ML Based Momentum Trading Strategy using Voting Classifier

The stock market is a dynamic and volatile system where predicting price movements can yield significant financial returns. This report presents a machine learning approach to predict stock market trends for multiple companies based on technical indicators and historical price data. The models employed include XGBoost, SVM, Random Forest, Logistic Regression, and a Soft Voting Classifier to combine the predictions of all models. This approach is aimed at predicting stock price movement signals such as "Buy", "Hold", or "Sell".

Data Collection:

The data for this analysis was sourced from Yahoo Finance (yfinance library), where stock prices of five companies—Apple (AAPL), Meta (META), Tesla (TSLA), JPMorgan Chase (JPM), and Amazon (AMZN)—were downloaded. The data spans from January 1, 2015, to January 1, 2025. The following features were collected for each stock:

1. **Open, High, Low, Close** prices.
2. **Volume** traded.

Additional features were derived using technical indicators from the ta library:

- **Moving Averages (SMA, EMA)**
- **RSI (Relative Strength Index)**
- **MACD (Moving Average Convergence Divergence)**
- **Stochastic Oscillator**
- **Bollinger Bands**
- **On-Balance Volume**

These features are known for their ability to indicate trends, momentum, volatility, and volume dynamics in stock prices.

Feature Engineering:

The feature engineering process involved calculating the following indicators:

1. **SMA (50-day and 200-day)**: Simple Moving Averages used to identify long-term trends.
2. **EMA (21-day)**: Exponential Moving Average to give more weight to recent data.
3. **RSI (Relative Strength Index)**: Measures overbought or oversold conditions in the stock.

4. **MACD (Moving Average Convergence Divergence)**: A trend-following momentum indicator.
5. **Stochastic Oscillator**: Identifies overbought or oversold levels based on closing price.
6. **Bollinger Bands**: Indicates volatility by plotting standard deviations away from a moving average.
7. **OBV (On-Balance Volume)**: Measures buying and selling pressure based on volume.

Custom Features(6 & 7):

In addition to the commonly used indicators in the dataset, two custom features were engineered to capture additional market dynamics and price action insights that are often useful in predicting price trends and breakouts. These features are designed to enhance the model's ability to interpret market conditions beyond basic price and volume indicators.

Feature 1: Bollinger Band Width (BBW)

Bollinger Band Width (BBW) is a volatility-based feature that measures the distance between the upper and lower Bollinger Bands, which are calculated based on a simple moving average (SMA) and a multiple of the standard deviations of price movements. The width of the Bollinger Bands serves as an indicator of market volatility:

- **Wider Bollinger Bands** indicate higher volatility, which often precedes strong price movements or momentum breakouts. A wider band suggests that the market is experiencing larger price fluctuations, and this may indicate the potential for significant price movements in the near future.
- **Narrower Bollinger Bands** suggest lower volatility, which usually corresponds to a consolidation phase. When the price moves within a narrow range, the market is likely to be in a period of indecision or quietness, often leading to a breakout once the price action breaks out of the range.

Feature 2: On-Balance Volume (OBV)

On-Balance Volume (OBV) is a cumulative volume-based indicator that helps to measure the flow of money into or out of an asset. The indicator adds the volume to the OBV when the price closes higher than the previous close and subtracts the volume when the price closes lower. This accumulation of volume is significant because it suggests buying or selling pressure:

- If **OBV is increasing** while the price remains relatively flat or moves in a narrow range, this suggests that there is underlying **buying pressure**, and the market may be preparing for a bullish breakout. The rise in OBV without a corresponding increase in price indicates that buying is happening despite the lack of price movement, which often signals that a price increase may soon follow.

- Conversely, if **OBV is decreasing**, it signals **sell pressure**, suggesting that there might be a potential price decline as more volume is associated with downward price movements.

Furthermore, return-based features were calculated:

- **Daily Return**: Percentage change in stock price from one day to the next.
- **Weekly Return**: Percentage change in stock price over a 5-day period.
- **Monthly Return**: Percentage change over a 21-day period (roughly one month).

After feature calculation, any rows containing missing values were dropped to ensure clean data for model training.

Labeling the Data: (Strategy to trade)

Stock movements were categorized into three signals:

- **Buy (1)**: Triggered when the RSI is below 30, MACD crosses above the signal line, or OBV is rising.
- **Hold (0)**: Default label, representing no significant action.
- **Sell (-1)**: Triggered when the RSI exceeds 70, MACD crosses below the signal line, or OBV is dropping.

