STOCK PREDICTION USING PYTHON

COS30018 Semester 2 2024

PROJECT SUMMARY REPORT

Muhammad Ali – 103960437

Contents

1. Introduction

This project aims to develop a stock price prediction system using a combination of deep learning models (LSTM, GRU, Backpropagation) and statistical models (SARIMA). The system is designed to predict stock prices over multiple future steps, using a comprehensive dataset from Yahoo Finance. The primary goal is to create an ensemble model that combines predictions from different techniques for improved accuracy and robustness.

2. Overall System Architecture

The architecture of the system is divided into three major components:

- **Data Processing:** Involves loading, cleaning, and preparing stock data for model training.

- **Model Training:** Includes training different models—LSTM, GRU, Backpropagation, and SARIMA—individually and combining them to form an ensemble.

- **Visualization and Evaluation:** Plots the results of the predictions and evaluates model performance.

3. Data Processing Techniques

1. load_and_process_data_with_gap()

- **Purpose**: Loads stock data from Yahoo Finance, handles missing values, splits data into training and testing sets, and applies scaling if needed.
- **Key Features**: Can create a gap between training and testing data for better simulation of real-world scenarios.
- **Output**: Returns processed training and testing data, along with the original dataframe.

2. prepare_data_for_model()

- **Purpose**: Converts the input data into 3D shape suitable for deep learning models like LSTM and GRU.
- **Output**: Returns reshaped data ready for model training.

3. prepare_multistep_data()

- **Purpose**: Prepares data for making predictions over multiple future steps.
- **Output**: Creates 3D input data and corresponding multistep targets.

6. create_backpropagation_model()

- **Purpose**: Builds a feedforward neural network (backpropagation model) with specified hidden layers and activation functions.
- **Output**: A compiled Keras model ready for training.

7. create_dl_model()

- **Purpose**: Creates a custom deep learning model (LSTM, GRU, Dense layers) based on user-defined parameters.
- **Output**: A compiled model tailored for time series forecasting.

4. Implemented Machine Learning Techniques

The following machine learning models were implemented:

1. Deep Learning Models:
- LSTM (Long Short-Term Memory): A recurrent neural network model that captures long-term dependencies in time series data.
- GRU (Gated Recurrent Unit): Another RNN variant that is computationally efficient compared to LSTM.
- Backpropagation Neural Network: A feedforward neural network trained using backpropagation.

2. Statistical Model:
- SARIMA (Seasonal Auto-Regressive Integrated Moving Average): A classical statistical model designed to capture seasonality and trends in time series data.

These models were trained separately and later combined to form an ensemble model that uses a weighted average of predictions from the individual models

5. Model Evaluation and Visualization

The following images display the evaluation results and predictions made by the models.

