Stock Price Prediction using TensorFlow

A Deep Learning & NLP Project

[Your Name], [Your Roll Number]

[Your Institution], [Your Department]

[Current Semester]

[Date]

Agenda

- Introduction & Motivation
- Problem Statement & Dataset
- Data Processing & EDA
- Model Design & Training
- Evaluation & Results
- Deployment & Future Scope
- Challenges & Conclusion

Introduction & Motivation

Stock price prediction is the act of forecasting the future value of a company's stock or other financial instrument traded on an exchange. It's crucial for investors, financial institutions, and businesses alike. However, the stock market's inherent non-linearity, stochasticity, and high volatility make accurate forecasting a formidable challenge, requiring sophisticated analytical models.

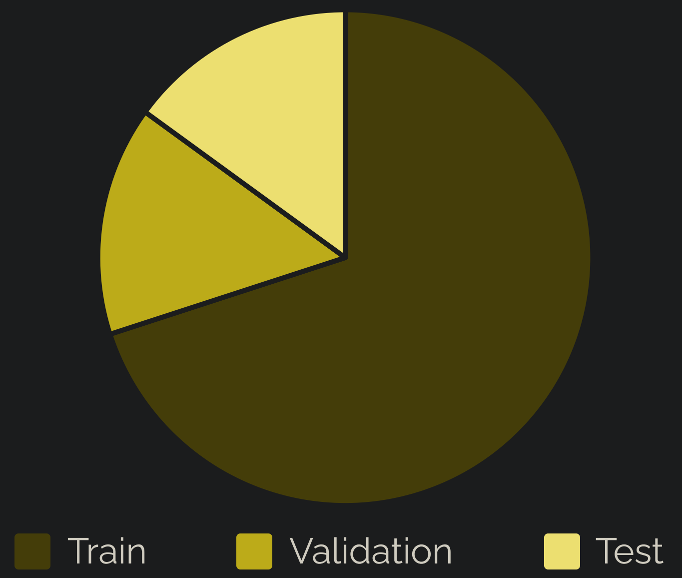
Problem Statement & Dataset

Problem Definition

Our objective is to predict the next-day closing price of a selected stock utilizing historical time series data. The target variable is the **closing price**, with input features including **Open, High, Low, Close, and Volume**. This project focuses on **short-term forecasting** to inform daily trading decisions.

Dataset Overview

The dataset was sourced from **Yahoo Finance**, comprising daily stock data over a five-year period. It includes approximately 1,258 records with 6 features. We employed a **70% train, 15% validation, and 15% test split** for robust model evaluation.



