**APPLE STOCK PRICE PREDICTION**

# **Introduction:**

The Apple stock price prediction project employs deep learning techniques, particularly Simple RNN and LSTM models, to analyze historical stock data and forecast future prices. These models are effective in capturing temporal dependencies, making them suitable for time-series forecasting tasks. The project focuses on leveraging Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, to predict the stock prices of Apple. By utilizing historical stock data, the model aims to identify patterns and trends that can inform future price movements.

# **Data Preprocessing:**

In the preprocessing phase of the Apple stock price prediction project, the dataset is first loaded using pandas, allowing for an exploration of its key features. The 'Adj Close' price is selected as the target variable for model training, and the 'Date' column is converted to a datetime format and set as the index for better time-series analysis. To enhance model convergence, the stock prices is scaled using MinMaxScaler, normalizing the values between 0 and 1. Finally, the data is prepared for the LSTM model by creating input-output sequences, utilizing a window of past (1,5 and 10 days) days to predict the subsequent stock price.

# **Model Development:**

The SimpleRNN and LSTM architectures are built using the Sequential API from TensorFlow Keras. Each model includes a SimpleRNN or LSTM layer designed to capture sequential dependencies in the stock price data. To enhance generalization and prevent overfitting, a Dropout layer is added. Finally, a Dense layer is used to output the predicted stock price, completing the model for time-series forecasting.

# **Compiling and Model Training:**

To compile a Simple RNN and LSTM model, the mean\_squared\_error is set as the loss function, and the Adam optimizer is utilized for efficient training. Incorporating early stopping helps prevent overfitting by monitoring validation loss, while ModelCheckpoint saves the best model during training. Additionally, the trained model is saved using the **pickle** library, allowing for easy retrieval and deployment for future predictions.

# **Model Evaluation And Prediction:**

In the model evaluation and prediction phase, the trained SimpleRNN and LSTM model is utilized to predict stock prices on the test set. The actual versus predicted stock prices are compared through visualizations using Matplotlib, providing a clear graphical representation of the model's performance. To quantitatively assess the model's accuracy, the Mean Squared Error (MSE) is calculated.

# **Conclusion:**

The SimpleRNN and LSTM models effectively capture stock price trends, but they are sensitive to market fluctuations. To improve predictions, future work could incorporate additional features like news sentiment, trading volume trends, and macroeconomic indicators for a more comprehensive analysis.