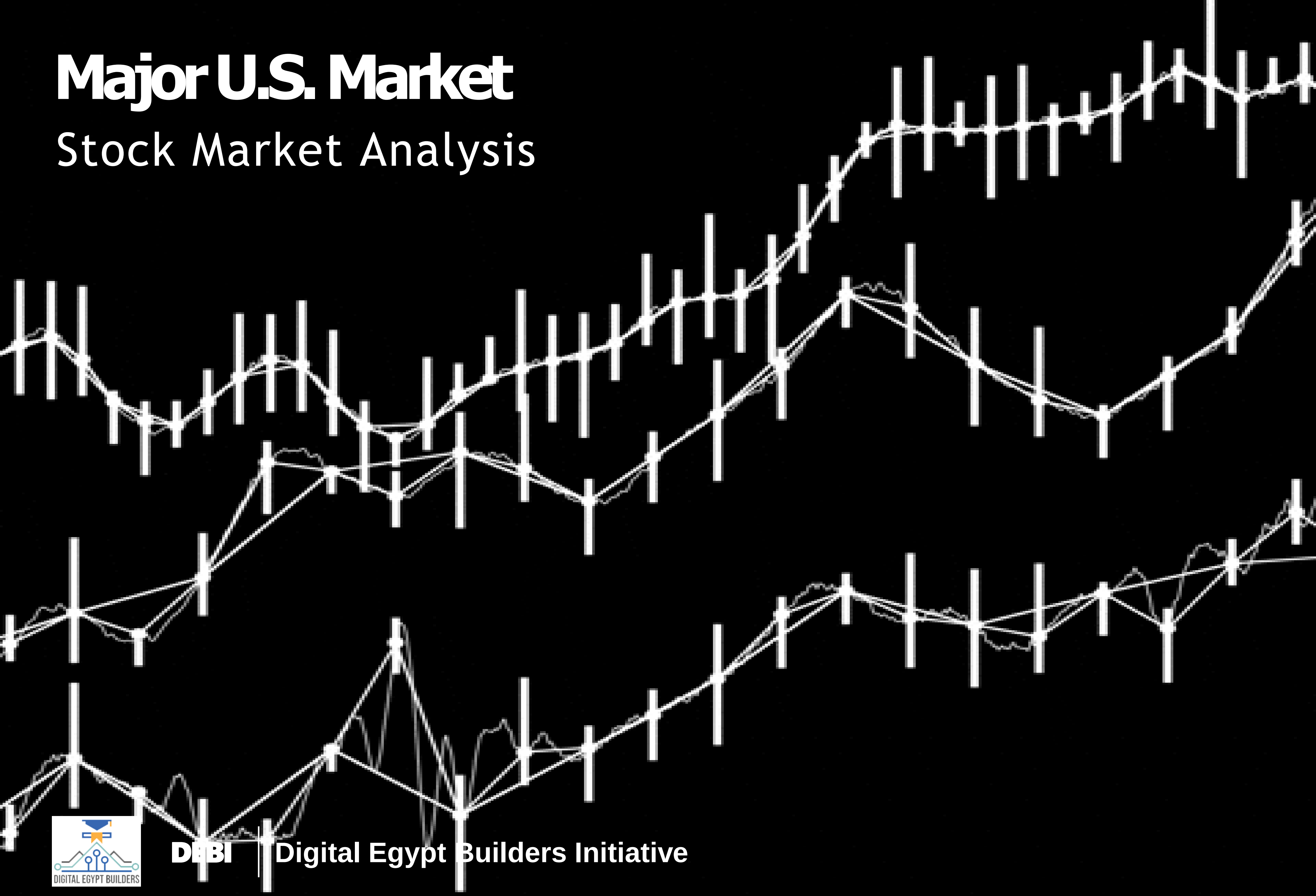
# Major U.S. Market

## Stock Market Analysis



DEBI | Digital Egypt Builders Initiative

.35%
1.91%
7.48%
.82%
86
103
45

# TABLE OF CONTENTS

# FOREWORD

Revised Content: "In today's data-driven world, the ability to predict stock prices is an ongoing challenge in financial markets. This project employs a combination of fundamental and technical analysis to enhance prediction accuracy. By integrating machine learning algorithms such as Decision Trees, Random Forests, and Long Short-Term Memory (LSTM) models, we explore a multifaceted approach to predicting stock price movements. Our goal is to demonstrate the transformative impact of modern data analytics and AI on stock forecasting, making market movements more transparent and understandable to investors. We've focused on the S&P 500 companies, gathering data from the Yfinance API, and processed it using Python to ensure accuracy and relevance.

Each stage plays a critical role in ensuring that the predictions are reliable, accurate, and actionable. Data Collection and Cleaning Using Python The foundation of any data-driven project is the quality of the data itself. We began by collecting historical financial data from the S&P 500 companies using Python's powerful Yfinance library. This allowed us to extract key financial metrics such as stock prices, trading volumes, and market capitalization.

The significance of using Yfinance lies in its ability to provide real-time and historical financial data, which forms the basis for our predictions. However, collecting raw data is just the beginning. The next essential step is data cleaning. Raw financial data often contains inconsistencies like missing values, zero entries, or other noise that can distort the accuracy of machine learning models. Using custom Python scripts, we cleaned the data to ensure that only meaningful and valid data points were used.

Our Python cleaning code played a pivotal role here. The script ensured that data preprocessing was automated, allowing us to handle large datasets efficiently. For example, the code identified any null or zero values and excluded them from the analysis.

This process also involved transforming the data into a structured format, ready for further analysis and visualization. Managing the Workflow Using Trello and the Agile Manifesto Managing a project of this scope requires careful planning and constant iteration. This is where Trello came into play, acting as the central hub for project management. Following the principles of the Agile Manifesto, we broke down the project into manageable "sprints," with each sprint focusing on a specific task—whether it was data collection, cleaning, or model training.

**Karim Hisham**

**Makarius Nader**

**Syed Mohamed**

## Problem Definition:

In this project, we aim to investigate the relationship between a company's fundamental financial health and its stock price performance. The financial health of a firm is typically assessed using fundamental analysis, which includes key metrics like Economic Value Added (EVA), EBIT, ROA, and working capital efficiency. These indicators are commonly used by investors to evaluate whether a company is a strong or weak investment candidate.

However, despite the use of such metrics, there is a growing belief that the stock market often moves independently of a firm’s actual financial health, and stock prices may not reflect the true value of a company. This could be due to market sentiment, external economic factors, speculation, or other unpredictable influences.

To investigate this hypothesis, we propose combining fundamental analysis in Power BI with a predictive stock price model using prediction models in Python. By analyzing the fundamental financial metrics of companies and comparing them with the predicted stock prices, we aim to demonstrate that there is no strong correlation between the fundamental health of a firm and its stock price. We hypothesize that whether a company is fundamentally strong or weak, it is not a clear indicator for investment decisions based on stock price performance.

## Methodology:

- **Fundamental Analysis in Power BI:**

  Perform detailed fundamental analysis of several firms using key financial metrics such as EVA, EBIT, ROA, and working capital ratios. This will provide insights into the financial strength or weakness of each company.

- **Predictive Modeling using Python:**

  Build a stock price prediction model using historical stock prices. The model will predict future stock prices based on past data trends.

- **Comparison of Fundamental Health and Stock Price Predictions:**

  Analyze the correlation (or lack thereof) between the predicted stock prices and the fundamental health metrics of the companies. The outcome of this analysis will help determine whether a fundamentally strong company necessarily leads to stock price growth or vice versa.

## Expected Outcome:

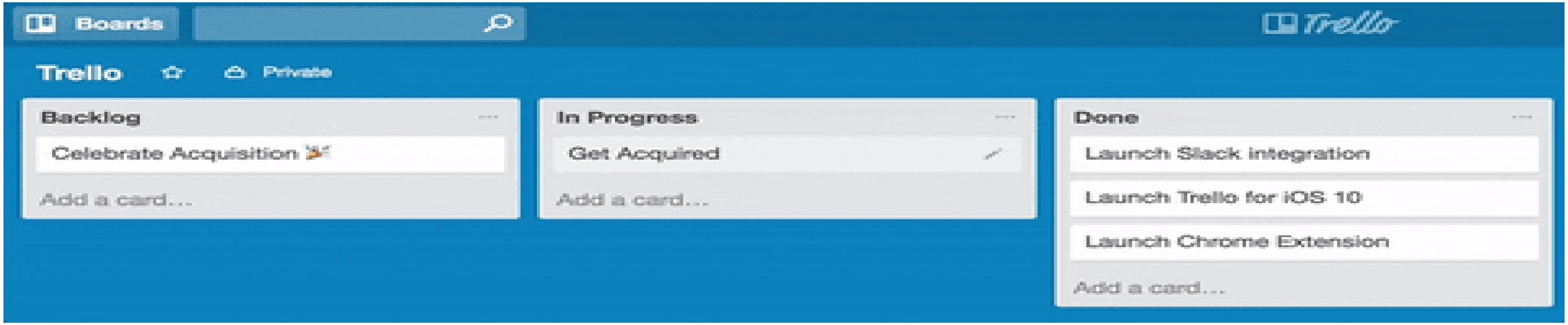
We expect to find that the fundamental financial health of a company is not a reliable indicator of its stock price performance. This will suggest that investors should consider other factors (such as market sentiment, macroeconomic conditions, and technical analysis) alongside fundamental analysis when making investment decisions.

