G R O U P 1 0

Machine Learning for Time Series Data

Short Term Stock Price Prediction Using LSTM Model

Overview

01 Data Collection and Preprocessing

02 Exploratory Data Analysis (visualizations)

03 Preparing Data for Model Training

04 Building and Training the LSTM Model

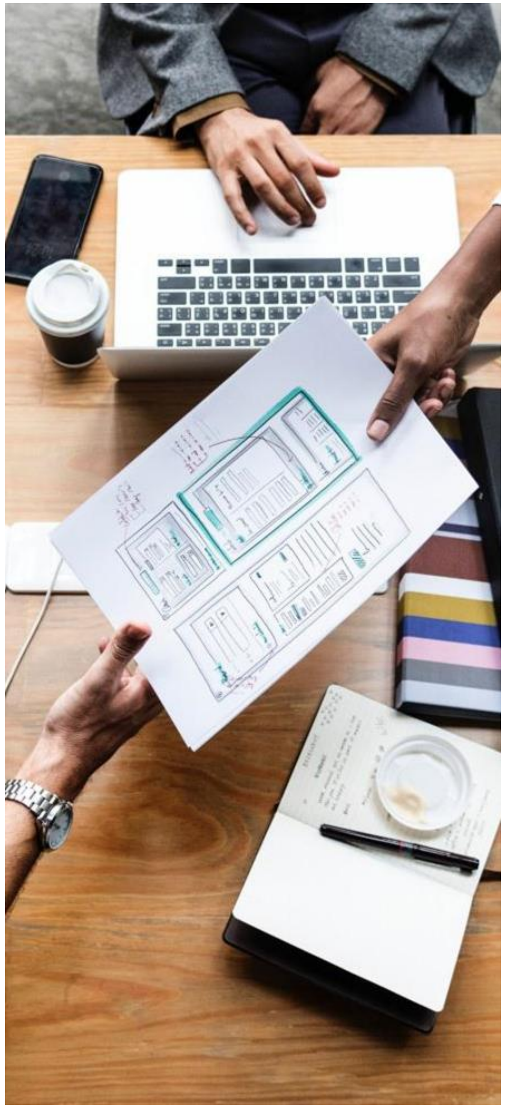
05 Model Evaluation and Results

06 Other considerations & Future Work

Data Collection and Preprocessing

Daily stock data for BEP was fetched from Yahoo Finance. The dataset is then cleaned, transformed, and enriched with various features, such as daily returns, volume change, and moving averages. This data is saved as a CSV file for ease of use and further validation.

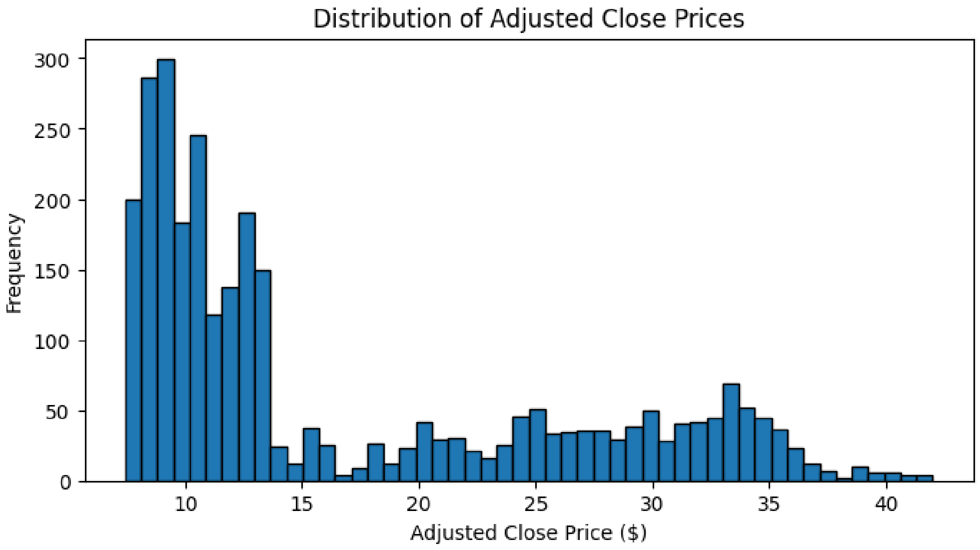
- **Data Collection**: fetching BEP stock data from Yahoo Finance between 2011 and 2024.
- **Data Preprocessing**: cleaning the data involves renaming columns, dropping null values, and generating useful features like returns and moving averages.
- **Feature Engineering**: calculated short-term and long-term moving averages, which we use later

