Predicting H1 Target for November 2024: A Comprehensive Predictive Modeling Project

Project Overview

This project focuses on predicting the **H1 (1-hour) Close price movements** for November 2024, based on historical **M1 (1-minute) OHLCV** data. The data was resampled and engineered into a more informative format, and various machine learning models were explored. The final solution leverages **XGBoost** with optimal feature selection and hyperparameter tuning, achieving a high level of accuracy.

Target Categories:

The project involves predicting two targets:

1. **Target 1: Percentage Change Categories**
   - Predict the percentage change in the Close price over the next 10 H1 candles:
     - **A:** (< -5%)
     - **B:** (-5% to -2%)
     - **C:** (-2% to +2%)
     - **D:** (+2% to +5%)
     - **E:** (> +5%)
2. **Target 2: Momentum Target**
   - Derived from the **Relative Strength Index (RSI)** to assess market momentum:
     - **A:** RSI < 30 (Oversold)
     - **B:** 30 ≤ RSI < 50 (Weak momentum)
     - **C:** 50 ≤ RSI < 70 (Moderate momentum)
     - **D:** RSI ≥ 70 (Overbought)

Understanding Momentum (RSI)

Momentum measures the strength of price movements over time. In this project:

- **RSI (Relative Strength Index):**
  - RSI is calculated using the ratio of average gains to average losses over a specified period (typically 14 intervals).
  - RSI values range from **0 to 100**, indicating market conditions:
    - **< 30:** Oversold (possible upward reversal).
    - **30-50:** Weak momentum (sideways movement).
    - **50-70:** Moderate momentum (upward trend).
    - **> 70:** Overbought (possible downward reversal).

Dataset Details

The dataset consists of historical **M1 OHLCV** data in CSV format, containing the following fields:

- **TimeStamp**: Date and time.
- **OHLCV**: Open, High, Low, Close, and Volume values.

The data was resampled to **H1 (1-hour)** intervals for analysis.

Key Features

1. **Lagged Values**: Lagged features up to 10 hours for each variable.
2. **Moving Averages**: Rolling averages to capture trends.
3. **RSI and Momentum Target**: Momentum metrics based on RSI.

Approach and Methodology

Step 1: Data Preprocessing

- Resampled the data from M1 to H1 format.
- Engineered lag features, moving averages, and RSI momentum metrics.
- Defined two target variables:
  - **Target 1:** Percentage change in Close price.
  - **Target 2:** RSI-based momentum target.

Step 2: Model Exploration

- Multiple machine learning models were evaluated for predicting both targets:
  - **Target 1:** XGBoost outperformed other models such as KNN, Random Forest, and RNN.
  - **Target 2:** Random Forest was used and yielded strong results for classifying RSI-based momentum categories.

Step 3: Hyperparameter Tuning

- Optimized the XGBoost and Random Forest models using grid search for key hyperparameters, including:
  - Learning rate
  - Maximum depth
  - Number of estimators

Step 4: Feature Selection

- **Variance Inflation Factor (VIF):**
  - Iteratively eliminated features with high multicollinearity to improve model performance and interpretability.

Step 5: Evaluation

- **Confusion Matrix:**
  - Evaluated classification performance for both targets.
  - Heatmaps were generated for categories A to E.
- **Time-Series Plots:**
  - Visualized train-test splits and predictions.
- **Momentum Insights:**
  - Used RSI categories to understand market behavior, which improved the model's ability to classify momentum shifts effectively.

Results

1. **Best Model:**
   - **Target 1:** XGBoost demonstrated the best performance with:
     - **RMSE:** 0.0673
     - **R^2 Score:** 0.9832
   - **Target 2:** Random Forest effectively classified RSI-based momentum categories.
2. **Visualization:**
   - Confusion matrix heatmaps for A-E categories.
   - Dynamic and static plots of predicted vs. true values.

Tools and Libraries

The following tools and libraries were utilized throughout the project:

- **Programming Language:** Python
- **Libraries:**
  - **Pandas & NumPy:** Data preprocessing and numerical operations.
  - **Scikit-learn:** Machine learning models and feature selection.
  - **Statsmodels:** Variance Inflation Factor calculation.
  - **XGBoost:** Gradient boosting model for Target 1.
  - **Matplotlib & Seaborn:** Static visualizations.
  - **Plotly:** Dynamic visualizations.

Repository Contents

- **Scripts:**
  - `data_resampling.py`: Preprocesses the data, adds lags, and generates Target 1.
  - `momentum_target.py`: Processes data and generates RSI-based Target 2.
  - `feature_selection.py`: Implements VIF-based feature selection.
  - `best_model.py`: Trains and evaluates models.
- **Notebooks:** Step-by-step workflows for different modeling approaches.
- **Visualizations:** Saved plots and results.

Steps to Reproduce

1. Clone the repository:

```
git clone https://github.com/mdai61/eDO.git
```

2. Install dependencies:

```
pip install -r requirements.txt
```

3. Run the preprocessing script for Target 1:

```
python data_resampling.py
```

4. Run the preprocessing script for Target 2:

```
python momentum_target.py
```

5. Train and evaluate the models:

```
python best_model.py
```