## Key Elements to Include:

- Introduction to the project management tool we used to manage this project and its impact on our results.
- Introduction to Fundamental Analysis and why it’s typically used for investment decisions.
- Introduction to the prediction model and its use in time-series prediction, specifically for stock price forecasting.
- Hypothesis: There is no significant correlation between fundamental health metrics and stock prices.
- Objective: To prove that stock prices can move independently of a firm’s fundamental performance.
- Tools Used: Power BI for the analysis and Python for predictive modeling and Trello for project management.

SECTION 1

# Project Management with Trello



Boards

Trello

Trello ☆ Private

Backlog

Celebrate Acquisition

Add a card…

In Progress

Get Acquired

Add a card…

Done

Launch Slack integration

Launch Trello for iOS 10

Launch Chrome Extension

Add a card…

# Project Management with Trello for Graduation Project

### Overview:

The Trello board you see here is an integral part of our project management for the Graduation project, providing a organized, visual workflow to track tasks and progress across various stages. It is divided into multiple lists, each representing a phase or aspect of the project, from Backlog & Resources to Done, including Design, Learning, Code Review, Testing, and Presentation preparation.

### How Trello Helps Our Workflow:

- Visual Clarity: Each stage of the project is visually distinct, making it easy to understand the current state of tasks immediately.
- Collaboration: all team members can collaborate by adding comments, attaching files, and updating task statuses in real-time.
- Tracking Progress: Moving tasks through the board's different lists helps track progress, ensuring that all deliverables are met within the deadlines.

### Key Sections:

- **Backlog & Resources**: Stores tasks and resources for future use, like Python projects and DAX issues.
- **Design**: Focuses on the visual design and research aspects, including integrating slides and reports.
- **To Learn**: Tracks learning objectives, such as Power BI and DAX, ensuring skill development during the project.
- **Weekly Meetings**: Organizes meeting notes and action items from weekly check-ins with Dr. Mohamed Ibrahim.
- **Code Review**: Ensures quality control by reviewing critical project components, like the Stock Price Prediction Model.
- **Testing**: Validates code and project functionality with tests to ensure accurate financial predictions.
- **Done**: Completed tasks, such as Financial Ratios Calculation and Presentation Slide Mapping, are moved here.
- **Presentation**: Holds final presentation materials, including QR codes and links to project documentation.

**This Trello board has proven essential in organizing and managing our complex graduation project, promoting teamwork, and ensuring a streamlined process from initial planning to final presentation.**

# SECTION 2

## Python Synergy Dataset
## Extracting and cleaning.

# Financial Data Extraction and Analysis Using Python & yFinance

## Overview:

In this project, we utilize Python's yfinance library to extract financial data for S&P 500 companies and structure it into a relational database format for analysis. The goal is to create a comprehensive dataset that combines financial statements, market data, and industry rankings, which will then be used for financial modeling and analysis.

## Key Components of the Code and Process:

### Fetching S&P 500 Company Data:

The code begins by fetching a list of S&P 500 companies, including their symbols, industry sectors, and headquarters location, from a Wikipedia page. This forms the foundation for our data pipeline.

### Financial Data Collection:

- For each company, we use the yfinance library to pull detailed financial information such as balance sheets, income statements, and cash flow statements.
- Balance Sheet: Includes key metrics like total assets, liabilities, and stockholder equity.
- Income Statement: Extracts items such as net income, revenue, and operating expenses.
- Cash Flow: Focuses on cash from operating, investing, and financing activities.

### Company Information & Additional Metrics:

In addition to financials, the script fetches supplementary data like the year of establishment, country, and market capitalization, which is used to rank companies within the S&P 500 and within their industry.

### Ranking by Market Cap:

The market capitalization of each company is used to rank it both in the S&P 500 index and within its specific industry. This creates two key metrics:

- Rank in S&P 500: Position of the company in the overall index.
- Rank in Industry: Position of the company relative to other companies in the same industry.

### Storing Data in Excel:

The collected data is saved into an Excel file, with separate sheets for:

- Financial Data: Contains financial metrics (e.g., revenue, equity) with their respective dates and categories (balance sheet, income statement, etc.).
- Index Data: Stores general information about each company (e.g., name, stock code, industry).
- Industry Summary: Summarizes industries, with the number of companies in each and regional information.

## Key Points:

- This automated process of fetching, analyzing, and organizing financial data for S&P 500 companies gives us a robust dataset for financial modeling. It eliminates manual data collection, ensuring data accuracy and efficiency. By incorporating rankings and financial ratios, the model allows for comparative analysis both within industries and across the index.

- This setup ensures that the project is scalable and can be applied to other indices or markets with minimal adjustments. The data can be further used to perform deeper analysis, visualize trends, or even predict financial performance.

Source: V6_[Historical]Consolidating_Financial_Data_for_YFinance_Stocks.ipynb

# Data Model & Structure

## Index Table:

Contains general information about each company (e.g., company name, stock code, rank in S&P 500, industry).

## FinancialData Table:

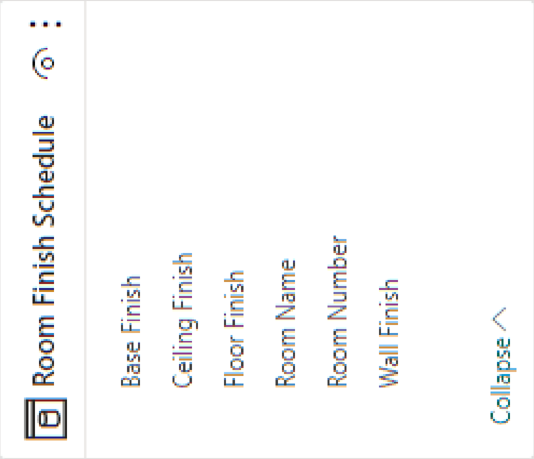
- Holds specific financial metrics for each company (e.g., revenue, total assets, net income) categorized by balance sheet, income statement, and cash flow.
- Each entry includes the amount, date, and industry.

## FinancialRatios Table:

Calculates and stores financial ratios for companies, such as the current ratio, debt-to-equity ratio, and return on assets (ROA), allowing for deeper financial analysis.

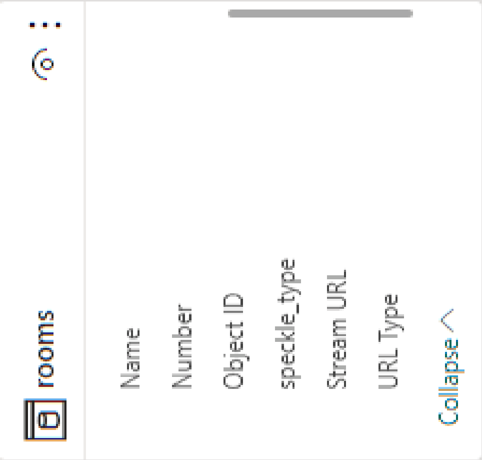
## IndustrySummary Table:

Summarizes industry-level data, showing the number of companies, countries, and regions represented in each industry.