Data Processing & EDA

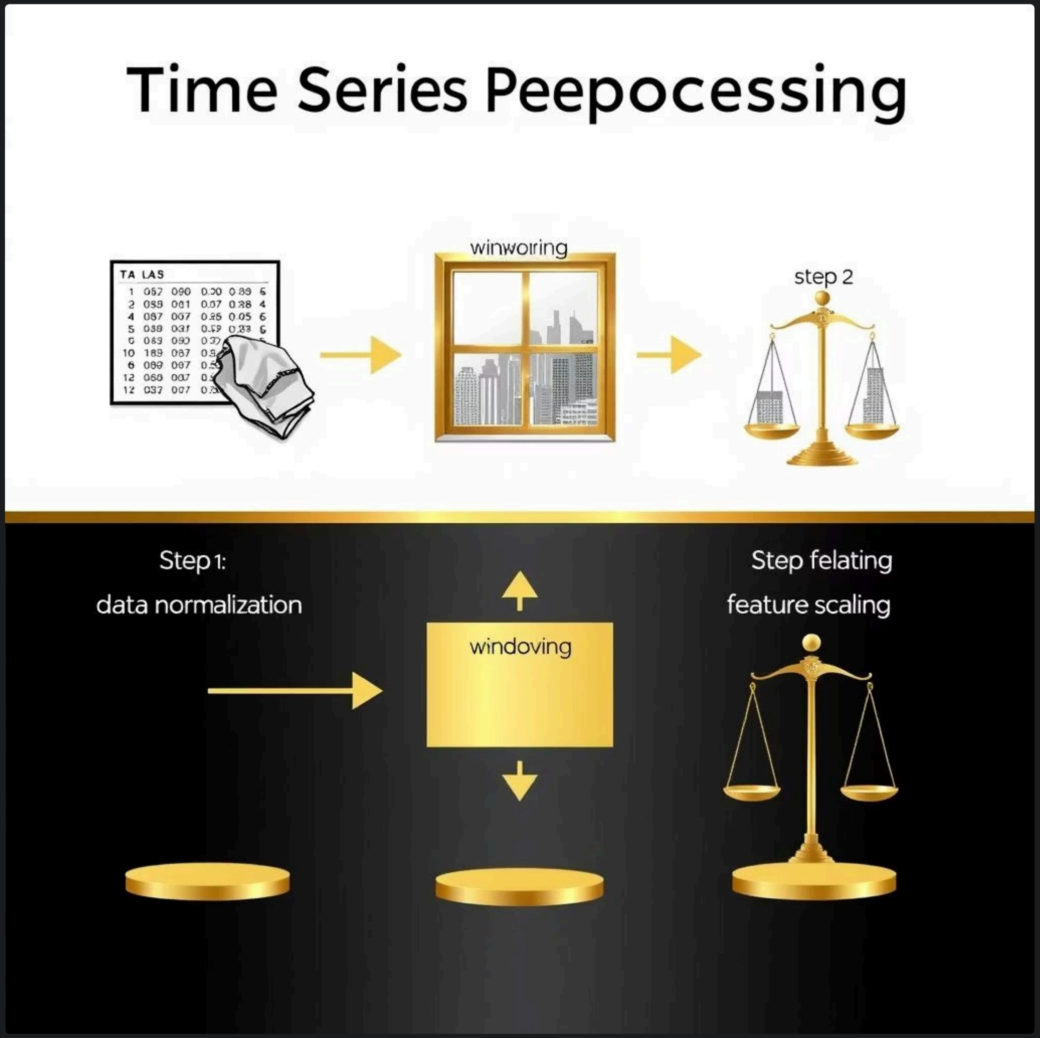
Exploratory Data Analysis (EDA)

Initial EDA revealed significant trends, cyclical patterns, and periods of high volatility in stock prices. Volume often correlated with price movements, providing insights into market activity. Moving averages were utilized to smooth out short-term fluctuations and highlight longer-term trends.

- Line plot of historical stock prices
- Moving average smoothing analysis
- Volume vs. price correlation insights

Data Preprocessing Pipeline

Missing values were handled using **forward fill**. Features were normalized via **Min-Max Scaling** to bound values between 0 and 1. A **sliding window technique** was applied to create time series sequences for model input, transforming the data into suitable formats for deep learning.

