Project Summary Report

Dr. Ru Jia

Friday 14:30-16:30

Min Khant Aung Stephen
ID: 103833225

Contents

Introduction

The FinTech101 Stock Price Prediction System aims to explore the potential of machine learning in forecasting stock prices based on historical data. With the rising influence of financial technology, accurately predicting stock price movements has become a significant focus for both academic research and practical applications. In this project, various machine learning models are applied and optimized to predict future stock prices based on past performance. My approach integrates traditional machine learning techniques, specifically **XGBoost, Support Vector Regression (SVR), and Random Forest models**, to generate predictions with enhanced accuracy.

In addition to historical stock price data, the system leverages technical indicators such as the **Relative Strength Index (RSI)** and **Moving Average Convergence Divergence (MACD)**, both of which are calculated using data from the Federal Reserve Economic Data (FRED) API. These indicators are widely used in financial analysis to detect trends and momentum in stock price movements, adding valuable predictive insights to the model.

To further extend the predictive capabilities and support independent research, additional technical indicators were implemented to refine the model's performance. This extension aims to improve the system's robustness by incorporating diverse data sources and advanced financial indicators, providing a comprehensive, data-driven approach to stock price forecasting.

Overall system architecture

The architecture of the **FinTech101 Stock Price Prediction System** follows a modular and layered design, enabling efficient data processing, feature engineering, model training, and prediction. Here’s a breakdown of each component:

1. Data Loading and Preprocessing

- **Data Sources**: The system fetches historical stock data from Yahoo Finance using yfinance and incorporates macroeconomic indicators (e.g., GDP, inflation, unemployment) from the Federal Reserve Economic Data (FRED) API, utilizing a user-provided API key(data_processing)(main).
- **Technical Indicators**: Essential indicators, including RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), Bollinger Bands, Stochastic Oscillator, ADX (Average Directional Index), and CCI (Commodity Channel Index), are computed and appended to the stock data. These indicators

enhance predictive accuracy by providing insight into market trends and volatility (data_processing).

- **Data Caching**: For efficiency, fetched data is stored locally and reloaded from cache if available, significantly reducing runtime for repeated executions.
- **Data Preparation**: The processed dataset is scaled using MinMaxScaler, and time-series data is organized into sequences (e.g., 60-day lookback) for model training. Data splitting is performed either by ratio or specific date, supporting flexibility in training and testing data configurations(data_processing).

2. Model Training and Prediction

- **Machine Learning Models**: The system employs three models for stock price prediction:
    - **XGBoost**: An ensemble gradient-boosting model, optimized for handling complex data relationships. Key hyperparameters include n_estimators, learning_rate, and max_depth.
    - **Support Vector Regression (SVR)**: A kernel-based model (using an RBF kernel) tuned with C and gamma parameters to capture non-linear patterns.
    - **Random Forest Regressor**: A robust, tree-based ensemble model, configured with n_estimators, max_depth, and min_samples_split (model_operations).
- **Training and Testing**: Data is reshaped to match the input requirements of each model. The models are trained on historical stock data, with technical indicators serving as input features, enhancing each model's predictive capacity.
- **Prediction and Scaling**: Predictions from each model are inverse-transformed to align with the original stock price scale, ensuring accuracy in price level predictions.

3. Visualization and Analysis

- **Prediction Visualization**: Using Matplotlib, the system plots the predictions from XGBoost, SVR, and Random Forest against actual stock prices, allowing for clear visual comparison of model accuracy over time.
- **Technical Indicators Plotting**: The system includes distinct plots for RSI, MACD, and additional indicators (Bollinger Bands, Stochastic Oscillator, ADX, CCI). These visualizations provide insights into market conditions and allow for a more comprehensive analysis of stock price movements alongside model predictions (model_operations).