Room Finish Schedule

Base Finish
Ceiling Finish
Floor Finish
Room Name
Room Number
Wall Finish

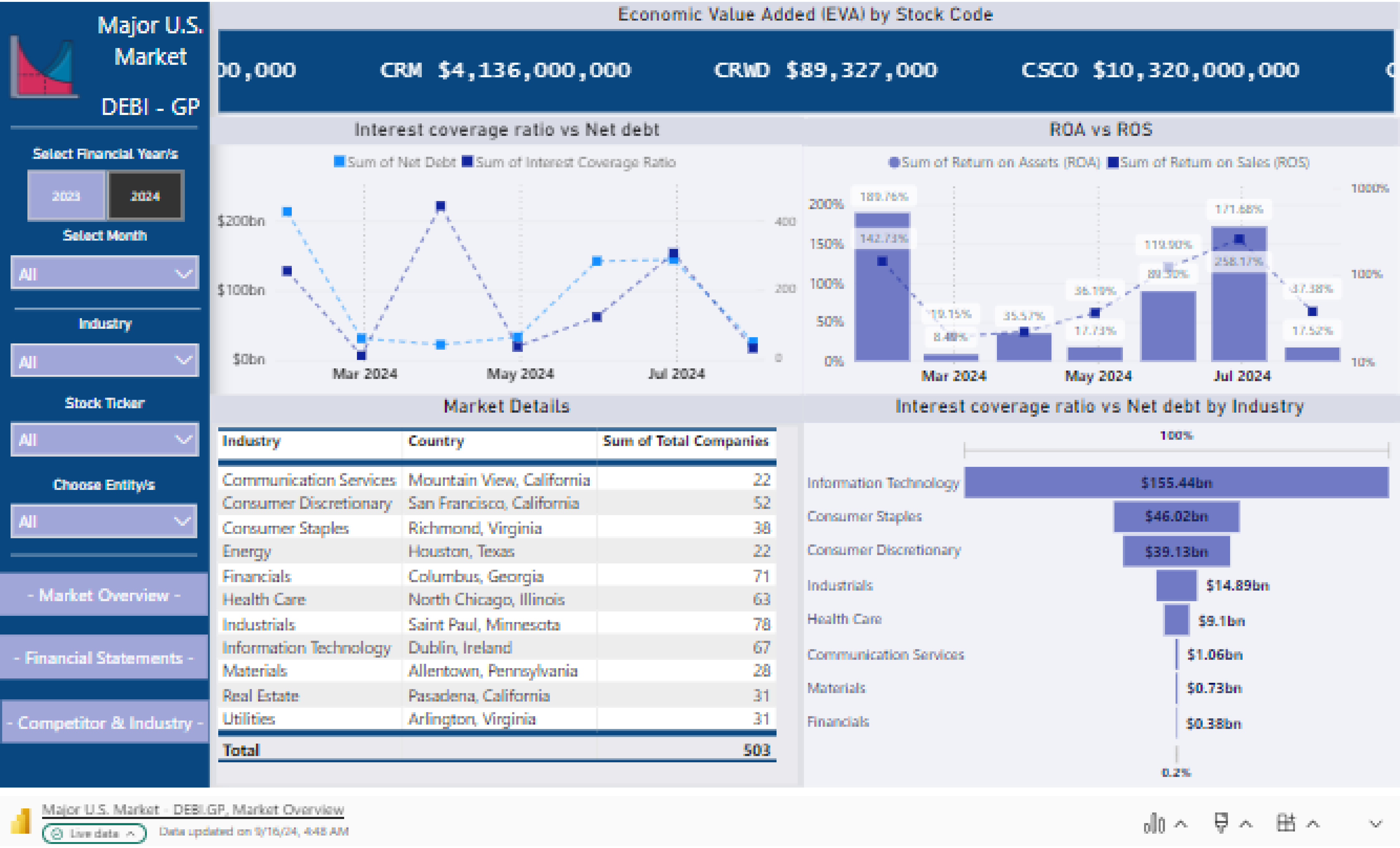


rooms

Name
Number
Object ID
speckle_type
Stream URL
URL Type

# SECTION 3

# Power BI Synergy for Fundamental Stock Analysis



Major U.S. Market

DEBI - GP

Select Financial Year/s

2023 | 2024

Select Month: All

Industry: All

Stock Ticker: All

Choose Entity/s: All

- Market Overview -

- Financial Statements -

- Competitor & Industry -

Economic Value Added (EVA) by Stock Code

00,000 CRM $4,136,000,000 CRWD $89,327,000 CSCO $10,320,000,000

Interest coverage ratio vs Net debt

ROA vs ROS

Market Details

| Industry | Country | Sum of Total Companies |
|---|---|---|
| Communication Services | Mountain View, California | 22 |
| Consumer Discretionary | San Francisco, California | 52 |
| Consumer Staples | Richmond, Virginia | 38 |
| Energy | Houston, Texas | 22 |
| Financials | Columbus, Georgia | 71 |
| Health Care | North Chicago, Illinois | 63 |
| Industrials | Saint Paul, Minnesota | 78 |
| Information Technology | Dublin, Ireland | 67 |
| Materials | Allentown, Pennsylvania | 28 |
| Real Estate | Pasadena, California | 31 |
| Utilities | Arlington, Virginia | 31 |
| **Total** | | **503** |

Interest coverage ratio vs Net debt by Industry

Major U.S. Market - DEBI.GP, Market Overview

Live data

Data updated on 9/16/24, 4:48 AM

## Report Overview: U.S. Market Sector Financial Metrics

The U.S. Market Sector Financial Metrics Dashboard provides a comprehensive view of key financial indicators across various companies and industries. At its core, the dashboard focuses on Economic Value Added (EVA), which measures the value companies create beyond their cost of capital. This metric enables users to compare the financial performance of companies like ADP and AMCR, highlighting those that generate the most value. EVA serves as a crucial indicator for identifying high-performing companies and sectors, guiding investment decisions and strategic focus.

The top section presents Economic Value Added (EVA) for various companies like ADP, AMCR, and BBWI. EVA is a critical metric for assessing financial performance, providing a snapshot of how much value each company has created beyond its cost of capital. For instance, ADP's EVA of $3.75 billion and AMCR's $730 million allow for a quick comparison of company performances. This focus on EVA at the top of the dashboard enables users to assess value creation for individual stock codes briefly.

## Interest Coverage Ratio vs. Net Debt

One of the key visuals is a bar and line chart that compares the sum of net debt (displayed in bars) against the interest coverage ratio (shown as a line) over time. This combination allows users to understand how well companies manage their debt. For example, in May 2024, a net debt of $44.3 billion corresponds to a declining interest coverage ratio, signaling potential financial risks. This chart helps users track the balance between debt burden and a company's ability to service its debt.

## ROA vs. ROS

Another important visual compares Return on Assets (ROA) and Return on Sales (ROS) over the same time. These metrics are essential for evaluating how efficiently companies are generating profit from their assets and sales. For example, a sharp increase in both ROA and ROS in July 2024 suggests improved financial performance. By displaying these metrics side-by-side, the dashboard enables users to assess operational efficiency and profitability trends.

## Market Details Table

In the middle section, the dashboard presents a Market Details table that breaks down companies by industry and location. This provides an overview of the geographic and industry distribution of companies, such as the dominance of Communication Services in Mountain View, California, and Energy companies in Houston, Texas. This table is helpful for users looking to understand industry concentrations and regional focus, giving a quick insight into market presence and distribution.

## Interest Coverage Ratio vs. Net Debt by Industry

Another bar chart displays net debt values for different industries, offering insights into which sectors carry the highest debt levels. For example, Information Technology leads with $155.44 billion in debt, followed by Consumer Staples with $46.02 billion. This chart highlights sectors that may be at greater financial risk if their interest coverage ratios decline. By presenting debt levels by industry, the dashboard helps users identify potential areas for financial monitoring and risk assessment.

## Financial Performance Over Time

The dashboard also tracks key financial metrics like gross margin, operating margin, current ratio, operating cash flow, EBIT margin, adjusted EBIT, and net assets over time. For instance, the gross margin spikes in July 2024 indicate improved profitability, while the operating cash flow rises from $134 billion in March to $176 billion in July, signaling strong operational health. Additionally, the net assets chart shows significant growth, with net assets increasing from $0.36 trillion in May 2024 to $1.01 trillion in July, reflecting company expansion or asset acquisition.

## Competitor Ranking and Working Capital Efficiency

A bar chart ranks companies by Economic Value Added (EVA), offering a competitive overview of their financial performance. For example, AAL ranks highest with an EVA of 498, followed by AGCL at 217. This ranking allows for easy comparison of how companies fare against their competitors. Additionally, a scatter plot visualizes working capital efficiency across various companies, highlighting how well they manage liquidity and operational cash flow relative to peers.

## EVA by Industry

A treemap visualization presents Economic Value Added (EVA) by industry, with Information Technology contributing $155.44 billion, followed by Consumer Staples with $46.02 billion. This visualization provides a clear perspective on which industries generate the most value, helping users identify dominant sectors and their relative contributions to overall market value.

## Debt Management and Financial Stability Risks

The Interest Coverage Ratio and Net Debt comparison highlights potential financial risks. As companies accumulate more debt without improving their interest coverage ratios, they face increased financial vulnerability. For instance, the declining interest coverage ratio in May 2024, despite high net debt, signals that companies may struggle to service their debts in the future.

The industries with the highest net debt, such as Information Technology and Consumer Staples, should be closely monitored for changes in their ability to manage debt, especially if macroeconomic conditions (e.g., rising interest rates) worsen.

