Details Doc - Phase 2

To enhance the accuracy of predictions and improve the visualization, we can implement additional algorithms and compare their performance. We'll extend the existing code by introducing other advanced deep learning models, such as:

* **Long Short-Term Memory (LSTM) Networks**
* **Gated Recurrent Units (GRU) Networks**
* **Convolutional Neural Networks (CNN) for Time Series**
* **Hybrid Models (CNN-LSTM)**

By training multiple models, we can compare their predictions against actual stock prices and select the one with the best accuracy and profitability potential.

**Explanation and Comparison of Models**

**1. Transformer Model**

* **Architecture**: Utilizes self-attention mechanisms to weigh the importance of different time steps in the input sequence.
* **Strengths**: Captures long-term dependencies and complex temporal patterns.
* **Performance**: Effective for sequences with complex patterns.

**2. LSTM Model**

* **Architecture**: A type of recurrent neural network (RNN) capable of learning long-term dependencies using gates to regulate information flow.
* **Strengths**: Good at capturing temporal dynamics in sequential data.
* **Performance**: Widely used for time series forecasting.

**3. GRU Model**

* **Architecture**: Similar to LSTM but with a simpler structure and fewer gates.
* **Strengths**: Faster to train than LSTM with comparable performance.
* **Performance**: Suitable for datasets where training time is a constraint.

**4. CNN-LSTM Hybrid Model**

* **Architecture**: Combines convolutional layers to extract local features and LSTM layers to capture temporal dependencies.
* **Strengths**: Effective at capturing spatial and temporal patterns.
* **Performance**: Useful when patterns span multiple time steps.

**Model Evaluation**

After training, each model's performance is evaluated using the Root Mean Squared Error (RMSE) on the test set. The model with the lowest RMSE is considered the most accurate.The actual vs. predicted prices are plotted for each model, enabling visual comparison.

**Trading Strategy and Backtesting**

The best-performing model is used to generate trading signals:

* **Buy**: If the predicted price is higher than the previous day's actual price.
* **Sell**: If the predicted price is lower than the previous day's actual price.
* **Hold**: If there is no significant change.

A simplified backtesting is performed to estimate the strategy's profitability:

* **Initial Capital**: $10,000
* **Trading Logic**:
  + **Buy as many shares as possible** when a buy signal is generated.
  + **Sell all shares** when a sell signal is generated.
* **Final Portfolio Value**: Calculated after applying the trading strategy over the test period.

**Results and Interpretation**

* **Visualization**: The plot shows how each model's predictions align with the actual stock prices. Models that closely follow the actual price movements indicate better performance.
* **Accuracy Comparison**: The RMSE values for each model are printed, allowing a quantitative comparison.
* **Best Model Selection**: The model with the lowest RMSE is selected as the best model.
* **Profitability Analysis**:
  + **Final Portfolio Value** indicates how much money the strategy would have made.
  + **Net Profit** shows the gain or loss compared to the initial investment.

**Conclusion**

By implementing multiple algorithms and comparing their performance, we enhance the likelihood of selecting a model that not only provides more accurate predictions but also improves potential profitability. The inclusion of different models allows for a comprehensive analysis, catering to the model that best captures the underlying patterns in the stock price data.

**Next Steps for Improvement**

* **Hyperparameter Tuning**: Adjust model parameters (e.g., number of layers, neurons, dropout rates) to optimize performance.
* **Ensemble Methods**: Combine predictions from multiple models to potentially improve accuracy.
* **Feature Expansion**:
  + Include additional technical indicators (e.g., moving averages, RSI, MACD).
  + Incorporate macroeconomic factors.
* **Risk Management**:
  + Implement stop-loss and take-profit mechanisms.
  + Consider position sizing strategies.
* **Advanced Backtesting**:
  + Include transaction costs, slippage, and market impact.
  + Use more sophisticated backtesting frameworks.