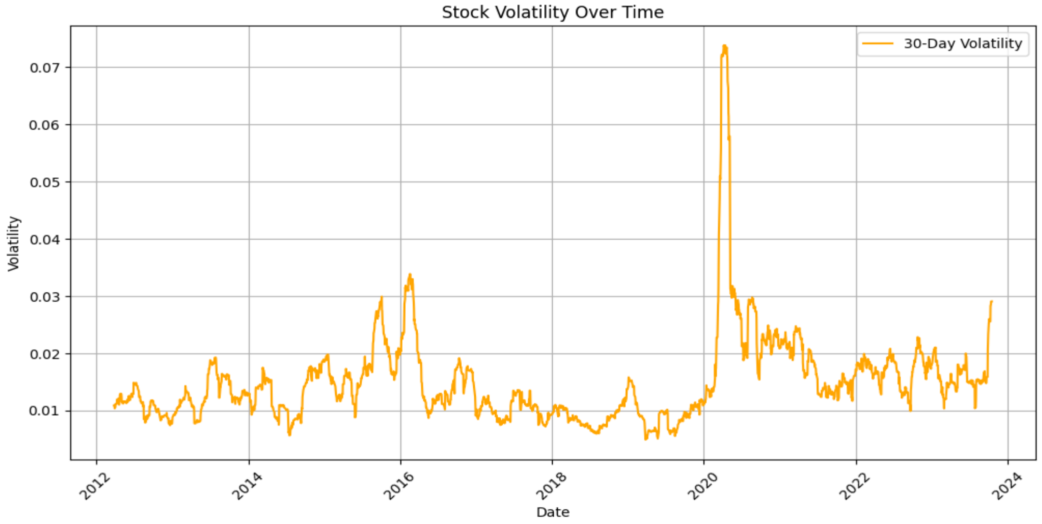


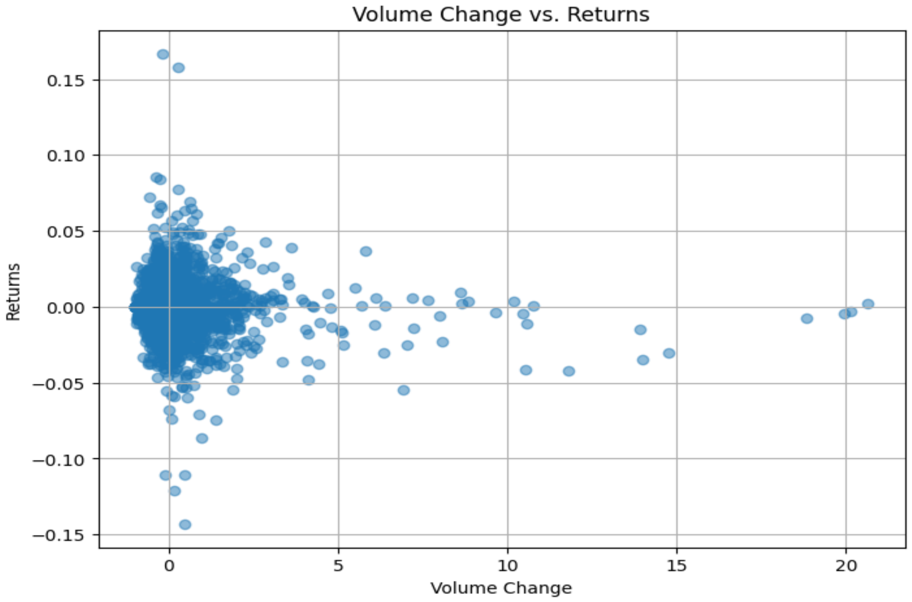
Exploratory Data Analysis (EDA)

EDA helps us understand our dataset visually. Below, visualize was provided to show the distribution of adjusted close prices, rolling volatility, and volume changes versus returns. These insights allow us to better understand how the stock behaves over time.

- **Distribution of Adjusted Close Prices**: Helps identify typical price ranges.
- **Volatility Plot**: Rolling standard deviation to measure the stock's price variability.
- **Moving Averages**: Comparing short-term (SMA 10) and long-term (SMA 50) moving averages to determine possible buy and sell signals.