4. Independent Research and Extended Indicators

- **Extended Technical Indicators**: As part of the independent research, the system integrates additional indicators beyond RSI and MACD. Indicators such as Bollinger Bands, ADX, and CCI offer further predictive power by capturing diverse market characteristics, including volatility and directional strength. These additions enable the model to consider a broader range of market dynamics, potentially increasing prediction accuracy and robustness (data_processing)(model_operations).

5. System Execution and User Interface

- **Main Script Execution**: The primary script, main.py, orchestrates data loading, model training, and visualization. It includes functions to call data preprocessing routines, train each model, and generate predictive plots.
- **Scalability and Modularity**: Each module (data processing, model training, visualization) is organized independently, allowing for easy modifications, scalability, and potential extensions. For instance, additional models or indicators can be integrated by simply extending the corresponding module (main).

This architecture ensures the FinTech101 system is comprehensive and adaptable, capable of handling complex stock price prediction scenarios through a well-rounded machine learning approach enriched by advanced financial indicators

Implemented data processing techniques

The **data processing** methods evolved significantly over the course of this project, with initial implementations focused on preparing stock data for LSTM and GRU models, followed by a refined approach to support machine learning models like XGBoost, SVR, and Random Forest, enhanced with technical indicators. Here's an overview of the approaches in each phase:

Initial Version: LSTM and GRU Model Preparation

In the initial version, the processing steps were tailored for sequential models like LSTM and GRU, which require a time-dependent input structure. Key steps included:

- **Data Loading and Handling Missing Values**: Data was fetched from Yahoo Finance using yfinance, with options to handle missing values by dropping rows, filling them with specified values, or forward- and backward-filling.

- **Data Scaling**: The MinMaxScaler was applied to transform features between 0 and 1, preserving the sequential nature of time-series data. Each feature was scaled individually to avoid disrupting temporal relationships critical for LSTM and GRU performance.
- **Data Splitting for Sequential Learning**: Data was split sequentially or randomly into training and testing sets based on configurable parameters. Additionally, input-output pairs were constructed using sliding windows, making each input sequence represent a defined prediction_days period(data_processing).

Final Version: Machine Learning Models with Technical Indicators

The final version expanded upon the initial processing by integrating various technical indicators and macroeconomic data, aligning with the requirements of XGBoost, SVR, and Random Forest models. The final pipeline included:

- **Technical Indicators Calculation**: Indicators such as RSI, MACD, Bollinger Bands, Stochastic Oscillator, ADX, and CCI were calculated and appended to the stock data. These indicators were essential in capturing complex financial signals, which helped enhance predictive accuracy across models (data_processing).
- **Macroeconomic Data Integration**: Additional macroeconomic features (e.g., GDP, inflation, unemployment) were fetched using the FRED API and joined with the stock data, filling missing values with forward filling for consistency.
- **Scaling and Data Preparation for ML Models**: The selected features, including the technical and macroeconomic indicators, were scaled and prepared for model training. The system used time-windowed sequences for input, supporting non-sequential ML models by reshaping the data to align with their requirements.
- **Flexible Splitting Methods**: The data was split based on time ratio or a specific date, ensuring flexibility for model evaluation across different periods (data_processing)(data_processing).

This evolution in data processing enabled the transition from sequential to machine learning models with enhanced predictive features, supporting more robust and diversified model implementations in the final system.

Experimented machine learning techniques

In this project, the machine learning techniques evolved through multiple stages, with initial experiments using deep learning models like LSTM and GRU, followed by a shift to machine learning models in the final implementation for increased versatility and feature support. Here’s a breakdown:

Initial Version (Up to Task B6): Deep Learning Models (LSTM, GRU)

The initial focus was on using Recurrent Neural Networks (RNNs) to capture the temporal patterns in stock prices. Key models experimented with included:

- **LSTM (Long Short-Term Memory)**: LSTMs are well-suited for time-series data as they can handle long-term dependencies, crucial for capturing extended trends in stock prices. Configurations allowed for multiple layers and tunable units per layer, enhancing the model’s complexity.
- **GRU (Gated Recurrent Units)**: GRUs are similar to LSTMs but with a simplified architecture, making them faster to train with comparable performance on time-series tasks.
- **Bidirectional Layers**: Both LSTM and GRU models were experimented with in bidirectional configurations (BiLSTM and BiGRU) to account for patterns from both past and future data points within the training sequences.
- **RNN Variants**: Standard RNNs were also experimented with for baseline comparisons, as well as hybrid architectures combining different RNN types.
- **Hyperparameter Tuning**: Each model's depth, layer size, and dropout rates were adjusted to achieve optimal performance in stock price prediction.

Each model was trained using mean squared error as the loss function and adam optimizer, aiming to minimize the error in predictions of stock price trends (model_operations).

Final Version (Task B7): Machine Learning Models with Feature Support

In the final version, the system transitioned to machine learning models that support a broader range of engineered features, including technical indicators and macroeconomic data:

- **XGBoost**: An ensemble model utilizing gradient boosting, XGBoost was selected for its efficiency in handling structured data and complex feature interactions. Hyperparameters such as n_estimators, learning_rate, and max_depth were tuned to optimize predictive accuracy.
- **Support Vector Regression (SVR)**: SVR, with an RBF kernel, was implemented to capture non-linear relationships between the input features and stock prices. SVR is well-suited for continuous outputs and was configured with parameters such as C and gamma for best fit.

- **Random Forest Regressor**: A robust ensemble model using multiple decision trees, Random Forest was added to provide reliable predictions by aggregating results from numerous trees. Parameters like n_estimators, max_depth, and min_samples_split were optimized to balance model complexity and performance.

Model Comparison and Prediction Evaluation

The machine learning models were evaluated based on their predictive accuracy and scalability with extensive feature sets, including technical indicators and external data. This approach allowed for refined control over model tuning and supported more complex feature engineering, ultimately enhancing the system's predictive capability and robustness(model_operations).

This evolution from RNN-based models to structured machine learning models facilitated improved accuracy and adaptability, aligning with the project's goal of producing a comprehensive stock price prediction system.

Details of the Independent Research developed (if applicable)

The independent research in this project extended beyond standard technical indicators to implement additional financial indicators that offer deeper insights into market conditions and can potentially enhance stock price prediction accuracy. Here's an overview of the extended features integrated as part of the independent research:

1. Additional Technical Indicators

- **Bollinger Bands**: Calculated to reflect price volatility by plotting upper and lower bands around a moving average, this indicator helps identify overbought or oversold market conditions. The system calculates the middle (moving average), upper, and lower bands, which are used to assess price trends relative to volatility.
- **Stochastic Oscillator**: This momentum indicator compares a stock's closing price to its price range over a certain period, helping detect overbought and oversold levels. The Stochastic %K and its moving average (Stochastic %D) were calculated and used to refine prediction models.
- **Average Directional Index (ADX)**: Used to evaluate the strength of a trend, the ADX indicator allows the model to understand the trend's durability, whether bullish or bearish, which is crucial for trend-following algorithms.

- **Commodity Channel Index (CCI)**: CCI measures the current price level relative to an average price over a specified period. This indicator aids in identifying new trends or warning of extreme market conditions.

These indicators were calculated directly from historical price data, providing additional predictive features for the models. Their integration aimed to improve model performance by enabling the system to capture a broader array of market dynamics and trends that single-point price data might miss(data_processing)(main).

2. Integration with Macroeconomic Data

- Using the FRED API, key macroeconomic indicators—such as **GDP**, **Inflation Rate**, and **Unemployment Rate**—were incorporated. These features provide contextual data that can be valuable for capturing external economic influences on stock prices, offering a more holistic approach to prediction.
- The system automatically joins this macroeconomic data with stock data, with missing data filled forward to maintain continuity and ensure feature consistency across the dataset(data_processing).