## Asset Expansion and Capital Allocation

The significant growth in Net Assets from $0.36 trillion to $1.01 trillion between May and July 2024 points to substantial asset accumulation. This could be due to mergers and acquisitions, increased investments in capital projects, or improved asset management.

A closer look at whether this growth is driven by asset quality or merely higher debt accumulation would help assess whether the companies are expanding sustainably or exposing themselves to future risks.

## Operational Efficiency and Profitability Trends

The simultaneous rise in Return on Assets (ROA) and Return on Sales (ROS) in July 2024 indicates a significant improvement in operational efficiency and profitability. Companies appear to be leveraging their assets more effectively, leading to better sales performance.

This improvement could suggest that companies are capitalizing on operational efficiencies, such as better resource allocation or cost-cutting measures, leading to higher profitability.

## Sector-Specific Financial Strength

Economic Value Added (EVA), shown both for individual companies and by industry, reveals that Information Technology leads in value creation. This suggests that the sector not only manages its capital well but also generates returns far beyond its cost of capital. Industries like Consumer Staples, though trailing, also show strong EVA contributions.

Monitoring EVA over time may provide insights into which sectors maintain sustainable growth or are more resilient during economic downturns.

## Geographic and Industry Distribution Impact

The Market Details table suggests a concentration of specific industries in certain regions, such as Communication Services in Mountain View, California, and Energy in Houston, Texas. This could provide insight into regional strengths or risks, such as dependency on a single industry or exposure to local economic shocks.

Companies in these regions might be more vulnerable to local economic changes (e.g., tech sector regulation in Silicon Valley or energy price fluctuations in Texas).

## Profitability and Operational Health

The sharp rise in Gross Margin and Operating Cash Flow between March and July 2024 points to improved profitability and operational health across the market sectors. This could indicate companies are increasing their revenue or better controlling costs.

The rising Operating Cash Flow suggests companies are not only reporting profits but also converting those profits into cash, which strengthens their ability to reinvest, pay down debt, or return value to shareholders.

## Working Capital Management

The scatter plot on working capital efficiency is useful in identifying which companies are best managing liquidity and operational cash flow. Companies that excel in working capital management may be better positioned to handle short-term obligations and invest in growth without taking on excessive debt.

## Competitive Landscape

The Competitor Ranking based on EVA provides a snapshot of financial strength relative to peers. Companies like AAL and AGCL lead the rankings, which could indicate stronger capital allocation strategies and more consistent value creation. This comparison allows users to quickly identify the market leaders and potential competitors for investment or partnership.

## Future Outlook

As the dashboard provides trends over time, it’s likely valuable for identifying emerging financial opportunities or threats. If current trends continue, sectors with improving EVA, increasing margins, and strong operational cash flow will likely outperform. On the other hand, industries with rising debt and declining interest coverage may be vulnerable during economic downturns or changes in market conditions.

## Conclusion

Overall, this dashboard provides a powerful combination of financial and operational metrics that allow for detailed company and industry analysis. Monitoring key trends like debt levels, profitability, asset growth, and efficiency metrics can help users assess market opportunities and risks, while the competitive insights offer a clear view of how well companies are managing their financial health compared to peers.

# SECTION 4

## Stock Price Prediction

Using Decision Tree, Random Forest Regressor and LSTM

## Data Reading and description

| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| 1/2/2015 | 27.8475 | 27.860001 | 26.8375 | 27.3325 | 24.37396 | 212818400 |
| 1/5/2015 | 27.0725 | 27.1625 | 26.352501 | 26.5625 | 23.687307 | 257142000 |
| 1/6/2015 | 26.635 | 26.8575 | 26.157499 | 26.565001 | 23.689531 | 263188400 |
| 1/7/2015 | 26.799999 | 27.049999 | 26.674999 | 26.9375 | 24.021715 | 160423600 |
| 1/8/2015 | 27.307501 | 28.0375 | 27.174999 | 27.9725 | 24.944687 | 237458000 |

**Date**: Displays the specific day corresponding to the stock data, indicating when the trading activity (open, high, low, close prices) occurred.

**Open**: Shows the price at which the stock begins trading at the start of the session for the day.

**High**: Shows the highest price reached by the stock during the trading session.

**Low**: Shows the lowest price reached by the stock during the trading session.

**Close**: Shows the final price at which the stock trades at the end of the session.

**Adjusted (Adj) Close**: The closing price of a stock adjusted for dividends, stock splits, and other corporate actions to reflect its true historical value.

# Data Preparation and Processing

**1- Handling Missing Values:**

- Check for Null Values:
  - **Identify** and address any missing or null values in our dataset to ensure data integrity and accuracy.
  - **Impact:** Missing values can skew analysis and model performance, leading to inaccuracies.
  - **Handling Methods**: Depending on the context, address missing values through imputation (mean, median, mode) or by removing affected rows/columns if they are not crucial.

**2- Data Normalization:**

- Normalize Data Using MinMaxScaler:
  - **Concept:** The MinMaxScaler is a normalization technique that transforms features by scaling them to a fixed range, usually between 0 and 1. This is achieved by subtracting the minimum value of the feature and dividing by the range (difference between the maximum and minimum values)
- Benefits:
  1. **Ensures** that all features contribute equally to the model's performance.
  2. **Facilitates** faster convergence of optimization algorithms.
  3. **Helps** in improving the overall performance of machine learning models by making data comparable.

**3- Data Splitting:**

- Split Data into Training and Validation Sets:
- Purpose:
  1. **Evaluate Model Performance**: Assess how well the model performs on unseen data to gauge its effectiveness and generalizability.
  2. **Prevent Overfitting**: Ensure that the model does not learn the noise or specific details of the training data too well, which could impair its performance on new data.

**4- Conversion to NumPy Arrays:**

- Advantages:
  1. **Performance**: NumPy arrays offer faster computations and are more memory-efficient compared to other data structures.
  2. **Compatibility**: They integrate seamlessly with most machine learning libraries and frameworks.
  3. **Mathematical Operations**: Enable rapid and efficient mathematical operations, essential for large-scale computations.
  4. **Data Manipulation**: Simplify the process of data manipulation and reshaping, making it easier to prepare data for various modeling tasks.

# Using of Decision Tree & Random Forest Regressor.

## 1- Decision Tree:

### Model Overview:

- The Decision Tree model was used to predict Apple’s stock prices over time based on historical data.

- The model splits data into branches by making decisions on stock price patterns, ultimately leading to a predicted price.

### Performance Evaluation:

- The chart shows a comparison between Actual Price (blue) and Predicted Price (red).

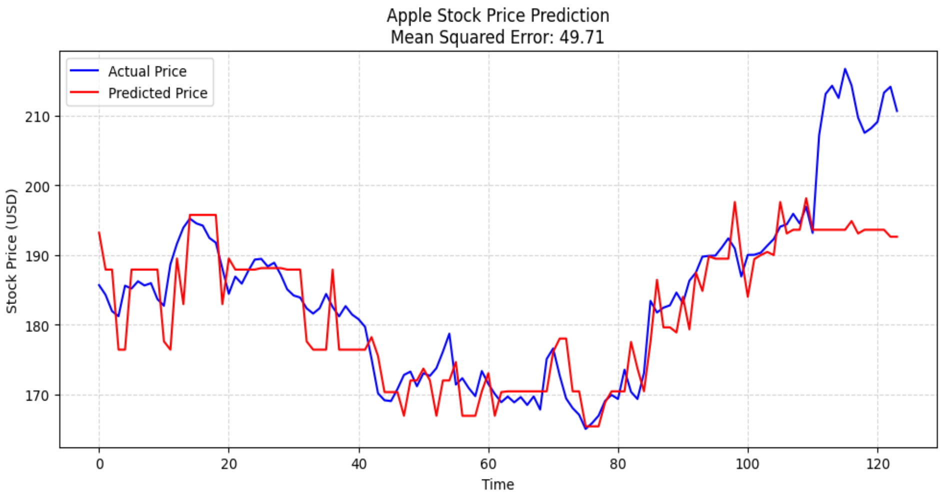
- The accuracy of the model is measured using Mean Squared Error (MSE). In this case, the model produced an MSE of 49.71, indicating that there is a noticeable gap between actual and predicted prices.

### Explanation:

- Decision Trees are simple and interpretable but prone to overfitting and can be unstable when the data fluctuates.

- This model captures trends to a certain extent but misses out on large price spikes or drops, as shown in the graph.

Stock Price Prediction using Decision Tree Model



Source: Stock_Price_Prediction_V_2_(1)

## 2- Random Forest Regressor:

### Model Overview:

- The Random Forest Regressor is an ensemble learning model that builds multiple decision trees and averages their predictions to reduce overfitting and improve accuracy.

