**Name : Mansi Mahabdi**

**PRN: 22420190**

**Roll No : 382073**

**Batch: B3**

**Assignment No 4**

**Time series prediction using RNN – stock market analysis or weather forecasting**

**Problem Statement:**

Implement a Recurrent Neural Network (RNN) to perform time series forecasting on Tesla stock market data to predict future stock prices

**Objective:**

* To understand the working of Recurrent Neural Networks (RNN) for sequential data.
* To preprocess and prepare stock market time series data for training.
* To implement an RNN-based model for prediction.
* To evaluate prediction performance using error metrics and visualization.
* To apply the model on Tesla stock dataset from Kaggle.

**Technical Apparatus used:**

* **Operating System:** Windows/Linux/MacOS
* **Kernel:** Python 3.x
* **Tools:** Jupyter Notebook, Anaconda, or Google Colab
* **Hardware:** CPU with minimum 4GB RAM; optional GPU for faster processing

**Libraries and Packages used:**

* TensorFlow / Keras
* NumPy
* Pandas
* Matplotlib
* scikit-learn

**Dataset:**

The dataset used is Tesla Stock Data from Kaggle:  
https://www.kaggle.com/datasets/varpit94/tesla-stock-data-updated-till-28jun2021  
  
The dataset contains historical stock data of Tesla including Open, High, Low, Close prices, and Volume.

**Theory:**

Recurrent Neural Networks (RNNs) are specialized neural networks designed to work with sequential data. They have a memory component that allows information to persist across time steps, making them suitable for time series forecasting. Stock market prediction is a common use case, where past stock prices are used to predict future prices. In this assignment, Tesla stock prices are used to train and evaluate an RNN model.

**Methodology:**

**Step 1: Data Acquisition**Download Tesla stock dataset from Kaggle.  
  
**Step 2: Data Preprocessing**  
- Load dataset using Pandas.  
- Normalize stock prices using MinMaxScaler.  
- Create sequences of past n-days to predict the next day.  
  
**Step 3: Splitting Data**  
- Split into training and testing sets.  
  
**Step 4: Model Building**  
- Define an RNN model with LSTM/GRU layers.  
- Use Dense output layer with linear activation to predict stock price.  
  
**Step 5: Model Compilation**  
- Compile with Adam optimizer and Mean Squared Error (MSE) as loss function.  
  
**Step 6: Model Training**- Train on training dataset for multiple epochs.  
  
**Step 7: Evaluation**  
- Evaluate predictions using RMSE or MAE.  
- Plot predicted vs actual stock prices.  
  
**Step 8: Forecasting**- Use the trained model to forecast future stock prices.

**Advantages:**

* Captures sequential dependencies in time series data.
* Suitable for stock market and weather forecasting.
* Provides better performance than traditional regression methods.

**Limitations:**

* Requires large datasets for accurate prediction.
* Computationally expensive compared to traditional models.
* Stock market influenced by external unpredictable factors (news, politics).

**Applications:**

* Stock Market forecasting
* Weather prediction
* Sales and demand forecasting
* Energy consumption forecasting

**Conclusion:**

In this assignment, RNN was applied to Tesla stock data for time series forecasting. The preprocessing and sequence modeling allowed the RNN to capture trends in stock price movements. While predictions aligned closely with actual data, stock market forecasting remains challenging due to external influences. Nonetheless, this demonstrates the potential of RNNs in analyzing sequential financial data.