Overview

This project implements a **univariate Long Short-Term Memory (LSTM)** neural network to forecast **daily stock closing prices** using historical data. The model is trained on past closing prices and evaluated on unseen data using standard regression metrics.

The goal is to demonstrate how deep learning models, specifically LSTMs, can capture temporal dependencies in time-series data such as stock prices.

Features

- Univariate time-series forecasting using LSTM
- Sliding window sequence generation
- Data normalization using Min-Max scaling
- Model evaluation using RMSE and MAE
- Visualization of predictions and training history
- Reproducible and easy-to-run code

Technologies Used

- **Python 3.x**
- **TensorFlow / Keras**
- **Pandas**
- **NumPy**
- **Matplotlib**
- **Scikit-learn**

Dataset

- **Source:** Kaggle
- **Dataset Name:** Apple Stock Price (1980–2021)
- **Link:** https://www.kaggle.com/datasets/meetnagadia/apple-stock-price-from-1980-to-2021
- **Target Variable:** Close

The dataset contains daily stock market data including Open, High, Low, Close, Adjusted Close, and Volume.
Only the **Close price** is used in this project, making it a **univariate time-series problem**.

Methodology

1. Load historical stock price data from CSV
2. Select the closing price and normalize it
3. Convert the time series into sliding window sequences
4. Build and train an LSTM model using TensorFlow/Keras

5. Evaluate the model using RMSE and MAE on a hold-out test set
6. Visualize predictions vs actual values and training loss

How to Run the Project

1. Clone the Repository

```
git clone https://github.com/your-username/LSTM-Stock-Forecasting.git
cd LSTM-Stock-Forecasting
```

2. Install Dependencies

```
pip install numpy pandas matplotlib scikit-learn tensorflow
```

3. Download the Dataset

- Download the dataset from the Kaggle link above
- Place it inside the data/ directory

4. Run the Code

You can run either:

Jupyter Notebook

```
jupyter notebook lstm_stock_forecasting.ipynb
```

OR Python Script

```
python lstm_stock_forecasting.py
```

Outputs

- **RMSE and MAE values** printed in the console
- **Prediction vs Actual Price plot**
- **Training and Validation Loss plot**

These outputs help evaluate model performance and learning behavior.

Evaluation Metrics

- **RMSE (Root Mean Squared Error)**
- **MAE (Mean Absolute Error)**

RMSE is typically greater than MAE because it penalizes larger errors more strongly

Conclusion

The LSTM model successfully learns temporal patterns in historical stock prices and provides reasonable forecasts for future closing prices. This project demonstrates the effectiveness of deep learning for time-series prediction tasks.