### Performance Evaluation:

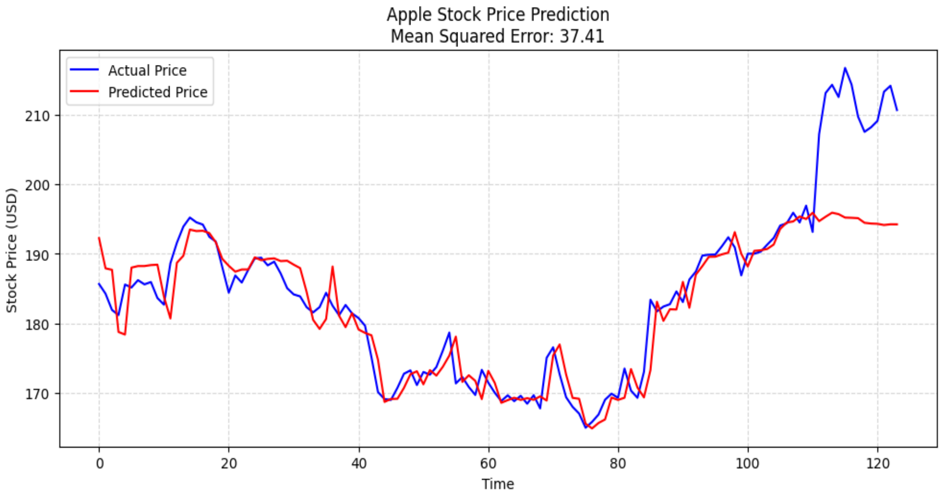
- With a Mean Squared Error of 37.41, the model has room for improvement, possibly through tuning hyperparameters or using different models like **LSTMs**, which might handle stock price time-series data better.

### Explanation:

- In this graph, we see the actual Apple stock price plotted against the predicted price. While the Random Forest model captures the general trend, it sometimes deviates from the actual price, as seen in the error at certain time points.

- The Random Forest Regressor effectively reduces overfitting, but it may not capture highly complex stock price patterns as well as time-series models.

Random Forest Regressor results



Source: Stock_Price_Prediction_V_2_(1)

# Neural Networks – From Origins to Modern Applications

## Description:

Neural Networks are machine learning algorithms inspired by the human brain's structure. They consist of layers of interconnected neurons designed to recognize patterns in data and solve complex problems.

## Explanation:

- Neural networks were originally developed to mimic brain functions, with the goal of enabling machines to learn from data and improve their performance.
- They became popular in the **1980s and 1990s**, with applications in speech and image recognition, but fell out of favor due to limitations in computational power and data availability.

## Resurgence:

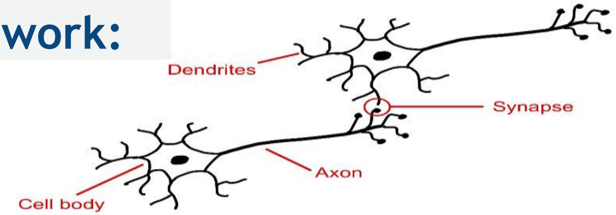
- Around 2005, neural networks saw a resurgence thanks to advances in technology, including powerful GPUs, access to large datasets, and the refinement of learning algorithms.
- Neural networks are now used extensively in modern applications like **speech recognition**, **image processing**, and **text-based Natural Language Processing (NLP)**.

## Conclusion:

- Neural networks have become one of the most powerful tools in **artificial intelligence (AI)**, revolutionizing how we handle complex tasks involving speech, images, and language.

## Biological Neural Network:

- Neurons in the brain process signals, inspiring artificial neural networks for data analysis.



## Artificial Neural Network (ANN).

### Model Overview:

- An Artificial Neural Network (ANN) is a computational model designed to mimic the way human brains process information. It consists of layers of interconnected neurons that adjust weights to make predictions based on the input data.

### Explanation of Layers:

- In this diagram, we have four input nodes (green) representing different features fed into the network. These inputs are processed in the hidden layer (blue), which learns complex patterns through weighted connections. The output layer (red) gives us the final prediction.

### Mechanism of Learning:

- The network works by adjusting weights during training to minimize the difference between predicted and actual outputs. It uses an activation function to introduce non-linearity, allowing the model to learn more sophisticated relationships in the data.

### Comparison with Traditional Models:

- Compared to traditional models like Random Forest, ANNs excel at handling complex and non-linear data, making them a strong choice for tasks like image recognition or stock price prediction when the underlying relationships are intricate.

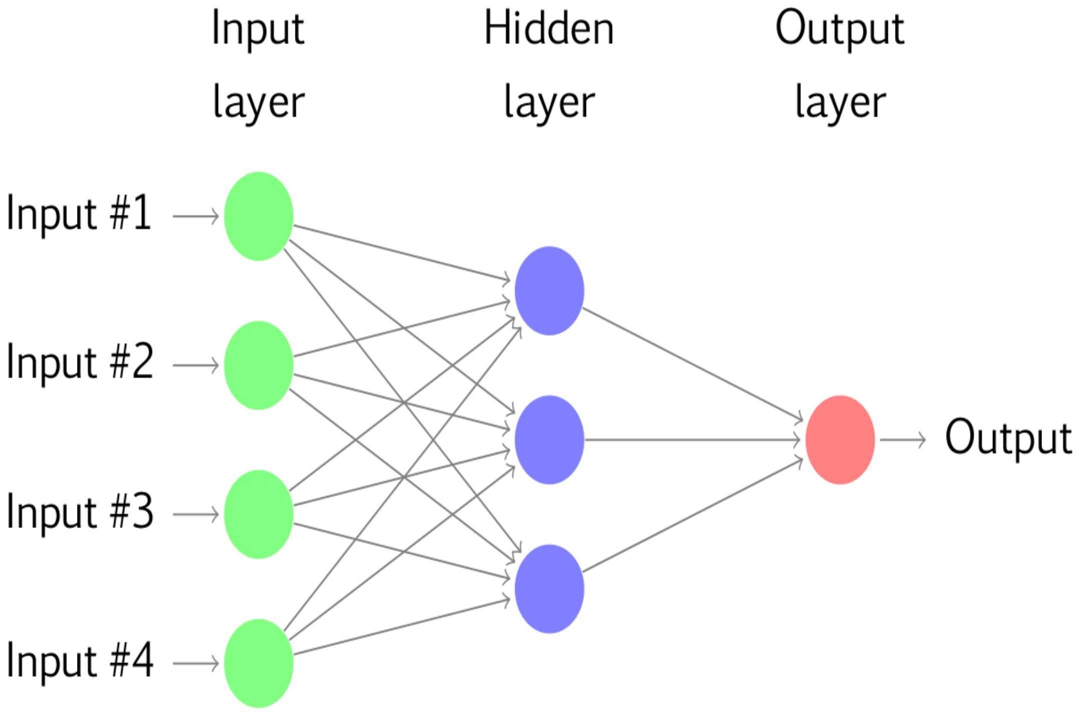
### Key Features of MLP:

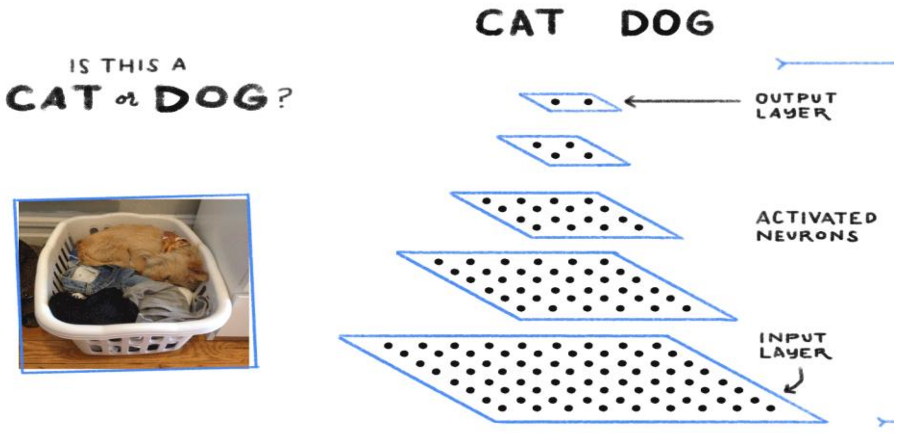
- Input layer, hidden layer(s), output layer: It consists of an input layer, one or more hidden layers, and an output layer.

- Fully connected network: Every node (neuron) in one layer is connected to every node in the next layer.

- Feedforward architecture: Information flows in one direction, from the input layer, through the hidden layers, to the output layer.

