Implementation

Paper: Machine Learning Algorithms for Forecasting and Categorizing Euro-to-Dollar Exchange Rates

Abstract:

This paper investigates the application of various machine learning algorithms to predict optimal trading times for the Euro-to-Dollar (EUR/USD) exchange rate, a key currency pair in the global foreign exchange market. Forecasting foreign exchange movements is a challenging task due to the volatility and complexity of market factors such as interest rates, geopolitical events, and economic indicators. The research focuses on leveraging machine learning models, including Logistic Regression, Random Forest Classifier, and Naive Bayes, along with an ensemble Voting Classifier to combine their strengths. The data set spans three years of daily exchange rate data, from December 2003 to July 2022, and incorporates technical indicators such as the Relative Strength Index (RSI) and Exponential Moving Average (EMA).
By preprocessing the data and integrating these technical features, the models aim to predict whether the EUR/USD exchange rate will rise or fall on the following trading day, assisting traders in making more informed buy and sell decisions. The experimental results show that the proposed ensemble model achieved an accuracy of 85.46%, outperforming individual models like Adaboost and Gradient Boosting. This high accuracy demonstrates the effectiveness of combining machine learning algorithms for financial forecasting. The study highlights the importance of utilizing machine learning in financial markets, providing a robust and data-driven approach to improve investment strategies and reduce risk in currency trading. Furthermore, the methodology can be generalized to other currency pairs and financial assets, making it a versatile tool for market prediction.

Problem Formulation:

Currency markets are inherently complex, influenced by a multitude of factors ranging from macroeconomic indicators (like inflation rates and GDP growth) to political events and market sentiment. Traditional statistical methods often struggle to capture the dynamic and non-linear nature of these influences, leading to suboptimal predictions. In contrast, machine learning offers the ability to model complex relationships and make more accurate predictions.

Methodology:

Dataset:

The dataset used in this research spans daily EUR/USD prices from December 1, 2003, to July 29, 2022, covering three years without missing data. The data points include four key financial features: Open, High, Low, and Close prices for each day and also it has Adjacent close and volume. Additionally, the dataset includes computed technical indicators such as RSI, EMA, and TargetNextClose (the future closing price), as well as TargetClass, which denotes whether the price is rising or falling.

Feature Engineering:

Feature engineering is a pivotal step in the machine learning pipeline as it directly impacts the model's ability to uncover meaningful patterns in the data. In this study, a range of technical indicators was computed from historical price data to enhance the model's understanding of market dynamics.

- **Relative Strength Index (RSI)**: As a momentum oscillator, RSI helps to evaluate the speed and changes in price movements. By identifying overbought or oversold conditions, it offers insights into potential price reversals.
- **Exponential Moving Average (EMA)**: Unlike the simple moving average, EMA assigns greater importance to recent data, making it more responsive to new price movements. This helps in identifying trend directions more quickly, allowing the model to detect short-term shifts in the market.

In addition to these indicators, traditional features such as the Open, High, Low, and Close (OHLC) prices were also incorporated, offering a comprehensive view of price action over time. By combining these technical indicators with raw price data, the model benefits from a more detailed understanding of both short-term fluctuations and long-term trends.

Before feeding these features into the machine learning models, **MinMaxScaler** was applied to scale all the features to a range between 0 and 1. This normalization process ensures that no single feature, regardless of its magnitude, dominates the learning process. By bringing all features to a uniform scale, the model can better interpret the relationships among them, ultimately improving its prediction accuracy and stability across different market conditions.

Machine Learning Models:

- Logistic Regression (LR): A binary classification algorithm that models the probability of an event occurring by fitting data to a logistic curve. It is well-suited for predicting whether the EUR/USD price will go up or down.

- Decision Tree Classifier: The Decision Tree Classifier is a supervised learning algorithm used for classification tasks. It splits the data into branches based on feature values, forming a tree structure that predicts the target class by following decision rules from the root to leaf nodes.

- Naive Bayes (NB): A probabilistic classifier that applies Bayes’ Theorem with the assumption that features are independent. Despite its simplicity, Naive Bayes can perform well in many classification tasks.

