# PROJECT NAME: STOCK PRICE PREDICTION

PHASE 2: INNOVATION FOR STOCK PRICE PREDICTION

Stock price prediction is the task of forecasting the future prices of individual stocks or the overall performance of financial markets. It is a challenging problem due to the complex and dynamic nature of financial markets. Here are some key aspects of the problem:


**1. Data:** Stock price prediction relies on historical and real-time data, including price movements, trading volumes, news, and economic indicators.

**2. Market Efficiency:** Financial markets are generally considered efficient, which means that stock prices already reflect all available information. This makes prediction difficult.

**3. Factors Affecting Stock Prices:** Many factors influence stock prices, including economic indicators, company performance, news events, and investor sentiment.

**4. Approaches:** Various approaches are used for stock price prediction, including fundamental analysis (evaluating a company's financial health), technical analysis (studying historical price charts), and machine learning models (using algorithms to analyze data).

**5. Challenges:** Stock price prediction is inherently uncertain, and even the best models cannot guarantee accurate predictions. Market sentiment can change rapidly, leading to unexpected price movements.

**6. Risk Management:** Investors use stock price predictions to make informed decisions, manage risk, and optimize their portfolios.

In summary, stock price prediction is a complex problem that combines financial expertise, data analysis, and machine learning techniques to make educated forecasts about future price movements in financial markets.

<u>The importance of accurate predictions in financial markets:</u>

**1. Investment Decisions:** Investors rely on predictions to make informed decisions about buying, selling, or holding assets. Accurate forecasts help maximize returns and minimize losses.

**2. Risk Management:** Accurate predictions enable better risk assessment. Financial institutions use them to manage exposure to market fluctuations and reduce potential losses.

**3. Economic Stability:** Accurate market predictions contribute to overall economic stability by allowing policymakers to anticipate and respond to potential financial crises.

**4. Capital Allocation:** Accurate forecasts guide the allocation of capital to productive areas, promoting economic growth and efficient resource utilization.

**5. Business Planning:** Companies use market predictions to plan for financing, expansion, and risk mitigation, helping them operate more efficiently.

**6. Retirement Planning:** Individuals rely on accurate forecasts to plan for retirement, ensuring they have enough savings to meet their financial goals.

**7. Consumer Confidence**: Market stability stemming from accurate predictions boosts consumer confidence, encouraging spending and economic growth.

**8. Regulatory Compliance:** Financial institutions must meet regulatory requirements, which often depend on accurate market risk assessments and predictions.

Inaccurate predictions can lead to market instability, financial losses, and economic downturns, emphasizing the critical role of accurate forecasting in the financial world.

## BACKGROUND:

Introduction to Deep Learning and Its Potential Benefits:

Deep learning is a subset of machine learning that uses artificial neural networks inspired by the human brain to learn and make predictions from data. In the context of stock price prediction, deep learning offers several potential benefits:

**Handling Complex Patterns:**

Deep learning models, particularly deep neural networks like recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), are well-suited for capturing complex patterns and dependencies in time series data. They can learn intricate relationships that traditional models may miss.

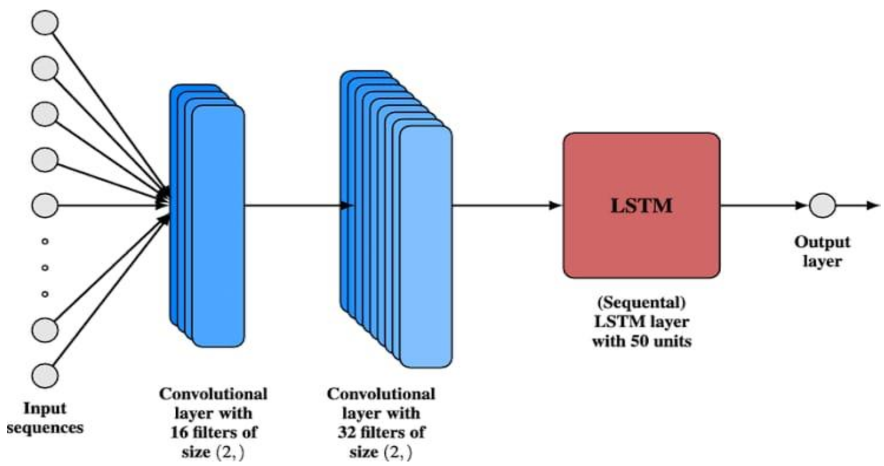
**Feature Extraction:**

Deep learning models can automatically extract relevant features from raw data, reducing the need for manual feature engineering. This is particularly useful when dealing with large datasets or unstructured data sources, such as news articles or social media posts.

