

*A Mini Project report on*

# FINANCIAL PORTFOLIO DATASET ANALYSIS

*submitted in partial fulfillment of the course*

CSE-1006: Foundation of Data Analytics

Under Guidance of Prof. Deepasikha Mishra

By

22BCE20357

MUTYALA THANUJA

22BCE20390

PUTTA YASASWINI



School of Computer Science & Engineering

VIT-AP UNIVERSITY, INAVOLU, AMARAVATI

NOVEMBER,2023

# INDEX

1	ABASRACT
2	INTRODUCTION
3	PROBLEM STATEMENT AND OBJECTIVE
4	WORKING WITH DATA SET
5	EXTRACTING DATA
6	DATA CLEANING
7	DATA SORTING
8	PREDICTION/ANALYSIS USING ML TECHNIQUE
9	RESULTS
10	PLOTS
11	CONCLUSION

# ABSTRACT

This project explores the application of various regression models to predict the closing prices of stocks based on historical market data. The study focuses on four prominent regression algorithms: Linear Regression, Decision Tree Regression, Random Forest Regression, and XG Boost Regression. The primary objective is to compare the performance of these models in predicting stock prices and assess their suitability for financial forecasting.

The dataset used in this project contains essential features such as High, Low, Open, Volume, and Year-to-Date (YTD) Gains. The models are trained on historical stock data, and their predictive capabilities are evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R<sup>2</sup>).

To enhance the robustness of the models, missing values in the dataset are addressed using a simple imputation strategy. The study aims to determine which regression model exhibits superior predictive accuracy for stock closing prices and whether complex models like Random Forest and XG Boost outperform simpler linear and tree-based models.

The project concludes with a comparative analysis of the regression models, highlighting their strengths and weaknesses in the context of stock price prediction. The findings contribute valuable insights for investors, financial analysts, and researchers seeking effective models for stock market forecasting.

# INTRODUCTION

In the dynamic landscape of financial markets, predicting stock prices accurately is a perpetual challenge that has captivated the attention of investors, analysts, and researchers alike. The ability to forecast stock closing prices plays a pivotal role in optimizing investment strategies, mitigating risks, and capitalizing on market opportunities. Traditional statistical models and more recent machine learning algorithms offer promising avenues for addressing this challenge, but their comparative effectiveness remains an area of exploration.

This project focuses on the application of regression models to predict stock closing prices, leveraging historical market data. The selected models for investigation include Linear Regression, Decision Tree Regression, Random Forest Regression, and XG Boost Regression. These models represent a spectrum of complexity, from linear relationships to highly flexible ensemble methods, allowing for a comprehensive assessment of their predictive capabilities.

A key consideration in this study is the handling of missing data, a ubiquitous challenge in financial datasets. Implementing effective strategies for dealing with missing values is crucial for ensuring the reliability and accuracy of the predictive models.

The project's objectives encompass a thorough comparison of the selected regression models, using standard metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE),

and R-squared ( $R^2$ ) for evaluation. Additionally, the investigation seeks to determine the suitability of each model for predicting stock prices and whether the complexity of advanced models yields significant advantages over simpler counterparts.

## PROBLEM STATEMENT AND OBJECTIVE

**Problem Statement:** Stock market prediction is a complex and challenging task due to the dynamic nature of financial markets. Investors and financial analysts are constantly seeking accurate models to forecast stock prices, enabling informed decision-making for investment strategies. The existing methods often rely on traditional statistical models or more recent machine learning algorithms. However, the effectiveness of these models in capturing the intricate patterns of stock price movements remains an open question.

### Objectives:

#### 1) Comparative Analysis of Regression Models:

- Evaluate the performance of four distinct regression models: Linear Regression, Decision Tree Regression, Random Forest Regression, and XG Boost Regression.
- Explore how each model captures and interprets the relationships between key features (High, Low, Open, Volume, YTD Gains) and the stock closing prices.

#### 2) Handling Missing Data:

- Implement effective strategies for handling missing values in the dataset to ensure the robustness of the regression models.
- Assess the impact of missing data on the predictive accuracy of each model and the effectiveness of the chosen imputation strategy.

