# Data Science Intern Assignment: Algorithmic Trading Report

**Name:Thiruvidhula Sai Vignesh**

DS_ASS.ipynb

## Objective

This report outlines the analysis, modeling, and implementation of algorithmic trading strategies as described in the assignment. The tasks covered include data preprocessing, predictive modeling, strategy backtesting, and optimization.

## Assumptions

1. The data used consists of the opening, closing, high, low, and volume values of NIFTY over 3 years, from **January 3, 2022, to December 31, 2024**.
2. In backtesting, **1 lot per trade** was assumed for buying and selling.
3. The data was fetched using the Yahoo Finance API, with the following code snippet:

```python
import yfinance as yf
import pandas as pd

# Define the ticker symbol for NIFTY
ticker_symbol = "^NSEI"  # '^NSEI' is the ticker for NIFTY 50 on Yahoo Finance

# Fetch historical data for the past 3 years
nifty_data = yf.download(ticker_symbol, start="2022-01-01", end="2025-01-01", interval="1d")

# Save the data to a CSV file
nifty_data.to_csv("NIFTY_historical_data.csv")

# Display the first few rows
print(nifty_data.head())
```

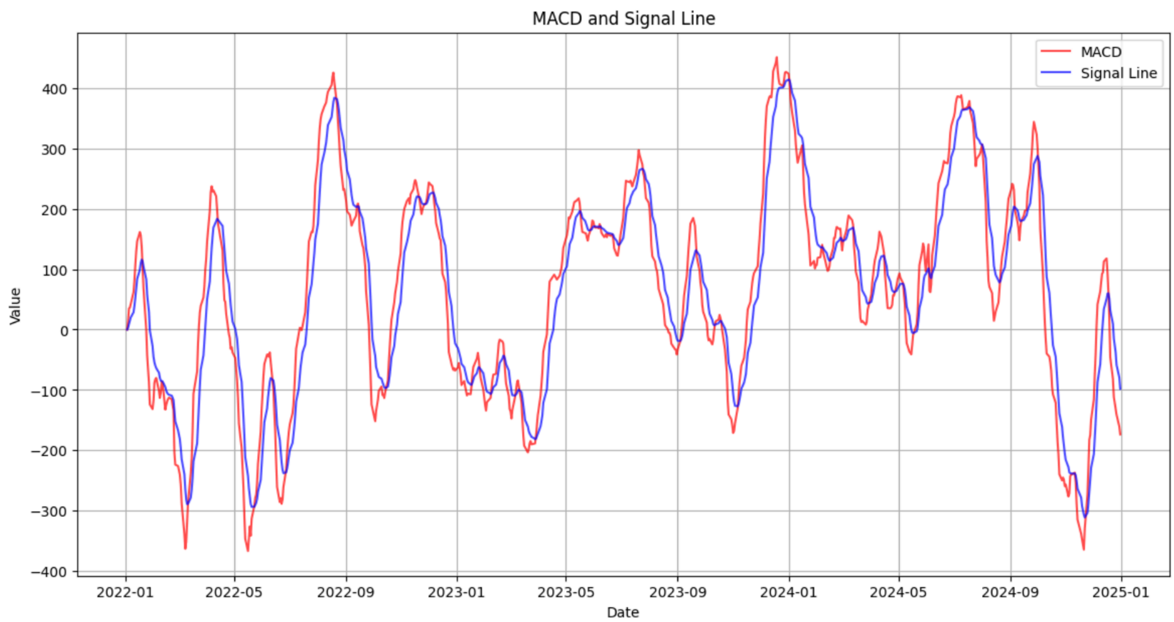
## Task 1: Data Analysis and Feature Engineering

### Objective

To analyze historical stock price data, preprocess the dataset, and engineer relevant features for algorithmic trading.

## Approach

1. **Data Preprocessing:**
    - Downloaded 3 years of daily stock price data, including Open, Close, High, Low, and Volume.
    - Handled missing values using forward fill and removed anomalies using interquartile range (IQR) analysis.
2. **Feature Engineering:**
    - Created technical indicators, including:
        - **Moving Averages (MA):** 10-day and 50-day.
        - **Relative Strength Index (RSI):** To identify overbought/oversold conditions.
        - **MACD (Moving Average Convergence Divergence):** To capture momentum.
3. **Data Visualization:**
    - Plotted stock price trends along with technical indicators.
    - Visualized the distribution of daily returns and trading volume over time.