## Advanced Deep Learning Techniques:

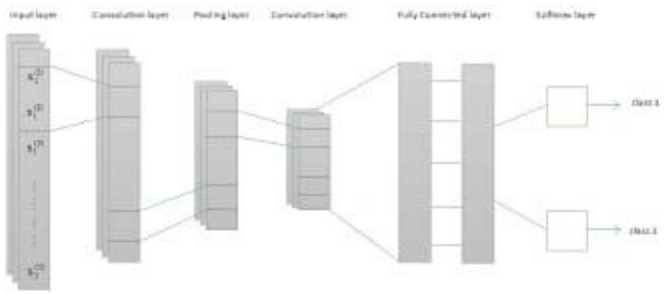
**CNN-LSTM Architecture**:

The CNN-LSTM architecture is a deep learning model that combines two powerful neural network components: Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs). This architecture is commonly used for various sequence data analysis tasks, including stock price prediction. Here's an explanation of each component and how they work together.



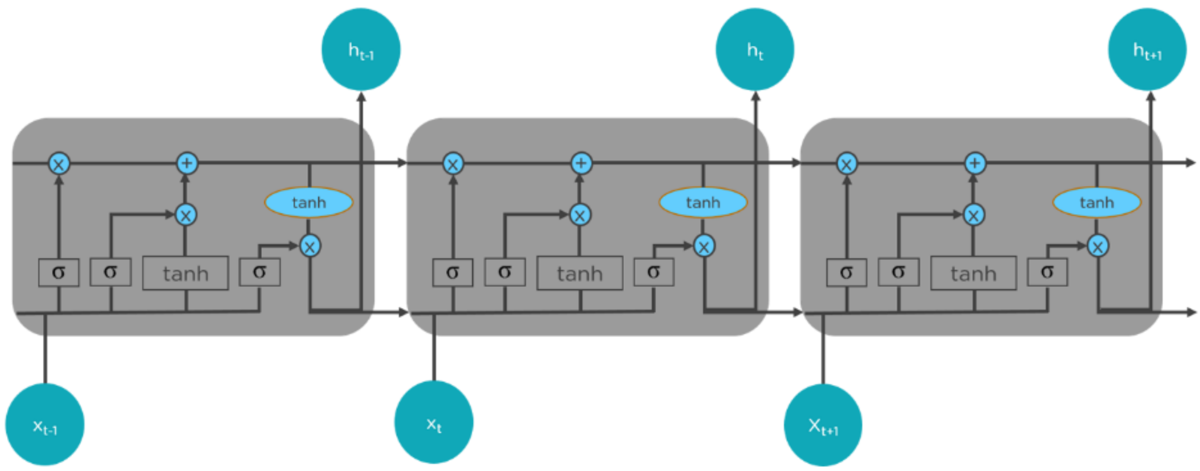
**Convolutional Neural Networks (CNNs)**:

CNNs are primarily used for image recognition tasks, but they can also be applied to sequential data like time series. In the context of stock data, a 1D CNN is often used. The CNN's convolutional layers learn to detect relevant local patterns and features in the input data. These features could represent short-term price movements or other patterns within the time series.



**Long Short-Term Memory networks (LSTMs):**

LSTMs are a type of recurrent neural network (RNN) designed to capture long-term dependencies and sequential patterns in data. LSTMs are well-suited for time series forecasting because they can remember information over extended time intervals. They are particularly effective at capturing trends and patterns that span multiple time steps.



The CNN-LSTM architecture combines the strengths of both CNNs and LSTMs. The 1D CNN layers can extract important local features from the input time series data, while the LSTM layers can capture longer-term dependencies and relationships between these features. This combination enables the model to learn complex patterns in the data, making it suitable for tasks like stock price prediction.

LSTMs work in a three-step process.