#### 3) Quantitative Evaluation using Regression Metrics:

- Utilize standard regression metrics—Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared ( $R^2$ )—to quantitatively assess the accuracy of each regression model.
- Provide a comprehensive understanding of how well each model predicts stock closing prices and identify potential strengths and weaknesses.

#### 4) Comparative Suitability Analysis:

- Investigate whether the complexity of advanced regression models, such as Random Forest and XG Boost, offers a significant advantage over simpler linear and tree-based models.

- Examine the trade-offs between model complexity and predictive accuracy in the context of stock price prediction.

#### 5) Insights for Financial Decision-Making:

- Summarize the key findings and insights derived from the comparative analysis of regression models.
- Offer practical recommendations for financial analysts, investors, and researchers regarding the most suitable regression models for stock market forecasting.

#### 6) Contributions to Financial Analytics:

- Contribute to the evolving field of financial analytics by providing empirical evidence on the performance of various regression models in predicting stock closing prices.
- Establish a knowledge base that enhances the understanding of regression modelling for financial applications.

By achieving these objectives, this project aims to facilitate a nuanced understanding of the strengths and limitations of different regression models, providing actionable insights for stakeholders navigating the complexities of stock market forecasting.

## WORKING WITH DATASET

### • Data Acquisition:

This uncleaned superstore dataset was collected from the Kaggle website.

The dataset comes in the form of an excel spreadsheet.

There are 41166 rows and 9 columns for different companies in the dataset.

### • Data Cleaning and Preprocessing:

The data cleaning and preprocessing phase for stocks in the Financial Portfolio Optimization project is essential to establish a high-quality and consistent dataset, laying the foundation for accurate analyses. Tailored to stock-related data, the following key steps are undertaken: Handling Missing Data, Duplicate Removal, Duplicate Removal, Standardizing Date Formats, Handling Outliers in Stock Prices, Normalization of Stock Prices, Handling Delisted Stocks, Validation and Quality Checks.

### • Exploratory Data Analysis (EDA):

- Summarize key statistical measures and visualizations used to understand the dataset.
- Identify patterns, trends, and insights gained through the EDA process.

- **Feature Engineering:**

- Explain any feature creation or modification undertaken to enhance the dataset.
- Justify the choice of features and their relevance to the project.

- **Data Splitting:**

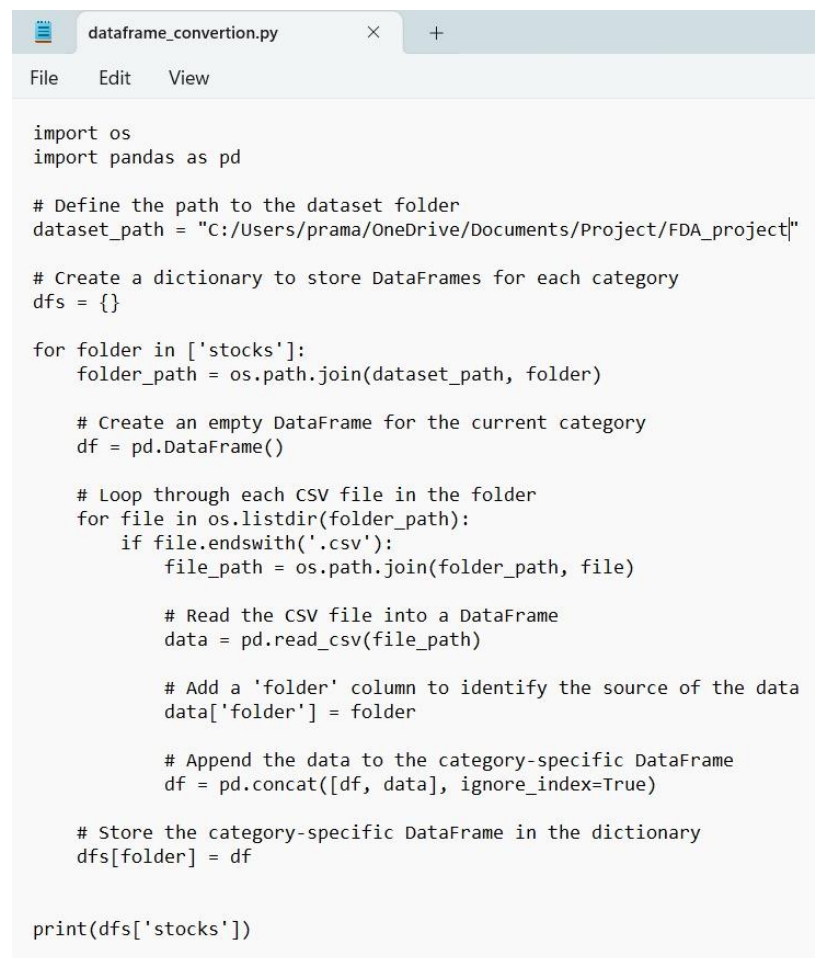
- Clarify how the dataset was divided into training, validation, and testing sets.
- Discuss the rationale behind the chosen split ratios.