Image 1: Candlestick Chart of KO Stock Prices

Image 2: Boxplot of KO Stock Closing Prices

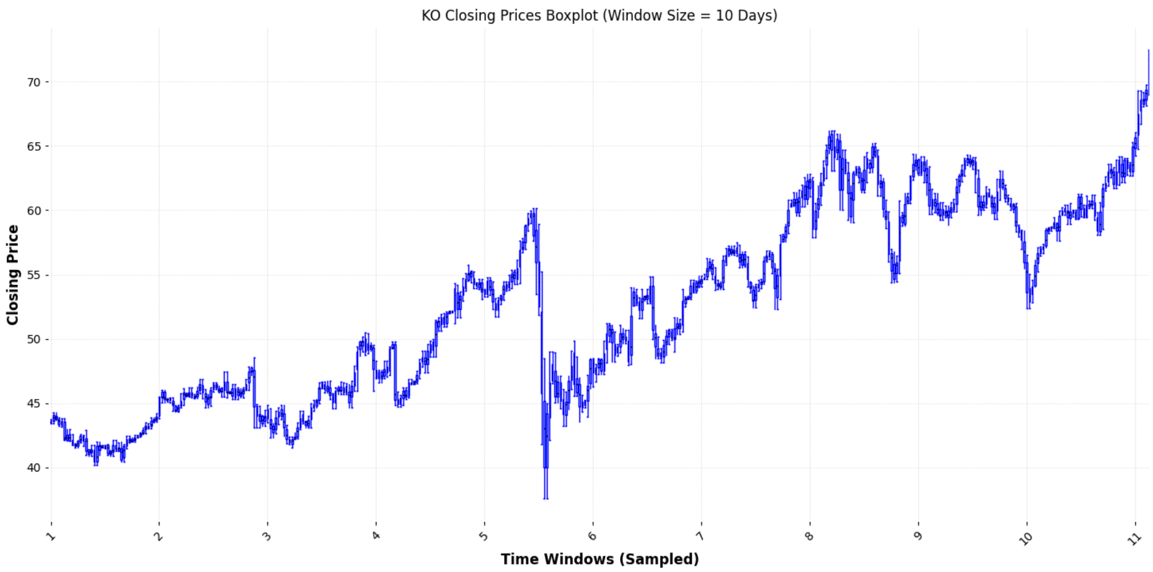
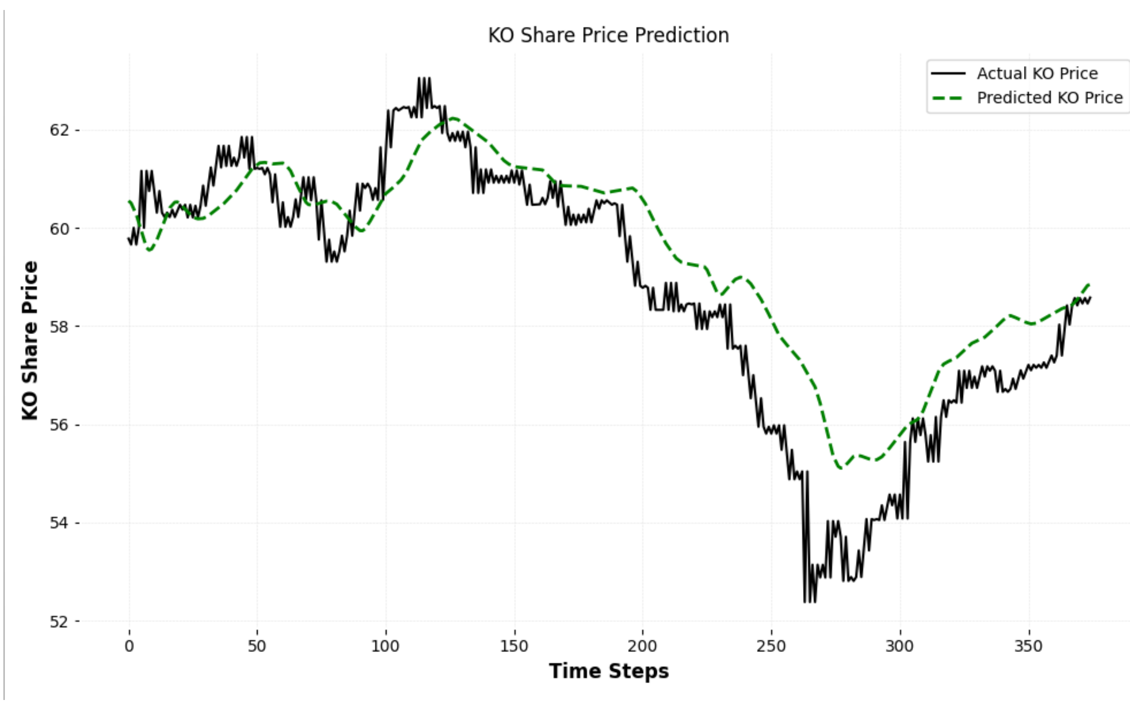


Image 3: Ensemble Model Predictions



6. Scenarios/Examples

The following workflow illustrates a typical run of the system:

Start: Load data and prepare it for training.
Step 1: Train LSTM and GRU models using the reshaped 3D input data.
Step 2: Train the backpropagation model using flattened input data.
Step 3: Train the SARIMA model using the unscaled 1D data series.
Step 4: Generate predictions for each model.
Step 5: Combine predictions into an ensemble.
Step 6: Evaluate performance and visualize results.

<u>To demonstrate functionality, the model uses a series of steps:</u>

1. Load the stock data for a given time range, ensuring that missing values are handled, and the data is scaled appropriately.
2. Train the individual models (LSTM, GRU, Backpropagation, SARIMA) on the training dataset.
3. Combine predictions from these models to generate ensemble predictions.
4. Plot the results using visualizations like candlestick charts, boxplots, and predicted vs. actual prices over time.

7. Critical Analysis of the Implementation

Strengths:

- Multiple prediction methods: By combining deep learning and statistical models, the system captures complex patterns as well as seasonality and trends in stock prices.

- Data handling: The system effectively handles missing values and normalizes data for optimal model performance.

- Ensemble approach: The ensemble model improves prediction accuracy by leveraging the strengths of individual models.

Weaknesses:

- Computational complexity: Training multiple models can be time-consuming and resource-intensive.
- Overfitting risk: The deep learning models, especially LSTM and GRU, can be prone to

overfitting if not carefully tuned.
- Limited feature set: Only historical prices are considered as features, without incorporating external factors like market news or macroeconomic indicators.

8. Summary/Conclusion

The stock prediction system successfully combines deep learning models (LSTM, GRU, Backpropagation) and a statistical model (SARIMA) to predict future stock prices. The ensemble approach demonstrates improved accuracy and robustness compared to individual models. Future enhancements could include incorporating more features and experimenting with advanced ensemble techniques, such as stacking or boosting, to further improve prediction performance.

9. References

Berejnoi, A. (2018, September 12). *How to implement backpropagation with numpy*. Medium. https://medium.com/@andresberejnoi/how-to-implement-backpropagation-with-numpy-andres-berejnoi-e7c14f2e683a

Brownlee, J. (2016, August 31). *How to implement the backpropagation algorithm from scratch in Python*. Machine Learning Mastery. https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/

Bharadwaj, A. (2018, October 11). *Predicting stock price using machine learning and deep learning techniques in Python*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/