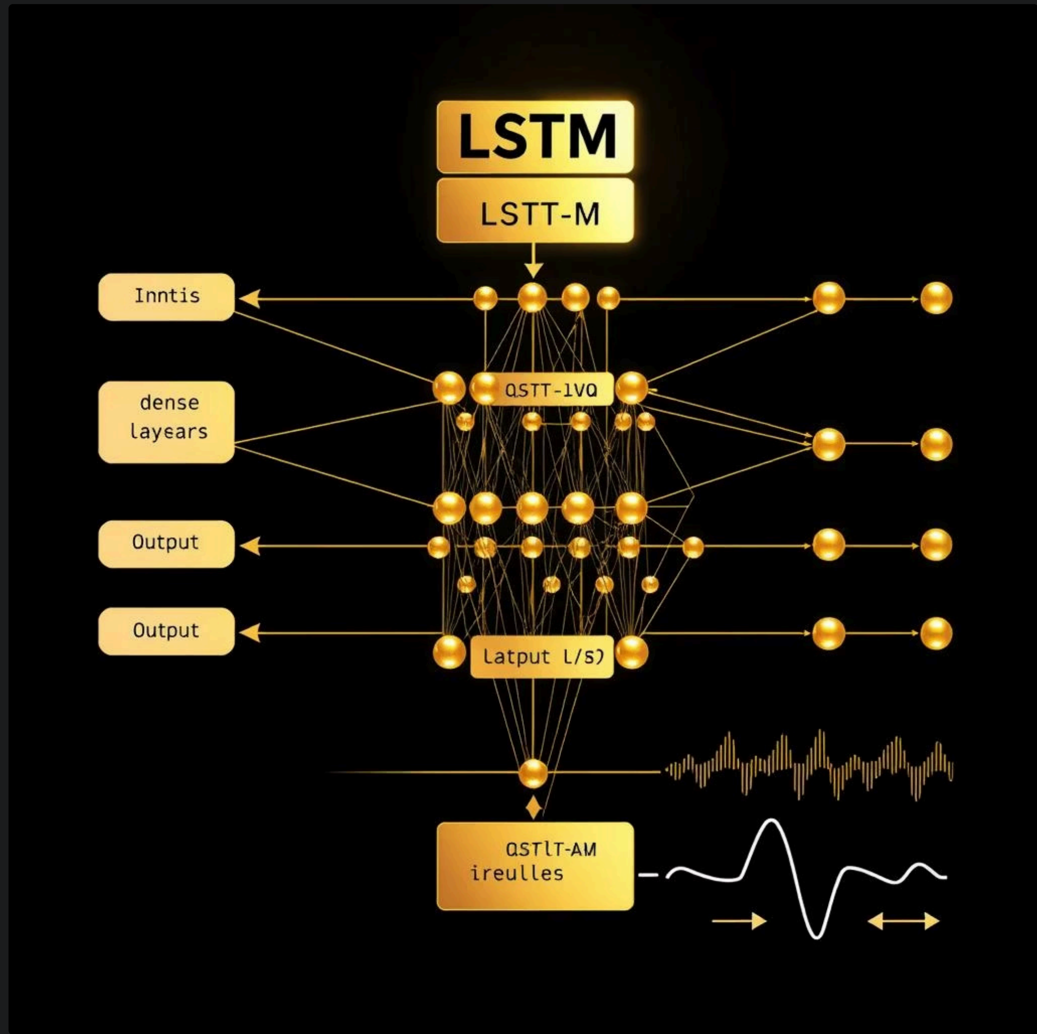


Model Design & Training

Model Architecture: Long Short-Term Memory (LSTM)

We selected an **LSTM** network for its proven capability in handling sequential data and capturing long-term dependencies. The architecture consists of:

- **Input Layer:** Sequential data of historical prices.
- **LSTM Layers:** Two stacked LSTM layers with 64 units each, utilizing a **tanh activation function**.
- **Dropout Regularization:** 0.2 dropout rate applied to prevent overfitting.
- **Dense Layer:** A fully connected layer with 32 units and ReLU activation.
- **Output Layer:** A single unit Dense layer with linear activation for price prediction.



Training Configuration & Hyperparameter Tuning

Training Configuration

- **Loss Function:** Mean Squared Error (MSE), optimized for regression tasks.
- **Optimizer:** Adam, selected for its adaptive learning rate capabilities.
- **Learning Rate:** Initialized at 0.001.
- **Batch Size:** 32 samples per update.
- **Epochs:** 100, with EarlyStopping callback.
- **Callbacks:** EarlyStopping (patience=10) and ModelCheckpoint for best model saving.

Hyperparameter Tuning

A **manual tuning strategy** was employed, focusing on key parameters critical to LSTM performance.

- **LSTM Units:** Explored 32, 64, 128 to balance complexity and performance.
- **Dropout Rate:** Tested 0.1, 0.2, 0.3 to mitigate overfitting.
- **Window Size:** Varied from 10 to 30 days to capture optimal temporal dependencies.
- **Best Result:** 64 LSTM units, 0.2 dropout, 20-day window yielded the lowest validation MSE.

Evaluation & Conclusion

0.002

MSE

Mean Squared Error on test set, indicating average squared difference between actual and predicted values.

0.045

RMSE

Root Mean Squared Error, providing error in the same units as the target variable.

0.032

MAE

Mean Absolute Error, representing the average absolute difference between predictions and actual values.

0.92

R^2 Score

Coefficient of Determination, indicating the proportion of variance in the dependent variable predictable from the independent variables.

The model demonstrated strong predictive capabilities, achieving a high R^2 score, signifying its ability to capture stock price movements. While challenges remain due to market volatility and non-stationarity, the LSTM model proved effective in short-term forecasting. Future work includes incorporating sentiment analysis and exploring Transformer models for enhanced accuracy and real-time deployment.