3. Analytical Visualization Enhancements

- Beyond stock price predictions, the system includes enhanced visualizations for each indicator, such as MACD, RSI, Bollinger Bands, and ADX. These plots provide both an interpretive tool for market analysis and an informative method to validate the model's predictive outputs.
- The combined visualization of various indicators on the same data timeline supports analytical verification, allowing users to cross-reference model predictions against technical and economic indicators in real-time (model_operations).

This independent research extended the project's functionality and enriched the model’s predictive framework, potentially increasing the robustness and accuracy of stock predictions through diversified financial and economic data. This enhancement aligns with the overall goal of developing a comprehensive and advanced stock price prediction system

Scenarios/examples to demonstrate how the system works

The FinTech101 Stock Price Prediction System provides predictive insights for stock prices based on historical data, technical indicators, and macroeconomic factors. Here are a few example scenarios to illustrate its use:

Scenario 1: Predicting the Stock Price of a Specific Company

1. **Objective**: Predict the stock price of the Commonwealth Bank of Australia (CBA.AX) for the next 30 days.
2. **Process**:
    - The user inputs the ticker symbol CBA.AX and sets the prediction parameters, such as the date range for historical data (e.g., January 2020 to August 2023).
    - The system fetches historical data from Yahoo Finance, processes it, and calculates indicators like RSI, MACD, and ADX. Macroeconomic data (e.g., GDP, inflation, unemployment) is also incorporated to provide additional context.
    - The data is split into training and testing sets, and the models (XGBoost, SVR, and Random Forest) are trained on these data features.
    - Predictions are generated for the testing period and visualized alongside the actual stock prices.
3. **Output**:
    - A plot shows the actual vs. predicted stock prices, allowing the user to visually assess prediction accuracy.
    - Technical indicators such as RSI and MACD are also displayed, showing underlying trends and confirming alignment with the model's predictions.
    - The user can observe how macroeconomic data might affect stock prices and verify if predictions capture significant trends influenced by broader economic conditions.

Scenario 2: Analyzing Market Volatility with Technical Indicators

1. **Objective**: Assess stock volatility using Bollinger Bands for a potential trading decision.
2. **Process**:

- The user selects a specific stock (e.g., Amazon, ticker AMZN) and sets the desired date range.
- The system calculates Bollinger Bands, showing upper and lower limits around the moving average. The user observes periods where the stock price touches or exceeds these bands, indicating potential buying or selling points due to high or low volatility.

3. **Output**:
   - The Bollinger Bands plot allows the user to quickly interpret volatility by identifying the upper and lower thresholds.
   - By observing how frequently the price approaches these thresholds, the user can gain insights into potential price reversals or continuations, informing trade decisions based on market trends.

Scenario 3: Comparing Prediction Accuracy Across Models

1. **Objective**: Evaluate the performance of XGBoost, SVR, and Random Forest on the same dataset to select the best model.
2. **Process**:
   - Using historical data for a specific stock (e.g., Apple, AAPL), the system trains all three models (XGBoost, SVR, and Random Forest) and produces stock price predictions for the testing period.
   - Each model's predictions are plotted against the actual prices to compare accuracy visually.
3. **Output**:
   - A comparative plot shows the predictions from each model alongside actual prices.
   - By observing which model's predictions closely align with the actual stock price movements, the user can determine the most effective model for this dataset, potentially applying it to other stocks for enhanced predictive reliability.

Scenario 4: Understanding Market Trends with Multi-indicator Analysis

1. **Objective**: Use multiple indicators to make an informed prediction for a high-growth stock in a volatile market (e.g., Tesla, TSLA).
2. **Process**:

- The user inputs the stock ticker TSLA and sets a recent date range for a rapidly changing market.
- The system calculates a range of indicators (e.g., RSI, MACD, ADX, Stochastic Oscillator) that provide insights into trend strength, momentum, and volatility.
- Using these indicators, the user can interpret whether recent price movements are likely to continue or reverse, making informed decisions on potential buy/sell actions.