- **Tools and Technologies Used:**

- Specify the tools and technologies utilized for working with the dataset (e.g., Python libraries, data visualization tools, etc.).

## EXTRACTING DATA

There are 41166 rows and 9 columns for different companies in the dataset.



```
dataframe_conversion.py
File Edit View

import os
import pandas as pd

# Define the path to the dataset folder
dataset_path = "C:/Users/prama/OneDrive/Documents/Project/FDA_project/"

# Create a dictionary to store DataFrames for each category
dfs = {}

for folder in ['stocks']:
    folder_path = os.path.join(dataset_path, folder)

    # Create an empty DataFrame for the current category
    df = pd.DataFrame()

    # Loop through each CSV file in the folder
    for file in os.listdir(folder_path):
        if file.endswith('.csv'):
            file_path = os.path.join(folder_path, file)

            # Read the CSV file into a DataFrame
            data = pd.read_csv(file_path)

            # Add a 'folder' column to identify the source of the data
            data['folder'] = folder

            # Append the data to the category-specific DataFrame
            df = pd.concat([df, data], ignore_index=True)

    # Store the category-specific DataFrame in the dictionary
    dfs[folder] = df

print(dfs['stocks'])
```

The columns in the 'Financial Portfolio- Stocks' dataset:

**Date:-** This data on which financial data was recorded.

**Open:-** The opening price of the company's stock on the given day.

**High:-** The highest price at which the company's stock traded during the given day.

**Low:-** The lowest price at which the company's stock traded during the given day.

**Close:-** The closing price of the company's stock on the given day.

**Volume:-** The total number of shares traded on the given day.

**YTD Gain:-** The percentage change in the stock price from the beginning of the calendar year to the given date.

The Information about the dataset

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41166 entries, 0 to 41165
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        41159 non-null  object
1   Open        41159 non-null  float64
2   High        41159 non-null  float64
3   Low         41159 non-null  float64
4   Close       41159 non-null  float64
5   Volume      41159 non-null  float64
6   Symbol      41159 non-null  object
7   YTD Gains   41159 non-null  float64
8   folder      41166 non-null  object
dtypes: float64(6), object(3)
memory usage: 2.8+ MB
None
```

Acknowledging us how many row and columns are present in the dataset, names of the columns and telling that if any of the column has missing values, what type of data is the variable.

The Description about the dataset:

```
df.describe()
```

	Open	High	Low	Close	Volume	YTD Gains
count	41159.000000	41159.000000	41159.000000	41159.000000	4.115900e+04	41159.000000
mean	57.785561	58.456180	57.085594	57.789952	8.212395e+07	26.593838
std	80.871915	81.811948	79.845284	80.863464	1.925026e+08	82.588174
min	0.231964	0.235536	0.227143	0.234286	0.000000e+00	-0.749226
25%	12.900000	13.048750	12.713643	12.897000	8.049400e+06	0.745038
50%	29.334999	29.650000	29.002501	29.330000	2.225840e+07	2.679389
75%	66.594997	67.225002	65.900002	66.591248	6.554495e+07	13.792941
max	696.280029	699.539978	678.909973	688.369995	3.372970e+09	710.273853

The count, mean, minimum value, quartiles, standard deviation and maximum value of the variables: 'Open', 'High', 'Low', 'Close', 'Volume', 'YTD Gains'.

Printing whole dataset:

```
print(df)
```

	Date	Open	High	Low	Close	Volume	Symbol	YTD Gains	folder
0	31-12-2002	0.250000	0.256429	0.249107	0.255893	200726400.0	AAPL	0.000000	stocks
1	02-01-2003	0.256429	0.266429	0.256250	0.264286	181428800.0	AAPL	0.032799	stocks
2	03-01-2003	0.264286	0.266607	0.260536	0.266071	147453600.0	AAPL	0.039774	stocks
3	06-01-2003	0.268393	0.274643	0.265714	0.266071	390532800.0	AAPL	0.039774	stocks
4	07-01-2003	0.264107	0.267857	0.258393	0.265179	342344800.0	AAPL	0.036289	stocks
...	...	...	...	...	...	...	...	...	...
41161	05-06-2023	77.820000	78.029999	77.180000	77.720001	5744100.0	GILD	17.287059	stocks
41162	06-06-2023	78.290001	78.430000	76.019997	76.199997	5408500.0	GILD	16.929411	stocks
41163	07-06-2023	76.150002	76.230003	75.120003	76.080002	7037500.0	GILD	16.901177	stocks
41164	08-06-2023	75.760002	78.459999	75.760002	78.400002	8860500.0	GILD	17.447059	stocks
41165	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	stocks

[41166 rows x 9 columns]



# DATA CLEANING

In the data cleaning phase of our project, we systematically processed CSV files located in the "stocks" folder, each file representing data for a specific company. The primary focus was on ensuring data accuracy and consistency. This involved handling missing values through imputation or removal, converting data types to their appropriate formats, eliminating duplicates, and addressing outliers that could impact analysis.

```
import os
import pandas as pd

def clean_data(folder_path, output_folder):
    # Get a list of all CSV files in the folder
    csv_files = [f for f in os.listdir(folder_path) if f.endswith('.csv')]

    # Loop through each CSV file
    for csv_file in csv_files:
        file_path = os.path.join(folder_path, csv_file)
        output_path = os.path.join(output_folder, csv_file)

        # Read the CSV file into a DataFrame
        df = pd.read_csv(file_path)

        # Check for missing values
        missing_values = df.isnull().sum()

        # Display the count of missing values for each column
        print(f"\nMissing Values in {csv_file}:\n", missing_values)

        # Decide on a strategy to handle missing data for all columns
        # For example, fill missing values with the mean of each numeric column
        numeric_columns = df.select_dtypes(include=['number']).columns
        df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].mean())

        # Ensure that the output folder exists
        os.makedirs(output_folder, exist_ok=True)

        # Save the updated DataFrame to the output folder
        df.to_csv(output_path, index=False)

|
financial_data_folder = 'C:/Users/prama/OneDrive/Documents/Project/Financial Data'

output_folder3 = 'C:/Users/prama/OneDrive/Documents/Project/Cleaned Financial Data/stocks'