Multilayer Perceptron (MLP)





Source:
https://galaxyinferno.com/explaining-the-components-of-a-neural-network-ai/
https://mohameddhaoui.github.io/deeplearning/LSTM/

## Linear Regression Equation

### Description:

The **Linear Regression Equation** is used to predict a continuous dependent variable based on one or more independent variables. The equation for simple linear regression (with one independent variable) is:

$$f(x)= wx + b$$

### Components of the Equation:

- **f(x): This is the predicted output or the dependent variable. It's the value we are trying to predict.**
- **w (weight or slope): This represents the slope of the line. It determines how much the predicted value of f(x) changes for a unit change in the independent variable, x.**
- **x (input feature or independent variable): This is the input feature or variable we use to make predictions. In a simple linear regression, we have only one independent variable.**
- **b (bias or intercept): This is the y-intercept, which is the value of f(x) when x is zero. It adjusts the height of the regression line.**

**Linear regression finds the relationship between the dependent variable (f(x)) and the independent variable (x) by fitting a line to the data. The goal is to determine the values of w (slope) and b (intercept) that minimize the difference between the predicted values (f(x)) and the actual observed values in the data.**

### Why it Matters:

- **Prediction: Once the line is fitted to the data, you can predict the output (f(x)) for new values of x.**
- **Interpretability: The equation gives a simple and intuitive understanding of how the input (x) affects the output (f(x)).**
- **Real-world Applications: Linear regression is used in various fields like finance, healthcare, and economics to make predictions based on historical data.**

### Visualization:

If we imagine a scatter plot of data points (x, f(x)), the linear regression model attempts to draw a straight line that best fits this data. The slope (w) and intercept (b) determine the position and angle of this line.

# Why LSTM (Long Short-Term Memory)?

## Description:

LSTMs are powerful tools in machine learning, particularly when dealing with time-dependent data. They excel at capturing long-term dependencies, managing complex and noisy data, and are widely used in tasks such as stock price prediction, natural language processing (NLP), and speech recognition. This makes them highly valuable in applications where sequence and time context play a crucial role.

## Great for Time Series Data:

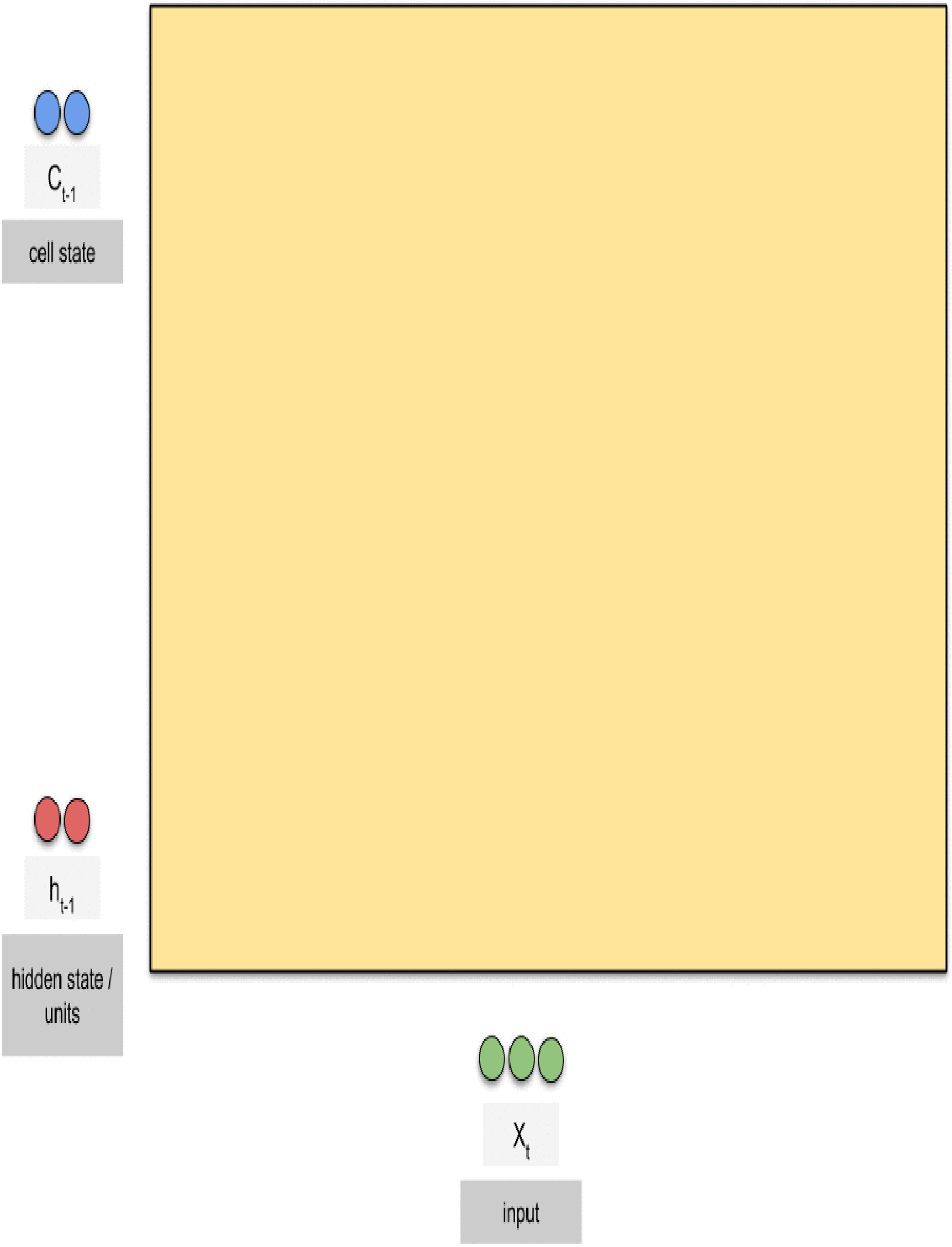
- Designed to handle sequential data like stock prices.
- Ideal for predictions where temporal order matters.

## Captures Long-Term Patterns:

- Learns and remembers patterns over long periods.
- Effective in predicting future trends by leveraging both short-term and long-term data dependencies.

## Handles Complex Data:

- Robust in understanding complex relationships within data.
- Performs well even with noisy and volatile datasets like stock market data.



Source: https://www.researchgate.net/publication/13853244_Long_Short-term_Memory

## LSTM Architecture

- LSTM (Long Short-Term Memory) is a special kind of RNN designed to retain information over long sequences.

- Useful for tasks like speech recognition, language modeling, and time-series predictions.

### Key Inputs:

- X t : Current input vector at time step t.

- h t−1 : Hidden state from the previous time step, carrying information from past steps.

- C t−1 : Cell state from the previous time step, representing long-term memory.

### Outputs:

- C t : Updated cell state that stores memory.

- h t : Hidden state or output at the current time step, passed to the next unit or used as output.

### Core Components:

- Forget Gate (σ): Decides which part of the previous memory to forget.

- Input Gate (σ): Determines how much of the new input should be written into the memory.

- Output Gate (σ): Controls how much of the memory will be used to generate the hidden state.

### Cell State Update:

- The previous cell state C t−1  is updated based on:

- Information from the forget gate.
- Contribution of the input gate, based on the new input.

### Activation Functions:

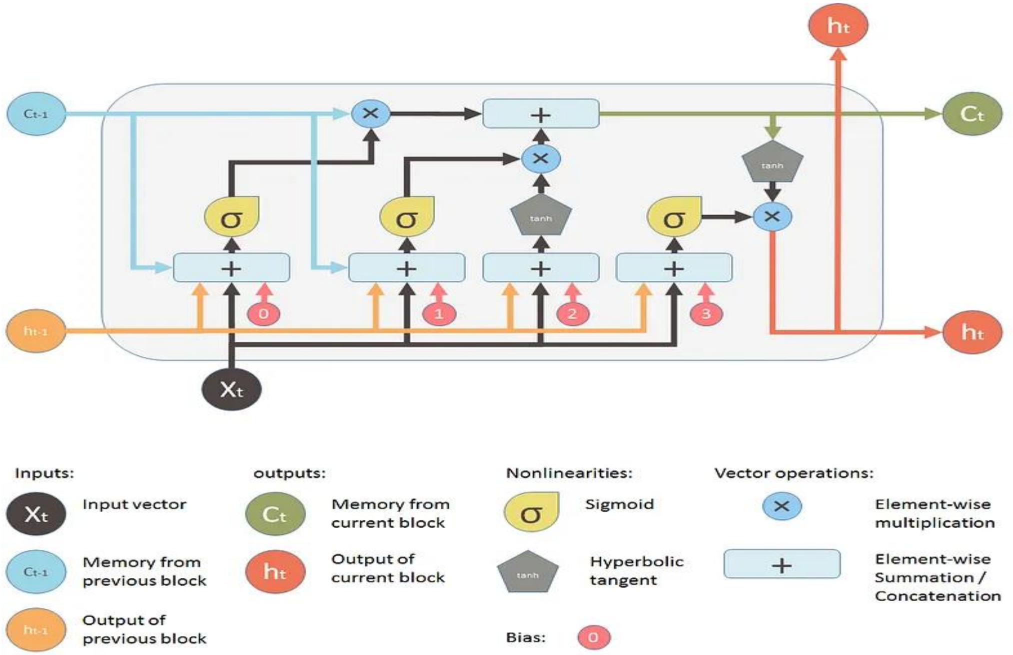
- Sigmoid (σ): Used in gates to scale values between 0 and 1 (deciding how much information passes through).

