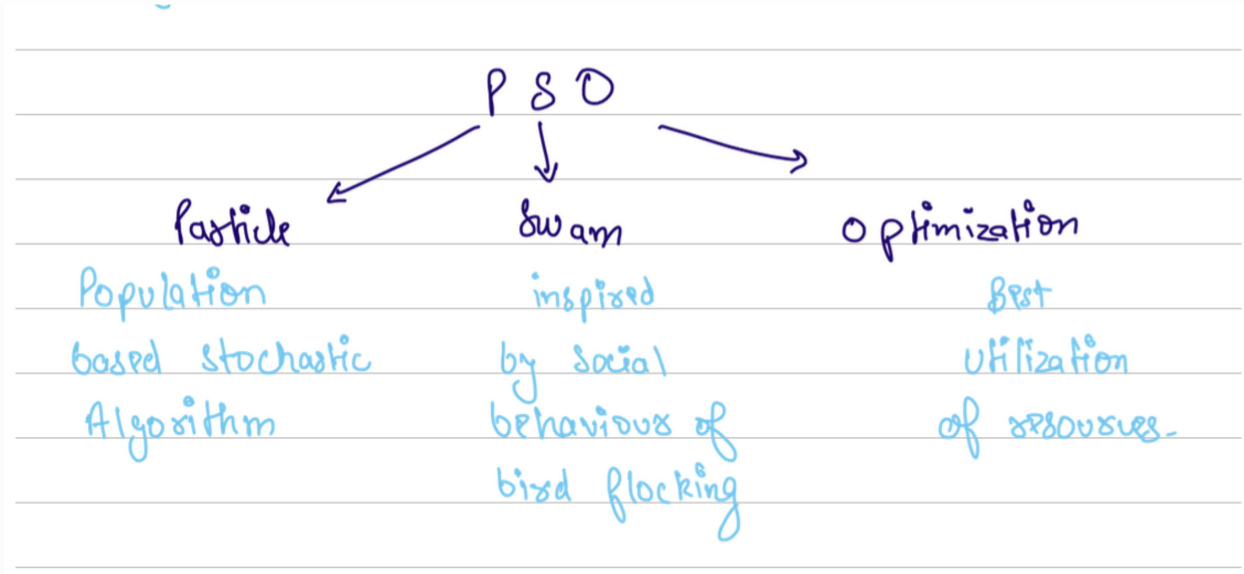
where not to used Genetic Algorithm :-

For elementary tasks such as regression and classification, a deep neural network,trained using a gradient-based optimizer will always converge faster than a genetic algorithm.

Genetic algorithm are not good for finding patterns, but rather having a wide scope of view, to find solutions that have not been considered

5.**Particle Swarm Optimization**

= Based on the application of neural networks (NN) and swarm intelligence technologies and is used to generate one-step ahead investment decisions.



ADVANTAGES OF TRADITIONAL ALGORITHMS

So , Having a lot of traditional Algorithms Why do we need RL based algorithms?
The reason is that the traditional statistical learning algorithms can not cope with the non-stationary and non-linearity of the stock markets.
The main working of traditional based Algorithms lies on the fact that we use the historical-data and on the basis of that we are building our strategies.

1.**Use of Available data.**
= In the Supervised and Unsupervised machine learning algorithm we have a huge amount of data and using this data they can produce data output from previous experience.

2 **Specific Target Function**
= In stock price optimization, the target function is well-defined, such as maximizing returns or minimizing risk. Supervised learning can directly optimize for these specific objectives, making it easier to evaluate and compare different models.

3 **Discovering hidden Patterns**
= Unsupervised learning can identify hidden patterns and structures within the data without the need for labeled target variables. This is beneficial for stock price optimization since it can reveal market segments, clusters, or other non-obvious relationships in the data.

4. **Early Stopping**
= For iterative algorithms like neural networks, model validation can be used for early stopping. If the performance on the validation set starts deteriorating, training can be halted before overfitting occurs.

5. **Backtesting and Validation**
= Before deploying a statistical-based strategy in real-time, it can be thoroughly tested using historical data to assess its performance and validate its effectiveness. This helps investors gauge how the strategy might have performed in the past and set realistic expectations.

Disadvantages of TRADITIONAL ALGORITHMS

1.**Quality Data**
= Ensuring high-quality data is crucial for accurate stock price optimization. Inadequate data quality, such as missing values, outliers, or errors, can lead to biased models and unreliable predictions.

2.**Market Noise and Uncertainty**
= Financial markets are influenced by various unpredictable factors, introducing noise and uncertainty into the data. This can make it challenging for models to discern genuine signals from random fluctuations.