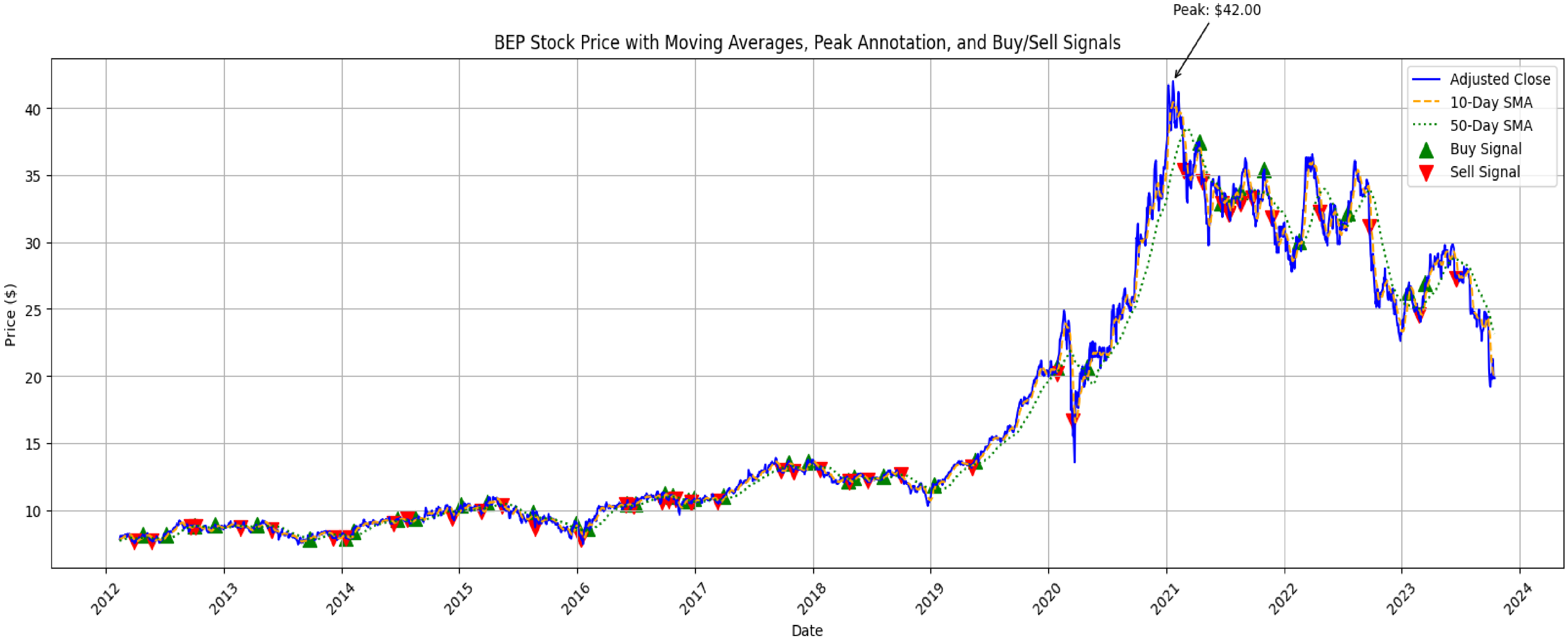






4

Exploratory Data Analysis (EDA)



Model Selection

- LSTM (Long Short-Term Memory networks) is suitable for **time-series data** due to its ability to capture **long term dependencies**, handles **non-linear and noisy data**
- Designed to learn **sequential data** patterns by retaining relevant information from previous time steps through their **memory cells**.
- The **gated mechanism** in LSTM (input, forget, and output gates) helps the model decide what information to keep or forget, which improves performance on time-series data with varying trends.

Training Details

- Training Details: data split into 80% training and 20% testing
- Training with Early Stopping to prevent overfitting
- Hyperparameters: number of epochs, batch size, etc.

Model Design

- Bi-directional LSTM Layers: processes the input data in both directions (forward and backward). This helps the model learn from both past and future context in the sequence.
- Dropout Regularization: used to prevent overfitting and smooth convergence in neural networks
- Output Layer: single prediction output