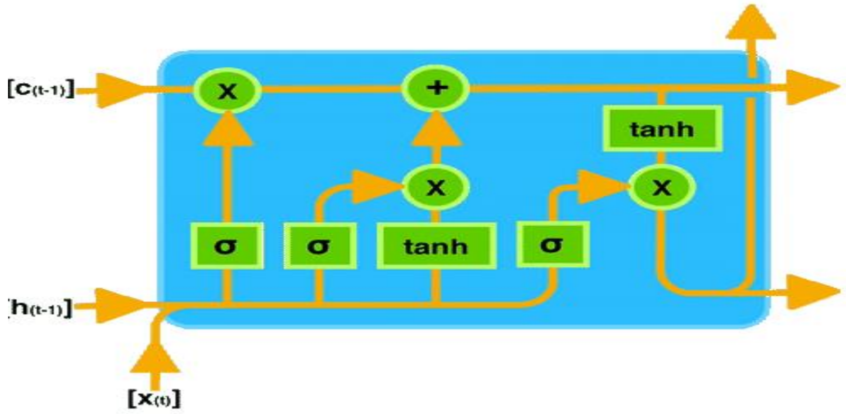
- Tanh (tanh): Scales input between -1 and 1, adding non-linearity for better learning.

### Operations:

- Element-wise Multiplication (×): Used in the gates to control information flow.

- Element-wise Addition (+): Combines information from the forget and input gates to update the memory.





Advantages of LSTM:

- Overcomes limitations of traditional RNNs by effectively handling long-term dependencies.

- Gating mechanisms allow selective forgetting and updating of information.

Source: https://medium.com/analytics-vidhya/introduction-to-long-short-term-memory-lstm-a8052cd0d4cd

# LSTM Model Architecture Description

## Purpose of the Model:

This LSTM model is designed to process time-series data, learning patterns across sequences of 100-time steps with 1 feature per time step. The architecture is ideal for tasks like time-series forecasting or sequence classification.

## LSTM Layers:

Each LSTM layer learns representations of the input data, with the dimensionality increasing from 30 to 50 features per time step as you go deeper into the model. The final LSTM layer outputs only the last time step (a single vector), which is then passed to the dense layer for final prediction.

## Dropout Layers:

Dropout is applied after each LSTM layer to reduce overfitting by randomly dropping neurons during training, ensuring the model generalizes better to unseen data.

## Dense Layer:

The final dense layer converts the 50 features from the last LSTM layer into a single output, suitable for making predictions.

## Why LSTM Layers?

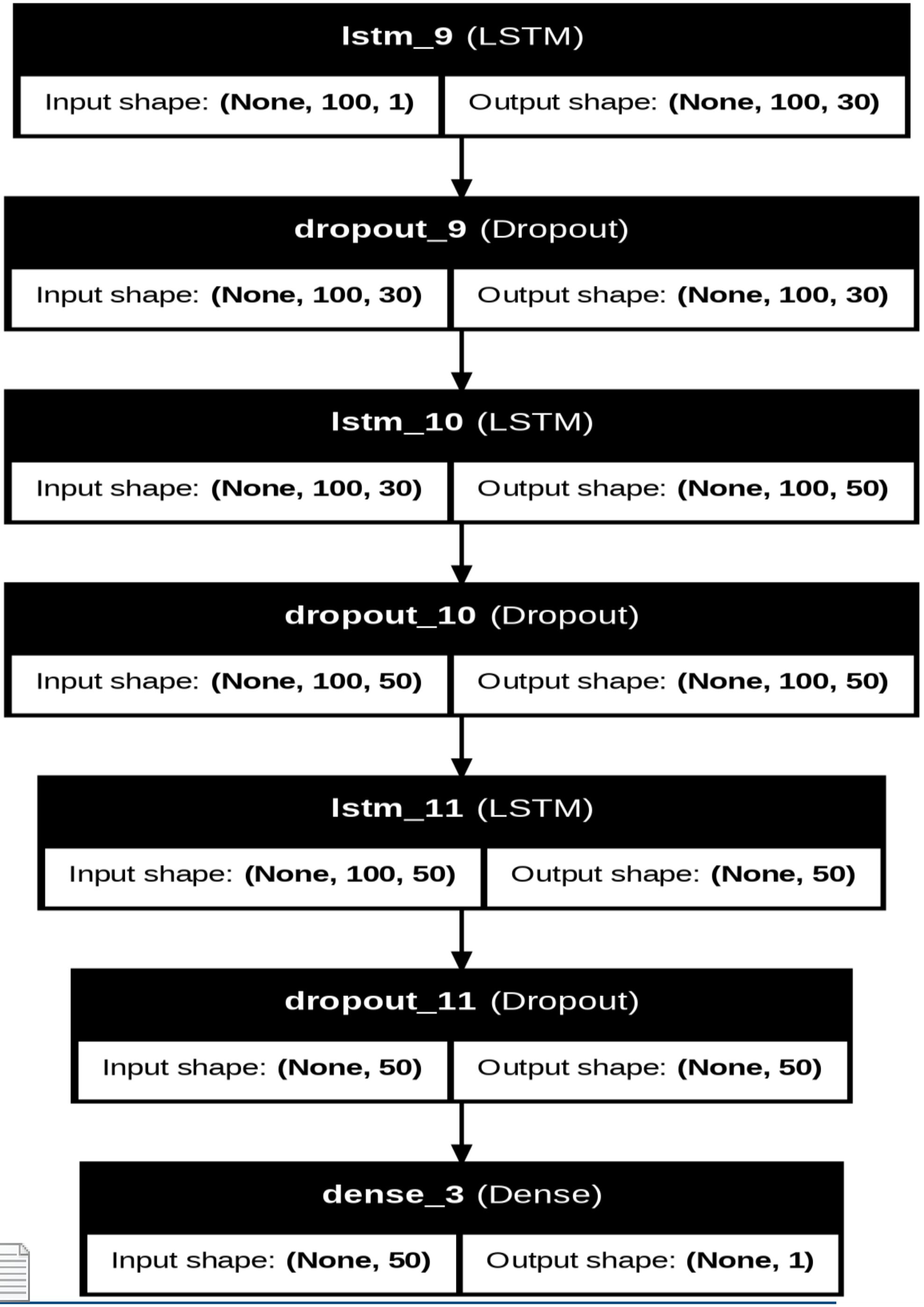
LSTM layers are used because of their ability to handle long-term dependencies in sequences. They "remember" important information over many time steps.

## Why Dropout Layers?

Dropout is crucial in deep networks to prevent overfitting, particularly when working with small datasets or highly complex models.

## Final Output:

The model outputs a single value, which is typically used in tasks like regression (predicting a continuous value) or binary classification (classifying into two categories).



**lstm_9** (LSTM)
Input shape: **(None, 100, 1)** | Output shape: **(None, 100, 30)**

**dropout_9** (Dropout)
Input shape: **(None, 100, 30)** | Output shape: **(None, 100, 30)**

**lstm_10** (LSTM)
Input shape: **(None, 100, 30)** | Output shape: **(None, 100, 50)**

**dropout_10** (Dropout)
Input shape: **(None, 100, 50)** | Output shape: **(None, 100, 50)**

**lstm_11** (LSTM)
Input shape: **(None, 100, 50)** | Output shape: **(None, 50)**

**dropout_11** (Dropout)
Input shape: **(None, 50)** | Output shape: **(None, 50)**

**dense_3** (Dense)
Input shape: **(None, 50)** | Output shape: **(None, 1)**

For More Explanation >>>

LSTM model Layer-by-Layer Breakdown.txt

# LSTM model's performance on Price Prediction.

This chart illustrates the results of the LSTM model used for predicting share prices over time. The model compares the actual share prices (blue line) with the predicted share prices (red line).

## Model Performance:

The relatively low MSE of 25.99 demonstrates that the model captures the general trend of the market, but some short-term fluctuations are missed

## Strengths of LSTM in Stock Prediction:

- LSTM models are effective at learning and predicting long-term trends in stock prices.

- They can handle complex sequential data, including market cycles and seasonal effects.

- The model is less likely to overfit compared to simpler models due to its ability to capture long-term dependencies.