3. **Output**:
   - Indicator plots display overbought and oversold levels, trend strength, and the stock's momentum. For example, an RSI above 70 may indicate an overbought condition, while a high ADX score signifies a strong trend.
   - By combining these indicators, the user gains a comprehensive view of the stock's potential direction and strength, which aids in forecasting price movements and determining optimal entry or exit points in trading.

Each scenario showcases the system's ability to provide detailed stock price predictions, analyze volatility, and leverage technical indicators to inform investment decisions based on data-driven insights.

Critical analysis of the implementation

The FinTech101 Stock Price Prediction System improved significantly from the B6 version to the final version, with each phase introducing strengths and addressing limitations.

1. **Data Processing and Feature Engineering**:
   - **Strengths**: The final version added technical indicators (Bollinger Bands, ADX, Stochastic Oscillator) and macroeconomic data (GDP, inflation, unemployment), offering richer insights and a more comprehensive basis for predictions.
   - **Limitations**: Handling missing values in macroeconomic data introduced complexity, and technical indicators required high data integrity, increasing sensitivity to gaps or inconsistencies.
   - **Comparison**: Compared to B6, the final version's data processing is more robust and versatile but also more complex and computationally demanding.

2. **Model Selection and Experimentation**:
    - **Strengths**: Machine learning models (XGBoost, SVR, Random Forest) in the final version provided flexibility with diverse data features and allowed effective hyperparameter tuning, leading to stable predictions.
    - **Limitations**: Transitioning from LSTM/GRU models reduced the system's ability to capture long-term dependencies, and SVR's non-linear kernel made it computationally intensive on larger datasets.
    - **Comparison**: B6's deep learning models were better suited for time-series dependencies, but the final version achieved higher accuracy and adaptability on a broader feature set.
3. **Technical Indicators and Macroeconomic Integration**:
    - **Strengths**: Integrating indicators like CCI and Stochastic Oscillator enriched the model's context-awareness, and macroeconomic data added resilience to economic shifts.
    - **Limitations**: Additional indicators increased data dependencies, and macroeconomic data didn't always enhance accuracy, as stocks can behave independently of economic factors.
    - **Comparison**: B6's simpler, indicator-free model was easier to deploy, while the final version's advanced indicators provided deeper insights, albeit with added complexity and data sensitivity.
4. **Visualization and Interpretation**:
    - **Strengths**: Detailed visualizations of indicators and predictions in the final version provided users with interpretative depth, helping identify market signals.
    - **Limitations**: The added plots could overwhelm non-expert users and, in cases of conflicting indicators, create uncertainty.
    - **Comparison**: B6's simpler visualizations were straightforward, while the final version's enhanced visuals improved interpretability but increased interface complexity.

Overall Summary: The transition from B6 to the final version improved prediction accuracy, interpretability, and data integration, though it also introduced new complexities and computational demands. While B6's simpler model was faster to deploy, the final version's rich feature set made it more suitable for in-depth stock analysis, especially for volatile or economically sensitive stocks. This transition underscores the balance between model sophistication and ease of use, with the final version offering a more nuanced, data-rich framework for stock prediction.

Summary/Conclusion.

The FinTech101 Stock Price Prediction System evolved from a simpler model-based approach (using LSTM and GRU) in the B6 version to a robust final version that integrates structured machine learning models (XGBoost, SVR, and Random Forest) with comprehensive technical indicators and macroeconomic data. This enhancement has improved predictive accuracy and interpretability, providing users with actionable insights through features like RSI, MACD, and Bollinger Bands visualizations.

While the final version offers a more versatile framework for stock analysis, the added complexity and sensitivity to data quality bring new challenges, including higher computational demands and interpretative complexity. Nevertheless, this system effectively demonstrates the synergy between machine learning, financial indicators, and external economic data, creating a strong foundation for advanced stock price forecasting in dynamic market environments.