Data

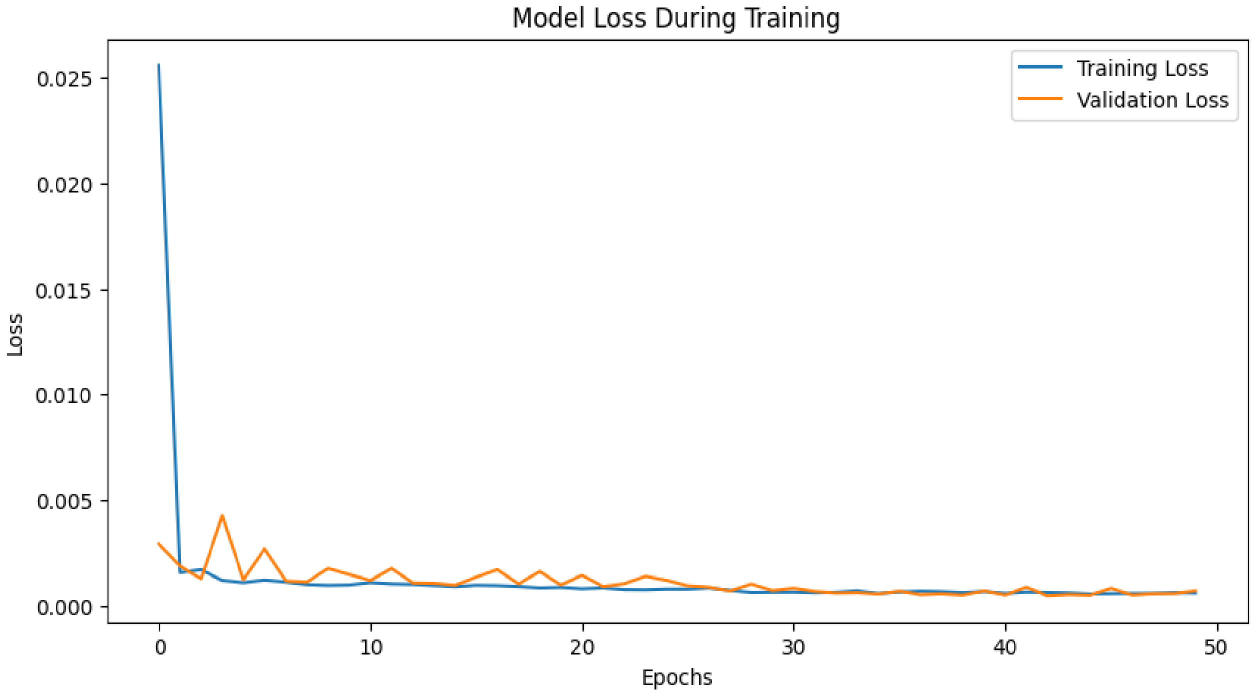
- Scaling both features and the target variable using MinMaxScaler to prepare them for the LSTM model.