## Challenges and Limitations:

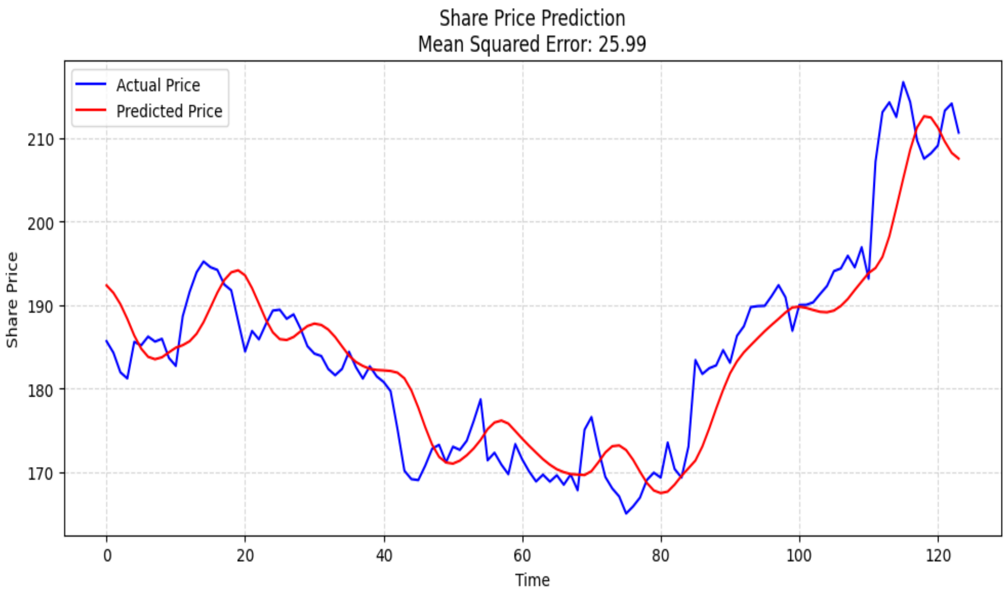
- Market noise and sudden price spikes may not be captured effectively, as seen in the deviations between the actual and predicted lines.

## Potential Improvements:

- To improve the model's accuracy, techniques like hyperparameter tuning, ensemble models, or integrating additional features (such as market sentiment or external economic data) could be explored.

- Regular retraining of the model with fresh data can also help improve its performance as market conditions change.

Stock Price Prediction using LSTM Model



Source: Stock_Price_Prediction_V_2_(1)

# EXECUTIVE SUMMARY

This project aimed to address the complex challenge of stock price prediction within the U.S. market, focusing on S&P 500 companies, by integrating machine learning techniques with fundamental financial analysis. In recent years, the unpredictable nature of the stock market, driven by factors such as market sentiment, speculation, and external economic conditions, has necessitated the development of more robust and accurate models. Traditional models often fail to account for these factors, prompting us to explore cutting-edge methodologies that combine both historical data and fundamental financial health metrics for comprehensive stock price forecasting.

We began by collecting a large dataset from S&P 500 companies using Python's Yfinance library. The dataset included a wide range of financial information such as historical stock prices, trading volumes, market capitalization, balance sheets, income statements, and cash flow statements. This vast collection of raw data was then processed through multiple phases of data cleaning and normalization. Missing values, outliers, and inconsistencies were handled systematically using Python scripts, ensuring that only high-quality data was fed into the predictive models. Without clean, structured data, machine learning models tend to produce unreliable results, which is why this stage was a critical foundation of the project.

Once the data was prepared, we employed several machine learning models, each tailored to capture different facets of stock price behavior.

- Decision Trees were among the first models tested due to their simplicity and interpretability. This model works by splitting data into branches, making decisions based on stock price trends. However, while the Decision Tree model provided a clear representation of historical patterns, it struggled with the nuances of volatile stock market behavior. Its propensity to overfit the training data, especially in scenarios with short-term price spikes, made it less reliable for generalizing to future predictions.
- To counteract the limitations of Decision Trees, we implemented a Random Forest Regressor. This ensemble learning model aggregates predictions from multiple Decision Trees, significantly improving accuracy by reducing overfitting and variance. The Random Forest model was able to better capture general price movements by smoothing out the predictions. However, despite these improvements, the model still faced difficulties with extreme market volatility, and short-term trends remained challenging to predict with high precision.
- Recognizing the inherent time-series nature of stock price data, we advanced to a more complex model: Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) specifically designed for sequential data. LSTMs are equipped with internal memory that allows them to retain information over long periods, making them ideal for predicting patterns in stock prices. The LSTM model excelled in identifying long-term trends and was particularly effective in predicting cyclical stock movements that other models missed. However, it wasn't without challenges: sudden, sharp fluctuations in the market—driven by external factors like economic announcements or unexpected news—still presented difficulties, as these events are not easily anticipated by historical data alone.

In parallel to our machine learning efforts, we conducted a comprehensive fundamental analysis using Power BI. This analysis focused on key financial metrics such as Economic Value Added (EVA), Return on Assets (ROA), and interest coverage ratios. EVA, for instance, measures how much value a company creates beyond its cost of capital, providing insights into the operational efficiency of the business. ROA assesses how well a company is utilizing its assets to generate profit, while interest coverage ratios help gauge a company's ability to pay interest on its outstanding debt. These metrics were visualized in Power BI dashboards to compare companies and sectors, highlighting trends over time.

# EXECUTIVE SUMMARY

A crucial finding from our analysis was that while financially sound companies often display stability and long-term growth, stock price movements—especially in the short term—are not always directly correlated with these fundamentals. This divergence between financial health and stock price behavior suggests the significant influence of speculative and macroeconomic factors on stock market performance.

From a project management perspective, we adopted an Agile methodology using Trello to coordinate and manage tasks. The project was broken down into several sprints, each focused on different aspects of the workflow: data extraction, data cleaning, model training, evaluation, and final analysis. Weekly meetings were held to review progress, and task assignments were dynamically updated based on the needs of the project. Trello enabled seamless collaboration among team members, helping us manage dependencies and stay on schedule. The sprint-based structure ensured that deliverables were completed in an iterative manner, allowing us to refine our models and analysis at each stage. This iterative approach was essential in fine-tuning the machine learning models to achieve better performance over time.

Throughout the project, we encountered several challenges that highlighted the complexity of stock price prediction. The unpredictability of market sentiment, driven by news cycles, investor psychology, and macroeconomic conditions, often caused deviations between predicted and actual stock prices. While machine learning models like LSTM demonstrated a greater ability to handle sequential data and long-term trends, they still struggled to account for abrupt changes in the market. This limitation points to the necessity of incorporating additional data sources, such as sentiment analysis or real-time economic indicators, into the predictive framework.

**Key Findings**:

- Machine learning models such as LSTM showed strong performance in predicting general market trends and long-term price movements, but they were less effective in forecasting short-term price volatility driven by external factors.
- Fundamental financial metrics like EVA, ROA, and interest coverage ratios provided valuable insights into the long-term stability and health of companies but did not always correlate with short-term stock price fluctuations. This indicates that stock prices are often influenced by a combination of fundamental performance and external market forces such as sentiment and speculation.

Conclusion: This project underscores the transformative potential of machine learning in financial markets, particularly when combined with fundamental analysis. Our findings suggest that while models like LSTM can provide significant insights into market behavior, they must be supported by other analytical techniques to account for the full spectrum of factors that drive stock prices. The volatility and unpredictability of the stock market, particularly in the short term, mean that no single model can perfectly forecast price movements. A multi-faceted approach, integrating machine learning, financial fundamentals, and market sentiment, is essential for creating more accurate and reliable stock price predictions.

In the future, expanding the model to incorporate real-time market sentiment data—such as social media sentiment, news headlines, and economic indicators—could enhance its accuracy in predicting short-term market fluctuations. This would enable investors to make more informed decisions in increasingly volatile financial environments. The synergy between AI-driven predictive models and human decision-making will likely define the future of stock market analysis, providing a more robust framework for understanding and anticipating market trends.

1.91%
7.48%
862
103

# REFERENCES

1. Source: https://medium.com/analytics-vidhya/introduction-to-long-short-term-memory-lstm-a8052cd0d4cd
2. Source: https://www.researchgate.net/publication/13853244_Long_Short-term_Memory
3. https://galaxyinferno.com/explaining-the-components-of-a-neural-network-ai/
4. https://mohameddhaoui.github.io/deeplearning/LSTM/
5. Source: Stock_Price_Prediction_V_2_(1)
6. Source: V6_[Historical]Consolidating_Financial_Data_for_YFinance_Stocks.ipynb

DIGITAL EGYPT BUILDERS

# Major U.S. Market

## Stock Market Analysis