- AdaBoost and Gradient Boosting Classifiers (GBC): Boosting methods that focus on improving the accuracy of weak classifiers by emphasizing the misclassified examples in each subsequent iteration.

- Voting Classifier (VC): The authors' final model, which combines Logistic Regression, Random Forest, and Naive Bayes using soft voting. The soft voting mechanism averages the predicted probabilities from all three models and selects the class with the highest average probability as the final prediction.

Experimental Design:

The dataset was divided into two parts: 80% for the training set and 20% for the testing set. The training set served as the foundation for the machine learning models, allowing them to learn underlying patterns, correlations, and relationships from the data. This phase is critical, as the models adjust their internal parameters to minimize errors and improve their ability to make accurate predictions.

Once trained, the models were tested on the remaining 20% of the dataset, which was reserved for evaluation purposes. The testing set consisted of unseen data that the models did not encounter during the training phase. By assessing model performance on this separate set, we can accurately measure the model's ability to generalize beyond the training data. This helps prevent overfitting, where a model performs well on training data but poorly on new, unseen data.

The primary performance metric used for evaluation was accuracy, which is a straightforward yet essential measure of the model's effectiveness. Accuracy calculates the proportion of correctly classified instances, indicating how well the model can distinguish between positive and negative classes. A high accuracy score suggests that the model can reliably predict outcomes in a real-world scenario. However, in cases where class imbalance exists, additional metrics such as precision, recall, and F1-score may also be necessary to gain a more comprehensive understanding of the model's true performance across different classes.

Results:

The experimental results indicate that the Voting Classifier, an ensemble approach, significantly outperformed individual models in predicting buy and sell signals for the EUR/USD currency pair, achieving an impressive accuracy of 85.46%. This method combines predictions from multiple models, enhancing robustness and capturing diverse patterns within the data.

Performance of Individual Models:

1. **Logistic Regression**: Achieved an accuracy of 81.51% when used independently. Known for its efficiency in binary classification, Logistic Regression effectively captures linear relationships but may miss complex patterns, explaining its slightly lower performance compared to the ensemble method.

2. **Decision Tree Classifier**: This model attained an accuracy of 83.59%, surpassing Logistic Regression by leveraging its capability to capture non-linear relationships and feature interactions. However, its performance was still below that of the Voting Classifier, possibly due to overfitting tendencies inherent in individual decision trees.
3. **Gradient Boosting Classifier**: With an accuracy of 77.67%, Gradient Boosting performed reasonably well but lagged behind other models. This ensemble technique builds models sequentially, correcting previous errors, yet its performance suggests challenges in capturing specific market patterns. Its sensitivity to noise and potential overfitting could explain this lower accuracy.
4. **Gaussian Naive Bayes**: Achieved only 50.05% accuracy. This model assumes feature independence, which is often not the case in financial markets, leading to its poor performance in predicting the complex buy and sell signals.
5. **AdaBoost Classifier**: Also underperformed, with an accuracy of 58.98%. Although AdaBoost combines multiple weak learners to create a stronger model, it struggled to capture the underlying market dynamics effectively.

The results highlight the superiority of the **Voting Classifier** in predicting buy and sell signals for the EUR/USD currency pair. By effectively combining different models, it achieved the highest accuracy, demonstrating the advantages of ensemble methods. The Decision Tree model, while slightly less accurate, showcased the potential of non-linear models in financial prediction. In contrast, the Gradient Boosting Classifier, despite its reasonable accuracy, indicated room for improvement in capturing complex patterns. Finally, the poorer performances of Gaussian Naive Bayes and AdaBoost suggest that these models may not be well-suited for the intricacies of financial market prediction. Overall, the findings underscore the importance of model selection and combination in achieving high accuracy in financial forecasting.

Code Link:
https://github.com/magical-king03/Machine-Learning-Project/blob/main/model.ipynb

Models	Accuracy
Voting Classifier	85.46
Decision Tree Classifier	83.59
Logistic Regression	81.51
Gaussian Naive Bayes	50.05
AdaBoost Classifier	58.98
Gradient Boosting Classifier	77.67