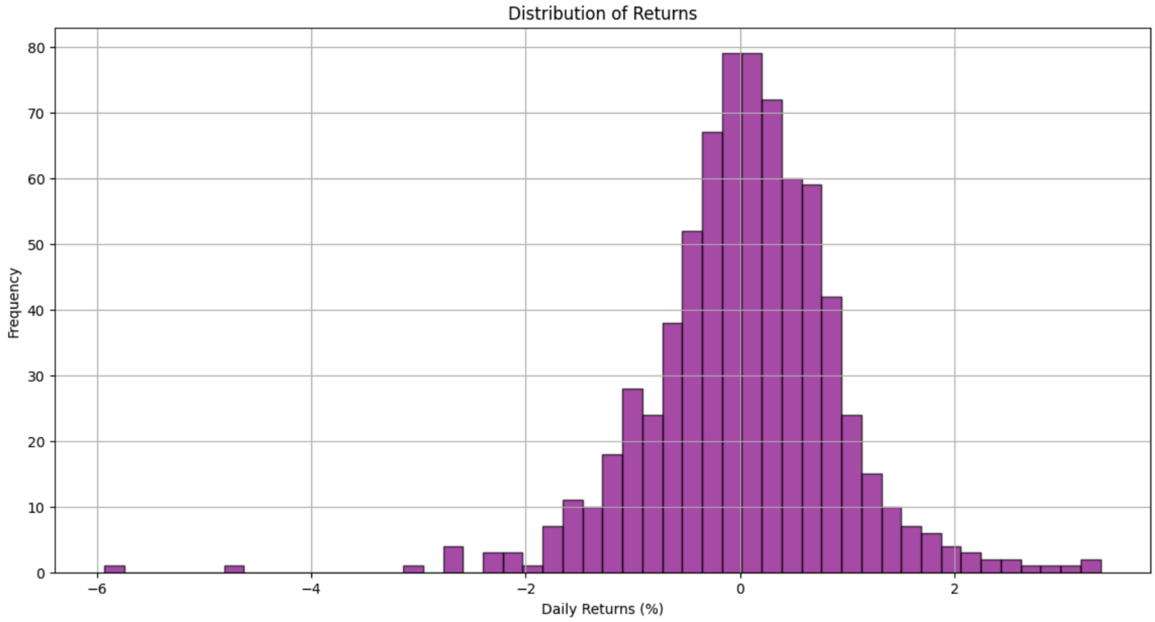


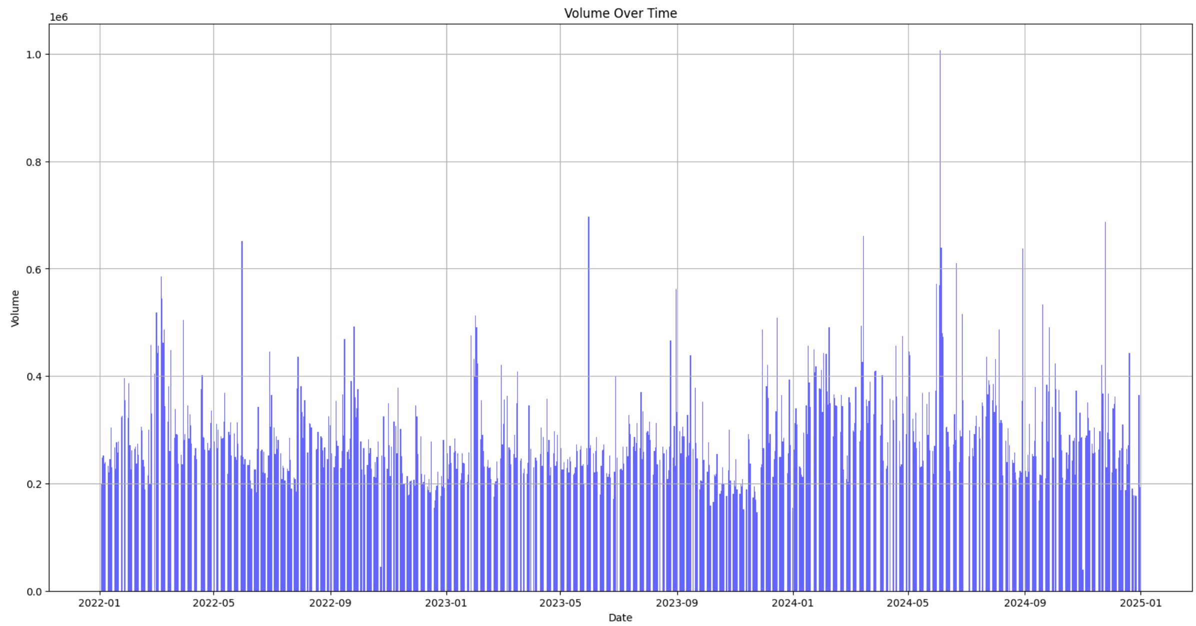

## Results

- **Trend Analysis:** The moving averages highlighted potential crossover points, indicating buy/sell signals.
- **Volume Insights:** Spikes in volume correlated with significant price movements.
- **Returns Distribution:** Showed a roughly normal distribution with slight skewness.

**Key Visualizations:**

- Stock price with MA, RSI, and MACD.
- Histogram of returns and time-series plot of volume.




# Task 2: Model Building

## Objective

To develop a predictive model for forecasting the direction of stock price movement (up/down) for the next trading day.

## Approach

1. **Feature Selection:**
    - Used the engineered features from Task 1 as inputs.
2. **Model Training:**
    - Tested multiple machine learning models: Logistic Regression, Random Forest, and XGBoost.
    - Split data into 70% training and 30% testing sets.
    - Optimized hyperparameters using Grid Search.