5

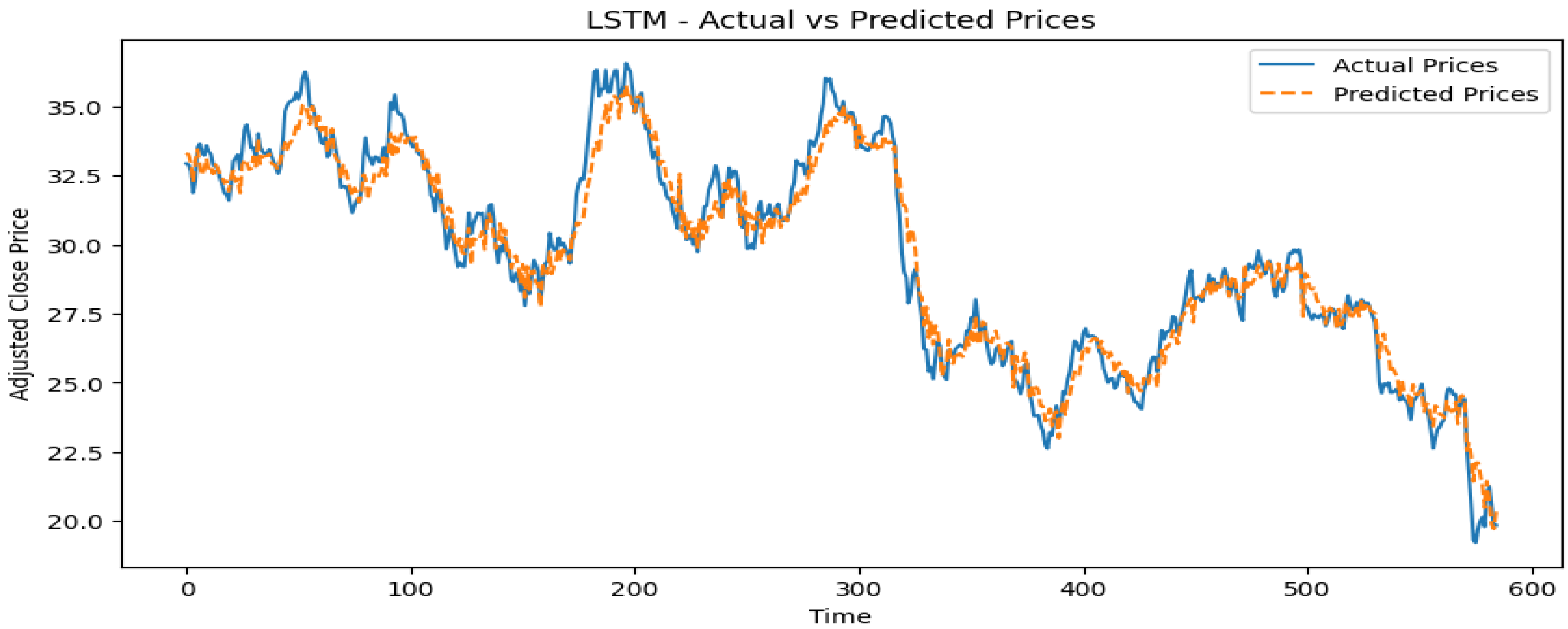
Training the Model

Plot training and validation loss during training to visually inspect overfitting or underfitting:

The sharp decrease in the blue line at the beginning indicates that the model quickly learned to minimize error, the orange line shows the error on the validation set, it fluctuates more at first, indicating that the model is adapting, but it eventually stabilizes at a low value, closely tracking the training loss, suggesting that the model is not overfitting.



Model Prediction



The predicted prices (orange line) closely follow the actual prices (blue line). The LSTM model was able to capture the general trends and fluctuations of the stock, although there are minor discrepancies in some peaks and troughs. Overall, the high similarity between the two lines indicates that the model has performed well in forecasting the future stock prices.

9

Model Evaluation

- A value of 0.5538 for MSE is relatively low, suggesting that the model has captured the stock's trends well
- A MAE of 0.5724 shows that the average prediction error is quite small.
- R-squared is 0.9624, a score close to 1 indicated that the model has strong predictive power.

Performance Metrics

Mean Squared Error

0.5538

Mean Absolute Error

0.5724

R-squared Score

0.9624

Judging from the Performance metrics, LSTM models are effective for stock price forecasting. The model captured historical trends well with a high R-squared value.

Other Considerations and Future Work

LSTM Model

- **Strengths**:
  - Can **capture non-linear relationships** and **learn patterns** across longer time horizons.
  - Works well with **multivariate data**, incorporating **technical indicators, volume, or market sentiment**.
  - More adaptable to **volatile markets** than ARIMA.
- **Limitations**:
  - **Requires extensive data**: stock price prediction with LSTM requires large datasets for meaningful results.
  - **Overfitting risk**: LSTM models can fit noise in the data, reducing real-world predictive power.
  - **Computationally intensive**: training LSTM models is resource-intensive, especially for short-term predictions.

ARIMA Model

- **Strengths:**
  - **Works well with linear patterns** and mean-reverting trends (like volatility in exchange rates or bonds).
  - Clear and Transparent Forecasting: ARIMA's results are **easy to interpret** for investors and analysts because it breaks down the time serious components and the parameters (p, d, q) provide sights on past behavior, makes it a great tool for **reporting and explaining trends** to stakeholders.
- **Limitations:**
  - Limited by stationarity: Stock prices are usually non-stationary.
  - Univariate nature: ARIMA only considers historical prices, ignoring other key factors like volume, sentiment, or macroeconomic data.

Future Work

- **LTSM model**
  - Incorporate External Data
  - Use more timesteps to allow the model to learn more complex patterns over longer horizons
  - Explore Transfer Learning: train the model on multiple stocks, use the knowledge gained from one stock to forecast others
- **ARIMA + LSTM hybrid approach**
  - Benefit from both **linear and non-linear dependencies** in stock data.
  - Can further **improve signal-based predictions** by leveraging the strengths of both models.

Thank you for your time!

Group 10

Yunnan Li

Sources:
https://www.kaggle.com/
https://unsplash.com/
YouTube
Python Machine Learning by Example, fourth edition, Yuxi (Hayden) Liu
Hans on Machine Leaning with Scikit, Keras & TensorFlow, Aurelien Geron