ML-Based Trading Strategy

1. Introduction

Algorithmic trading, or automated trading, is gaining popularity in financial markets. Machine learning offers potential to enhance trading strategies by analyzing data and identifying patterns. In this report, we develop and evaluate a machine learning-based trading strategy.

Our objectives are to develop a trading strategy based on financial data indicators and predict buy, sell, and hold signals using multinomial logistic regression. We evaluate our strategy using metrics such as AUC-ROC, F1 score, and accuracy.

The report is structured as follows. Section 2 provides background on algorithmic trading and machine learning in finance. Section 3 covers data extraction , indicator and new features creation. Section 4 is about trading strategy . Section 5 includes Regression model, evaluation metrics and results.

2. Background

Algorithmic trading, or algo-trading, uses computer algorithms to execute trading orders. It's becoming popular due to technology advances and offers advantages like speed and reduced costs. Machine learning techniques analyze data to extract insights, aiding in decision-making. Despite challenges like data quality and regulatory compliance, algorithmic trading and machine learning offer opportunities to improve strategies and gain a competitive edge.

Previous research in finance covers topics such as asset price prediction and sentiment analysis, contributing valuable insights to our approach.

3. Data extraction , Indicators and New features

I used yahoo finance for extracting historical data of Reliance for the year 2022 . i created several indicators from scratch for using them for my trading strategy and for ML model those indicators are calculating Simple Moving Average(EMA) , Exponential Moving Average(EMA) Moving Average convergence divergence(MACD), Standard deviation,stochastic oscilater and On balance volume.
After that using these indicators i calculated SMA of 9 day and 21 day,EMA of 9day and 21 day and stored them in my dataframe ‘useful_data_coll’ as ‘s_sma’ ,’l_sma’

In this section, we detail the process of data extraction from Yahoo Finance for historical data of Reliance for the year 2022. We then introduce the indicators developed for the trading strategy .

3.1 Indicators Used

We created several indicators from scratch to derive insights from the historical price and volume data. These indicators include:

- **Simple Moving Average (SMA)**: A moving average calculated by adding the closing prices of a security for a specific period and then dividing the sum by the number of periods. We calculated two SMAs: a short-term SMA over a 9-day period (`s_sma`) and a long-term SMA over a 21-day period (`l_sma`).
- **Exponential Moving Average (EMA)**: Similar to SMA, but it gives more weight to recent prices. We computed two EMAs: a short-term EMA over a 9-day period (`s_ema`) and a long-term EMA over a 21-day period (`l_ema`).
- **Moving Average Convergence Divergence (MACD)**: A trend-following momentum indicator that shows the relationship between two moving averages of a security's price. We calculated the MACD line, signal line, and MACD histogram.
  - **MACD Line**: The MACD line is the difference between the short-term EMA (12-day period) and the long-term EMA (26-day period). It reflects the strength and direction of the trend.
  - **Signal Line**: The signal line is an EMA (9-day period) of the MACD line. It smoothens out the MACD line and provides trading signals.
  - **MACD Histogram**: The MACD histogram represents the difference between the MACD line and the signal line. It visualizes the convergence and divergence between the two lines, indicating potential trend reversals.
- **Standard Deviation (Std Dev)**: A measure of the dispersion or volatility of a set of values. We computed the standard deviation over a window of 37 days (`std_dev`).
- **Stochastic Oscillator**: A momentum indicator comparing a particular closing price of a security to a range of its prices over a certain period of time. We used a period of 14 days (`stc_osc`).
- **On-Balance Volume (OBV)**: A technical analysis indicator that relates volume to price change. We calculated the OBV values (`obv`) based on changes in closing prices and trading volumes.

3.2 New Features Created from Indicators

In addition to the standard indicators, we created new features derived from existing indicators to capture additional insights into market trends and momentum:

- **SMA Difference (sma_diff):**
  - Description: SMA difference represents the difference between the short-term and long-term simple moving averages (SMAs). It indicates the relative position of short-term and long-term trends.
  - Calculation: **`sma_diff = short-term SMA - long-term SMA`**
  - Interpretation: Positive values suggest a short-term uptrend, while negative values indicate a short-term downtrend.
- **EMA Difference (ema_diff):**
  - Description: EMA difference represents the difference between the short-term and long-term exponential moving averages (EMAs). Similar to SMA difference, it helps identify short-term trends relative to long-term trends.