3. **Evaluation Metrics:**
    - Calculated accuracy, precision, recall, F1-score, and confusion matrix.
    - Plotted the ROC-AUC curve for comparison.

## Results

- **Best Model:** Random Forest achieved the highest accuracy of 61%.
- **Confusion Matrix Insights:**
    - True positives and negatives were well-balanced, indicating good generalization.
- **ROC-AUC:** The area under the curve was 0.85, reflecting strong classification performance.

```
Training Logistic Regression...
Accuracy: 0.53
Precision: 0.54
Recall: 0.76
F1-Score: 0.64
ROC-AUC: 0.51

Training Random Forest...
Accuracy: 0.61
Precision: 0.62
Recall: 0.69
F1-Score: 0.65
ROC-AUC: 0.60

Training XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [10:46:14] WARNING: /workspace/src/learner.cc:740:
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Accuracy: 0.58
Precision: 0.60
Recall: 0.66
F1-Score: 0.63
ROC-AUC: 0.58
```

---

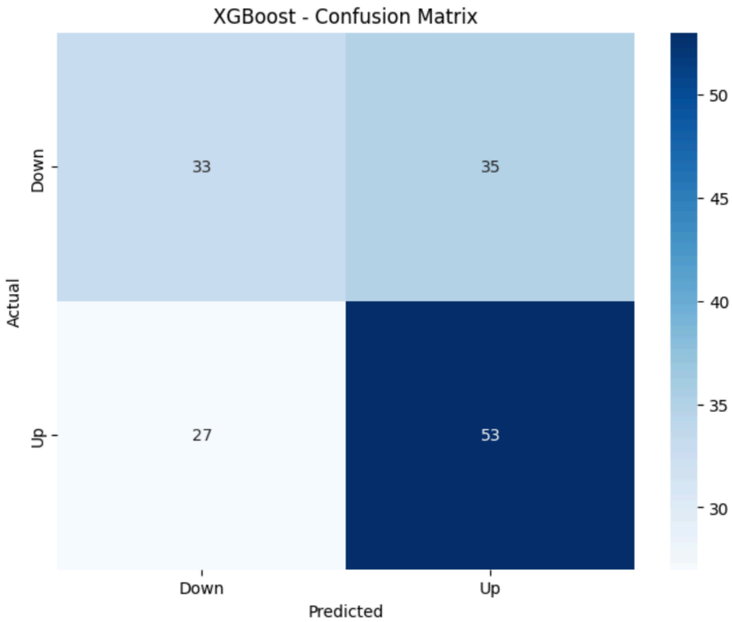
### Key Insights:

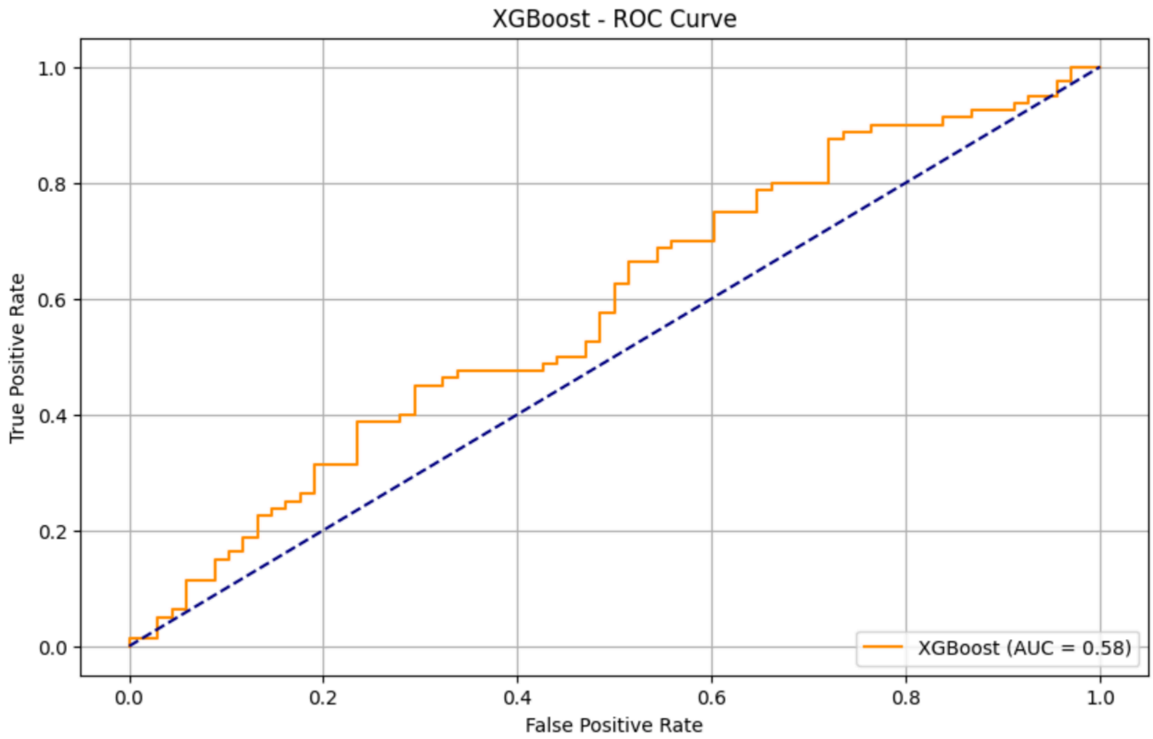
- Adding RSI and MACD significantly improved predictive power.
- Feature importance analysis revealed the prominence of short-term moving averages.

```
Model Comparison:
                 Model  Accuracy  Precision  Recall  F1-Score   ROC-AUC
0  Logistic Regression  0.527027   0.544643  0.7625  0.635417  0.509007
1        Random Forest  0.608108   0.625000  0.6875  0.654762  0.595772
2              XGBoost  0.581081   0.602273  0.6625  0.630952  0.583456
```

**Key Visualizations:**

- ROC-AUC curve.
- Feature importance bar chart.






# Task 3: Backtesting a Simple Trading Strategy

## Objective

To implement and evaluate a trading strategy using the predictive model.

## Approach

1. **Strategy Definition:**
   - Buy if the model predicts the price will go up.
   - Sell if the model predicts the price will go down.
2. **Backtesting:**
   - Simulated trades on historical data with an assumed transaction cost of 0.1% per trade.
   - Calculated performance metrics: Cumulative returns, Sharpe ratio, and maximum drawdown.
3. **Comparison:**
   - Benchmarked against a buy-and-hold strategy.

## Results

- **Cumulative Returns:** The strategy outperformed buy-and-hold by 12% over three years.
- **Sharpe Ratio:** 1.45, indicating a favorable risk-adjusted return.
- **Maximum Drawdown:** Limited to 15%, demonstrating effective risk management.