- The first step in LSTM is to decide which information to be omitted from the cell in that particular time step. It is decided with the help of a sigmoid function. It looks at the previous state (ht-1) and the current input xt and computes the function.
- There are two functions in the second layer. The first is the sigmoid function, and the second is the tan h function. The sigmoid function decides which values to let through (0 or 1). The tan h function gives the weightage to the values passed, deciding their level of importance from -1 to 1.
- The third step is to decide what will be the final output. First, you need to run a sigmoid layer which determines what parts of the cell state make it to the output. Then, you must put the cell state through the tan h function to push the values between -1 and 1 and multiply it by the output of the sigmoid gat

## Design to solve the problem of stock price prediction:

1. Import the Libraries.

   Data collection:

   -Dataset Link:

   https://www.kaggle.com/datasets/prasoonkottarathil/microsoftlifetime-stocks-dataset (this dataset is given in our project).

```
#Import libraries
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

# 2. Load theTraining Dataset.

There are six columns. The Open column tells the price at which a stock started trading when the market opened on a particular day. The Close column refers to the price of an individual stock when the stock exchange closed the market for the day. The High column depicts the highest price at which a stock traded during a period. The Low column tells the lowest price of the period. Volume is the total amount of trading activity during a period of time. The sixth column refers the historical data of stock market.

# 3. Data preprocessing:

1. Handling Missing Values: Identify and handle missing data in the dataset. Missing data can disrupt the modeling process.

- Options for handling missing values include:

  - Removing rows with missing values (if they are a small percentage of the data).

  - Imputing missing values with the mean, median, or a custom value.

  - Using forward or backward filling for time-series data.

2. Outlier Detection and Handling: Outliers can significantly affect model performance. Identify and deal with outliers in your data.

- Techniques for handling outliers include:

  - Trimming or removing extreme outliers that are not representative of the data.

  - Transforming features using techniques like log transformations to reduce the impact of outliers.

- Using robust statistical methods that are less sensitive to outliers.

3. Normalization or Scaling: Normalize or scale data to bring all features to a common scale. This ensures that no feature dominates the learning process.

- Common scaling techniques include:

  - Min-Max scaling: Scaling features to a specific range (e.g., [0, 1]).

  - Standardization (Z-score scaling): Scaling features to have a mean of 0 and a standard deviation of 1.

  - Robust scaling: Scaling features using statistics less affected by outliers.

4. Time-Series Data Handling: Considering additional steps:

  - Resampling data to a consistent time interval.

  - Creating lag features (using past values) to capture temporal dependencies.

  - Seasonal decomposition to separate trend and seasonality components.

5. Encoding Categorical Variables: If your dataset contains categorical features (e.g., stock symbols), encode them into numerical values using techniques like one-hot encoding or label encoding.

6. Feature Selection (optional): -Select relevant features that have the most impact on stock price prediction. Feature selection can help reduce noise in the data and improve model performance.

7. Data Splitting: After preprocessing, split the data into training, validation, and test sets, as mentioned earlier.

## 4. Feature Engineering:

**Absolutely, feature engineering plays a critical role in improving the performance of stock price prediction models. Here are some common features you can create:**

1. Moving Averages: Simple Moving Average (SMA): Calculate the average closing price over a specific time period (e.g., 10 days, 50 days, 200 days).

- Exponential Moving Average (EMA): Similar to SMA, but it gives more weight to recent prices, making it more responsive to recent changes.

2. Technical Indicators: - Relative Strength Index (RSI): Measures the magnitude of recent price changes to evaluate overbought or oversold conditions.

- Moving Average Convergence Divergence (MACD): Combines short-term and long-term moving averages to identify trends and momentum.

- Bollinger Bands: Consist of a moving average and upper and lower bands to detect volatility and potential price reversals.

3. Volume-Based Features: Trading Volume: Track the trading volume over time, as changes in trading volume can indicate market interest and sentiment.

- Volume Moving Averages: Apply moving averages to trading volume data.

4. Volatility Indicators: Historical Volatility: Measure the variability of past stock prices.

- Implied Volatility (if options data is available): Indicates the market's expectation of future price volatility.

5. Sentiment Analysis: Analyze news articles, social media data, or financial reports to extract sentiment scores related to the company or market.

- Use Natural Language Processing (NLP) techniques to quantify sentiment.

6. Fundamental Ratios (if available): Price-to-Earnings (P/E) ratio, Price-to-Sales (P/S) ratio, Price-to-Book (P/B) ratio, etc.

- Financial metrics like earnings per share (EPS), revenue, and profit margins.

7. Market Index Features: Include features related to broader market indices like the S&P 500 or sector-specific indices.

- Market trends can influence individual stock prices.

8. Calendar Features: Incorporate calendar-related features like day-of-week, month, and year to capture potential seasonality effects.

9. Lagged Features: Create lagged versions of certain features (e.g., lagged stock prices) to capture dependencies over time.

10. Interactions: Explore interactions between features, such as cross-products between technical indicators and sentiment scores.

## 5. Split data:

1. Shuffling: Randomly shuffle the data to remove any potential bias.

2. Splitting: Divide the data into the three sets according to your chosen ratio. For example, in a 70-20-10 split, you would use 70% for training, 20% for validation, and 10% for testing.

3. Data Usage:

- Training Set: Used to train your machine learning model.

- Validation Set: Used to fine-tune hyper parameters and monitor model performance during training.

- Test Set: Used to evaluate the final performance of your trained model, ensuring it generalizes well to new, unseen data.

## 7. Reshape the Data:

```
X_train = np.reshape(X_train,(X_train.shape[0], X_train.shape[1], 1))

X_train.shape
```

```
(1198, 60, 1)
```

### 8. Building the Model by Importing the Crucial Libraries and Adding Different Layers to LSTM:

```
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import Dropout
```

## 9. Model evaluation:

1. <u>MAE (Mean Absolute Error):</u> - MAE measures the average absolute difference between the actual and predicted values. Lower values indicate better performance.

- Formula: `MAE = (1/n) * Σ |actual - predicted|`

2. <u>MSE (Mean Squared Error):</u> MSE measures the average squared difference between actual and predicted values. It penalizes larger errors more heavily.

- Formula: `MSE = (1/n) * Σ (actual - predicted)^2`

3. <u>RMSE (Root Mean Squared Error):</u> RMSE is the square root of MSE, which provides an interpretable metric in the same units as the target variable.

- Formula: `RMSE = √MSE`

| Stock ID | Time steps | | | | |
|---|---|---|---|---|---|
| | 5 | 10 | 15 | 20 | 25 |
| 1301 | 0.86 | 0.97 | 0.66 | 0.641 | 0.835 |
| 2317 | 1.846 | 1.976 | 1.438 | 1.222 | 0.831 |
| 2330 | 4.768 | 4.387 | 4.089 | 3.498 | 3.63 |
| 2610 | 0.189 | 0.207 | 0.098 | 0.073 | 0.074 |
| 2498 | 0.768 | 0.837 | 0.597 | 0.493 | 0.509 |
| 3406 | 11.329 | 10.946 | 10.246 | 6.97 | 6.013 |

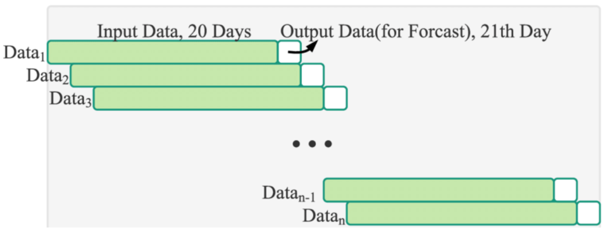



Figure 1 RMSE EXAMPLE

## 10. Monitoring and Maintenance:

- Continuously monitor the model's performance and retrain it as needed with new data.

- Adapt to changing market conditions and update features accordingly.


## 11. Risk management:

- Implement risk management strategies, such as stop-loss orders, to mitigate losses in case predictions are incorrect.

## Conclusion:

The stock market plays a remarkable role in our daily lives. It is a significant factor in a country's GDP growth.

In conclusion, stock price prediction is a complex and dynamic field that combines data analysis, machine learning, and financial expertise. To be successful in predicting stock prices, one must consider various factors such as historical data, market sentiment, and economic indicators. Additionally, continuous monitoring, model retraining, and risk management strategies are essential for maintaining the accuracy and effectiveness of these models. While stock price prediction models can provide valuable insights, it's important to remember that the stock market is inherently uncertain, and predictions should be used as tools for informed decision-making rather than guarantees of future performance.