3.Non-Stationary Nature of Financial Data
= Financial data often exhibits non-stationary behavior, meaning that statistical properties change over time. This poses a problem for traditional machine learning algorithms, which assume stationarity, and may result in models that become less effective as market conditions evolve.

Advantages of Using the Rl-Based Algorithms.

1.**The exploration-exploitation technique**
=Balances trying out different new things and taking advantage of what's figured out. This is different from other learning algorithms. Also, there is no requirement for a skilled human to provide training examples or labeled samples. Furthermore, during the exploration process, the agent is encouraged to explore the uncharted human experts.

2.**Continuous Learning and Improvement**
= RL agents can continuously learn and improve their strategies as new data becomes available. This adaptability to changing market conditions and continuous learning capability can lead to more robust and adaptive stock optimization models.

3.**Learning from Trial and Error**
=RL algorithms learn from trial and error, continuously improving their strategies over time. Through repeated interactions with the market, RL agents can discover optimal trading policies that maximize returns while minimizing risks.

4. This learning model is very similar to the learning of human beings. Hence, it is close to achieving perfection.

5. It can be useful when the only way to collect information about the environment is to interact with it.

Disadvantages of Using the Rl-Based Algorithms.

1.Model Instability
= RL models can be sensitive to hyperparameter choices and initial conditions, leading to potential model instability. This instability might cause fluctuations in performance and hinder consistent and reliable returns.
2.Limited Understanding of Market Behavior

= Financial markets are influenced by human behavior, sentiment, and unforeseen events that might not be adequately captured by RL algorithms. Understanding market dynamics beyond historical data is crucial for making informed investment decisions.

3.**Overfitting and Backtesting Bias**
= RL algorithms can be prone to overfitting, especially when historical data is limited. Overfitting can lead to strategies that perform well on historical data but fail to generalize to future market conditions. Backtesting bias, where a model is unintentionally optimized to perform well on historical data, can also result in inflated return estimates that do not hold in live trading.

ROADMAP→ 2

1.**State Representation**

= In RL, choosing an appropriate state representation is crucial for capturing market conditions effectively. The state representation should encode relevant information about the market that can aid the RL agent in making informed trading decisions.We are going to use the Technical indicators to provide additional information about market conditions. Examples of technical indicators include moving averages, relative strength index (RSI), stochastic oscillators, and Bollinger Bands. These indicators can be calculated based on historical price data and included in the state representation.(These are state representation in trading environments using Gym and Stable Baselines)

2. **Designing The Action Space**
= We will design the action space which our agent can take during the trading process. The action space will include the options to buy, sell and hold.
We are going to use the Discrete action approach

Discrete Actions: One common approach is to define a discrete action space where each action corresponds to a specific trading decision. For example:

- Action 0: Hold (no trading action)
- Action 1: Buy a certain quantity or percentage of the available cash balance
- Action 2: Sell a certain quantity or percentage of a specific stock holding

3.**Reward function**

= We are going to use the reward function that encourages the RL agent to optimize stock returns.So we are going to use the Risk-Adjusted Return
To account for risk, we can incorporate a risk-adjusted component in the reward function. For example, we can use a measure such as the Sharpe ratio, which considers the excess return earned relative to the risk taken. The Sharpe ratio can be calculated as the ratio of the excess return (i.e., return above a risk-free rate) to the volatility of returns.

4.***Training the RL Agent***

= To train our RL agent, we will use a suitable algorithm such as Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C) or Soft Actor Critic (SAC). We will preprocess the historical stock data and set up our environment. For making the environment for our RL agent we can use the gym library. For implementing the algorithms we can utilize popular libraries such as stable-baselines3 or Tensorforce. These libraries provide the implementation of above mentioned libraries.

5.***Evaluating the RL Model***

=As with other models, we will use a different dataset for testing our model. We will assess various performance metrics, including cumulative returns, annualized returns, risk-adjusted measures (Sharpe ratio, Sortino ratio), maximum drawdown, and other relevant evaluation measures. Analyze the agent's performance to understand its strengths, weaknesses, and areas for improvement.

6.**Fine-Tuning**
= We can now iteratively fine tune the RL model and its parameters to enhance the agent's performance. We can experiment with our reward function or try using different RL algorithms. We can utilize the common tuning techniques like hyperparameter tuning, model architecture modifications or ensemble approaches to optimize the performance of our RL agent. We can take assistance from libraries like Optuna or scikit-optimize to help us optimize the hyperparameters.