    - Calculation: `ema_diff = short-term EMA - long-term EMA`
    - Interpretation: Positive values indicate a short-term uptrend, while negative values suggest a short-term downtrend.
- **Rate of Change (ROC):**
    - Description: Rate of Change measures the percentage change in price over a specified period. It quantifies the momentum of price movements.
    - Calculation: `ROC = ((current price - price 'n' periods ago) / price 'n' periods ago) * 100`
    - Interpretation: Higher ROC values signify stronger momentum, while lower values may indicate weakening momentum or potential trend reversals**.**

3.3 Working Principles and Period Selection

Each indicator serves a unique purpose in analyzing price movements and market trends. The periods chosen for each indicator were determined based on commonly used timeframes in technical analysis and empirical observations of market behavior.

- For short-term indicators (e.g., 9-day SMA and EMA), shorter periods were chosen to capture more recent price movements and react quickly to market changes.
- Conversely, longer periods were selected for long-term indicators (e.g., 21-day SMA and EMA) to smooth out fluctuations and identify broader trends in the market.

3.4 Calculation and Data Storage

We implemented custom functions to calculate each indicator from the historical price and volume data. These functions utilize mathematical formulas and algorithms specific to each indicator's calculation method. The resulting indicator values were then appended to a dataframe named `useful_data_coll`, facilitating further analysis and model development and in this step we also remove the NaN rows from over dataframe so that it not cause harm in future calculations .

4. Building strategy

In this section, we elaborate on the process of generating trading decisions based on a predefined strategy .We have devised a trading strategy that leverages various technical indicators, including OBV, EMA, SMA, and MACD histogram, to make buy, sell, or hold decisions. The strategy is implemented through the `generate_trading_decisions(df)` function, which evaluates specific conditions for buying and selling based on changes in indicator values.

- **Buy Condition**: A buy signal is generated when OBV is increasing, and EMA, SMA, and MACD histogram differences are positive. Additionally, a buy signal is triggered when the MACD histogram crosses from negative to positive.
- **Sell Condition**: Conversely, a sell signal is identified when OBV is decreasing, and EMA, SMA, and MACD histogram differences are negative. Moreover, a sell signal is initiated when the MACD histogram crosses from positive to negative.
- **Hold Condition**: When there is no buying and selling trigger , there is hold .

The function assigns a numerical value to each trading decision (-1 for sell, 0 for hold, and 1 for buy), which is stored in the `td_ds` column of the DataFrame `df`.

5. Regression Model Training and Evaluation

5.1 Splitting the data

Before training the regression model, it's essential to divide our dataset into separate training and testing sets. This step ensures that our model is trained on one set of data and evaluated on another, thereby providing an unbiased assessment of its performance.

Process Overview:-

We start by shuffling the dataset to randomize the order of samples, minimizing any potential biases in the data.We allocate 80% of the shuffled dataset for training and reserve the remaining 20% for testing. The shuffled dataset is split into training and testing sets based on the determined ratio. The independent features (X_train and X_test)consist of all columns except the 'td_ds' column, representing trading decisions. The dependent variable (y_train and y_test) is the 'td_ds' column.

5.2 Regression model training

We proceed with training a logistic regression model using the training data. Following the model training, we evaluate its performance using various metrics such as AUC-ROC, accuracy, F1 score, and Confusion Matrix.

5.2.1 Model Training: We utilize a logistic regression model with the 'newton-cg' solver and 'multinomial' multi-class strategy. The model is fitted to the training data, aiming to learn patterns and relationships between input features and trading decisions using the sklearn.linear_model.

5.2.2 Model Coefficients and Intercept: After training the model, we inspect the coefficients (m) and intercept (b) learned by the logistic regression model.

5.2.3 Model Prediction: Using the trained model, we predict the trading decisions for the testing dataset.

5.3 Model Evaluation

We evaluate the performance of the trained logistic regression model using various metrics:

- **AUC-ROC Score:** Measures the model's ability to distinguish between different classes.
- **Accuracy**: Calculates the percentage of correctly predicted instances.
- **F1 Score**: Harmonic mean of precision and recall, providing a balance between the two.
- **Confusion Matrix**: Summarizes the actual and predicted classifications.

These evaluation metrics provide insights into the model's performance and its ability to make accurate predictions regarding trading decisions based on historical data.