TIME SERIES ANALYSIS AND FORECASTING FOR STOCK MARKET

Presented By

Bolle Nihanth Bhargav

Introduction

Stock price forecasting is one of the most challenging tasks in financial data analysis due to its high volatility and non-linear patterns. Accurate forecasting helps investors, traders, and businesses make better financial decisions.

This project integrates **classical statistical methods (ARIMA, SARIMA, Prophet)** with **deep learning (LSTM)** to predict Apple Inc. (AAPL) stock prices. By combining these models, we aim to analyze short-term and long-term patterns, compare their performance, and generate reliable forecasts.

Dataset Used

- **Source:** Yahoo Finance / Kaggle (AAPL stock data)
- **Time Period:** 2015 – 2024 (~2,500 business days)
- **Features:**
    - Date
    - Close Price (used as target for forecasting)
- **Preprocessing:**
    - Missing values handled using forward fill
    - Converted Date column to time index
    - Scaled prices (for LSTM)
    - Train-test split for evaluation

Methodology

ARIMA Model

- **Concept:** AutoRegressive Integrated Moving Average.
- **Objective:** Captures linear trends and short-term dependencies.
- **Parameter Selection:** Grid search over (p, d, q).
- **Evaluation Metric:** AIC for best model selection.

SARIMA Model

- **Concept:** Seasonal ARIMA adds seasonal differencing.
- **Objective:** Captures seasonality in stock prices.
- **Parameter Selection:** Grid search over (p, d, q) × (P, D, Q, m).
- **Challenges:** Seasonal patterns in stock data were weak, leading to limited improvement.

Prophet Model

- **Concept:** Additive model combining **trend + seasonality + holidays**.
- **Advantages:** Handles missing dates, outliers, and flexible seasonality.

- **Implementation:** Daily + yearly seasonality enabled.
- **Output:** Smooth long-term predictions, interpretable trends.

LSTM Model

- **Concept:** Long Short-Term Memory (RNN variant).
- **Objective:** Captures non-linear and long-term dependencies in stock data.
- **Pipeline:**
  - Normalize data using MinMaxScaler.
  - Create input sequences (past 60 days → next day prediction).
  - Train on Close price using LSTM layers + Dense output.
- **Loss Function:** Mean Squared Error (MSE).

Integration Pipeline

- Load and preprocess stock data.
- Train ARIMA, SARIMA, Prophet, and LSTM models.
- Forecast future prices (30–180 business days).
- Save forecasts in CSV files.
- Generate combined comparison plots of all models.
- Evaluate models using MAE, RMSE, and MAPE.

Results

Model Performance (Test Data)

- **ARIMA:** Moderate accuracy, weak for long-term.
- **SARIMA:** Limited improvement due to weak seasonality.
- **Prophet:** Strong long-term trend forecasting, interpretable results.
- **LSTM:** Best short-term accuracy, captured non-linear dynamics.

Evaluation Metrics (Example Values)

- ARIMA → MAE: ~20.0, RMSE: ~26.2
- SARIMA → Limited (seasonality weak)
- Prophet → MAE: ~5.5, RMSE: ~7.6
- LSTM → Lowest MAE & RMSE among all

Observations

- LSTM performed best for short-term, Prophet gave smooth long-term predictions.
- ARIMA/SARIMA worked as baselines but struggled with volatility.

Applications

- **Finance & Trading:** Assist traders in stock price forecasting.
- **Risk Management:** Identify future price volatility.
- **Investment Strategy:** Support portfolio optimization.
- **Research & Education:** Benchmarking classical vs deep learning models.

Conclusion & Future Work

Achievements

- Built and compared **ARIMA, SARIMA, Prophet, and LSTM** models.
- Conducted comprehensive evaluation using MAE, RMSE, and MAPE.
- Generated future forecasts (30–180 days).
- Found LSTM to be the most accurate, Prophet most interpretable.

Limitations

- Dataset limited to single stock (AAPL).
- Market factors (news, sentiment, macroeconomics) not included.
- Hardware constraints limited deep learning hyperparameter tuning.

Future Work

- Extend to multiple stocks (MSFT, AMZN, TSLA).
- Include external features (sentiment, indicators, global events).
- Explore hybrid/ensemble models (Prophet + LSTM).
- Deploy as a real-time forecasting web app using Flask/Streamlit.