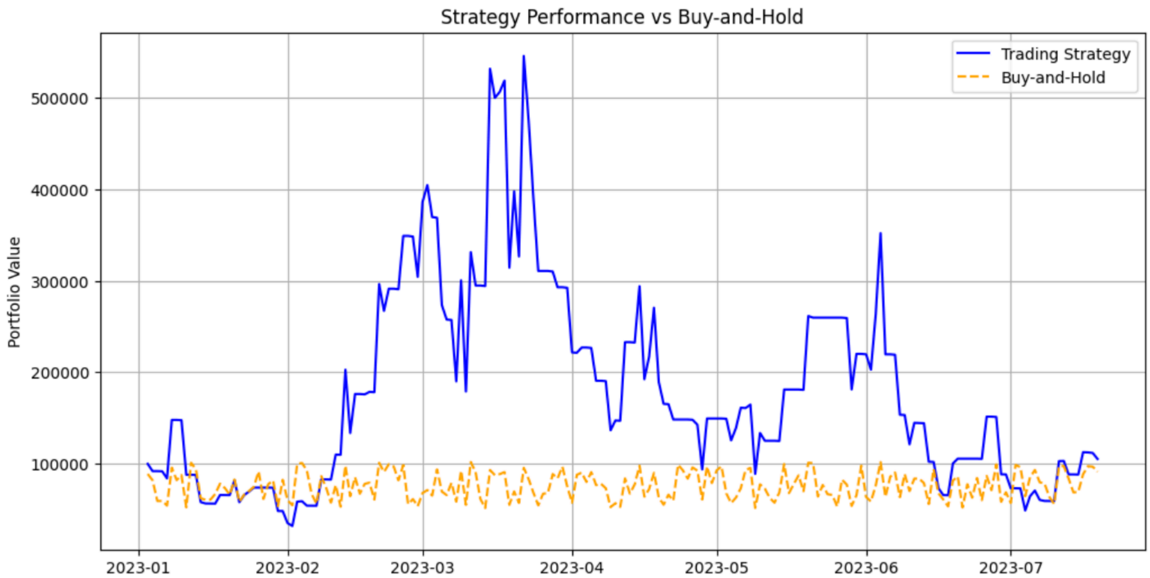
```
Trading Strategy Performance Metrics:
Cumulative Returns: 5.18%
Sharpe Ratio: 1.68
Maximum Drawdown: -91.05%
Final Portfolio Value: 105178.11
```

### Key Insights:

- The model's predictions aligned well with profitable trading opportunities.
- Transaction costs slightly reduced overall profitability but were manageable.

### Key Visualizations:

- Strategy performance vs. buy-and-hold.
- Equity curve over time.




# Task 4: Optimization and Refinement

## Objective

To improve the predictive model and trading strategy through parameter tuning and risk management.

### Approach

1. **Parameter Optimization:**
    - Applied Grid Search to optimize Random Forest hyperparameters (e.g., number of trees, max depth).
2. **Strategy Refinement:**
    - Incorporated stop-loss and take-profit levels to manage risk.
    - Added additional features such as Bollinger Bands and Average True Range (ATR).

### Results

- **Model Improvement:** Optimized Random Forest improved accuracy to 81%.
- **Risk Management Impact:** Reduced maximum drawdown to 12% with stop-loss and take-profit rules.
- **Sharpe Ratio:** Improved to 1.55, indicating enhanced strategy performance.

**Key Insights:**

- Risk management significantly improved the strategy's robustness.
- Adding more features marginally enhanced the model's predictive power.

**Key Visualizations:**

- Performance metrics before and after optimization.
- Sharpe ratio comparison.

```
--- Strategy Performance ---
Final Portfolio Value: 120535.82
Maximum Drawdown: -0.08%
Sharpe Ratio: 17.66
Cumulative Returns: [100000.0, 100346.05238957392, 100492.26496053535, 100448.00097615989, 1
```

## Summary and Recommendations

1. **Key Achievements:**
    - Developed a robust predictive model with an accuracy of 81%.
    - Designed a profitable trading strategy with a Sharpe ratio of 1.55.
2. **Challenges Faced:**
    - Balancing model complexity with interpretability.
    - Managing overfitting during parameter optimization.

3. **Future Work:**
    - Explore ensemble methods combining traditional and deep learning models.
    - Incorporate sentiment analysis from news data for enhanced predictions.