These conditions were defined based on standard technical analysis principles, where an overbought condition or a downward trend in volume suggests it is time to sell, while an oversold condition or upward trend suggests it is time to buy.

Models:

Several machine learning models were tested for classification:

1. **XGBoost**: A gradient boosting algorithm known for its efficiency and high performance, especially in time series problems.
   - **Parameters**: Number of trees (n_estimators), tree depth (max_depth), learning rate, subsample ratio, and L2 regularization were tuned.
2. **Support Vector Machine (SVM)**: A robust model for classification tasks, implemented using the One-vs-Rest strategy.
   - **Kernel**: Radial Basis Function (RBF) kernel was chosen for its non-linear decision boundaries.
3. **Random Forest**: An ensemble model that builds multiple decision trees and averages their results to reduce overfitting.

   - Hyperparameters such as the number of trees (n_estimators) and maximum depth of the trees were tuned using grid search.

4. **Logistic Regression**: A simple yet effective linear model for binary classification, extended here using a One-vs-Rest classifier to handle the multi-class problem.
   - **Regularization**: L1 regularization was applied to prevent overfitting.
5. **Soft Voting Classifier**: A meta-model that combines predictions from all the above models using the class probabilities for each model and takes the class with the highest average probability as the final prediction.

Model Evaluation:

The models were evaluated on several metrics:

- **Classification Report**: Shows precision, recall, F1-score for each class (Buy, Hold, Sell).
- **Accuracy**: The overall accuracy of the model in predicting the correct class.

Additionally, each model was trained and tested using a train-test split of 80%-20% on the data up to January 1, 2025. The evaluation on a separate test set for the period from January 1, 2025 to January 30, 2025 was also conducted.

Results:

Training Phase:

Model	Precision (Macro Avg)	Recall (Macro Avg)	F1-Score (Macro Avg)	Accuracy
XGBoost	**0.94**	**0.94**	**0.94**	**0.94**
SVM	**0.83**	**0.79**	**0.81**	**0.81**
Random Forest	**0.91**	**0.90**	**0.90**	**0.90**
Logistic Regression	**0.83**	**0.84**	**0.83**	**0.83**
Soft Voting Classifier	**0.92**	**0.92**	**0.92**	**0.92**

Testing Phase:

Model	Precision (Macro Avg)	Recall (Macro Avg)	F1-Score (Macro Avg)	Accuracy
XGBoost	**0.88**	**0.91**	**0.89**	**0.93**
SVM	**0.79**	**0.81**	**0.80**	**0.81**
Random Forest	**0.85**	**0.88**	**0.86**	**0.90**
Logistic Regression	**0.84**	**0.88**	**0.85**	**0.89**
Soft Voting Classifier	**0.86**	**0.90**	**0.88**	**0.91**

The results suggest that the models are generalizing well with no significant overfitting, as indicated by the consistent performance across both training and testing phases.

Limitations:

Despite the successful implementation, several limitations are noted:

1. **Feature Selection**: The technical indicators were selected based on common practice, but other potential features (e.g., external factors like news sentiment) were not considered.
2. **Data Preprocessing**: Missing values were dropped, which could lead to loss of information. Imputation or advanced techniques might have been more beneficial.
3. **Static Model**: The models were trained on historical data and do not account for sudden market changes, such as economic events or external shocks.

Future Work:

To improve the current model and further explore its potential, the following enhancements can be considered:

1. **Deep Learning Models**: Incorporating LSTM (Long Short-Term Memory) networks for sequential pattern recognition could improve the model's ability to understand time series data. LSTMs are known for capturing long-term dependencies, which are vital for financial data.
2. **Reinforcement Learning**: Reinforcement Learning could be explored for dynamic portfolio optimization. Instead of predicting individual stock signals, this approach could optimize buy/sell actions for a set of assets in real-time.
3. **Sentiment Analysis**: Integrating sentiment analysis from financial news and social media (e.g., Twitter, Reddit) could enhance the model by capturing public sentiment about stocks. This would provide a broader view of market psychology and its effect on stock movements.

Conclusion:

This project successfully demonstrated the use of multiple machine learning algorithms to predict stock market signals, which can assist investors in making better trading decisions. The models showed good performance in terms of accuracy and classification metrics. While the existing model can be improved by incorporating deep learning techniques and external factors such as sentiment analysis, it lays the foundation for more advanced stock market prediction systems.

Submitted by,

Manoj Kumar C M

DA24S018