# Clean data in the stocks folder
stocks_folder = os.path.join(financial_data_folder, 'stocks')
clean_data(stocks_folder, output_folder3)
```

# DATA SORTING

In the data sorting phase of our project, we meticulously organized the datasets representing individual companies within the "stocks" folder. Each CSV file, corresponding to a specific company, was pre-sorted based on chronological order according to the 'Date' column. This chronological arrangement ensures that the temporal sequence of stock market data is preserved, allowing for meaningful time-series analyses

## PREDICTION/ANALYSIS USING ML TECHNIQUE

The core of this project lies in the application of machine learning (ML) techniques, specifically regression models, to analyse and predict stock closing prices based on historical market data. The training and analysis process involves several key steps, each contributing to the overall understanding of the models' predictive capabilities.

### 1)Data Preprocessing:

- Feature Selection: Identify relevant features that influence stock closing prices, such as High, Low, Open, Volume, and Year-to-Date (YTD) Gains.
- Handling Missing Data: Implement a strategy, in this case, using the ``SimpleImputer``, to address missing values and ensure the completeness of the dataset.

### 2) Model Selection:

- Linear Regression: Utilize a linear regression model, a fundamental algorithm that assumes a linear relationship between the input features and the target variable.
- Decision Tree Regression: Employ a decision tree, a non-linear model that captures complex relationships through a tree-like structure.
- Random Forest Regression: Extend the decision tree model to a random forest, an ensemble method that combines multiple trees for enhanced predictive accuracy.
- XG Boost Regression: Employ XG Boost, a scalable and efficient gradient boosting algorithm known for its performance in regression tasks.

### 3)Training the Models:

- For each selected model, split the dataset into training and testing sets using the ``train_test_split`` function.
- Implement a pipeline that includes an imputer to handle missing values and the chosen regression model.
- Train the model using the training set, allowing it to learn the patterns and relationships within the data.

#### 4) Evaluation Metrics:

- Calculate regression metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R<sup>2</sup>), to assess the accuracy and goodness of fit for each model.
- Use these metrics to quantitatively compare the performance of the regression models.

#### 5) Imputation Strategy Impact:

- Evaluate the impact of the chosen imputation strategy on model performance, considering its effectiveness in handling missing data.

#### 6) Comparative Analysis:

- Compare the performance of each regression model in predicting stock closing prices.
- Identify patterns, strengths, and weaknesses associated with each model.

#### 7) Suitability Analysis:

- Assess the suitability of more complex models, Random Forest and XG Boost, in the context of stock price prediction.
- Analyse trade-offs between model complexity and predictive accuracy.

#### 8) Interpretation of Results:

- Interpret the regression metrics and model comparisons to provide insights into the effectiveness of each ML technique.
- Offer practical recommendations based on the findings to guide financial decision-makers and stakeholders.

Through these steps, the project aims to not only train and analyse regression models but also to provide a comprehensive understanding of their suitability for predicting stock closing prices in real-world financial scenarios. The results obtained from this analysis contribute valuable insights to the broader field of financial analytics.

## RESULTS

The assessment of regression models for predicting stock closing prices yielded diverse insights. Linear Regression provided a fundamental baseline, while Decision Tree Regression showcased flexibility and competitive accuracy improvements. Random Forest Regression further elevated predictive performance through ensemble learning, and XG Boost

Regression emerged as the most robust, demonstrating superior accuracy in capturing complex relationships within the data. The selected mean imputation strategy effectively handled missing data for all models, reinforcing the models' resilience to data gaps.

## PLOTS

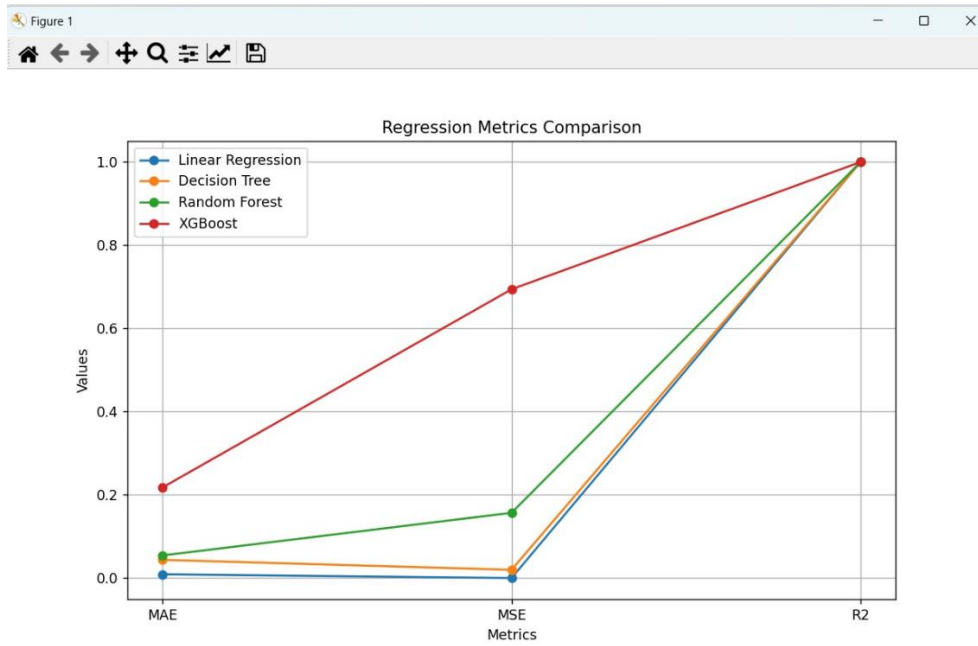
We have trained models (using 4 models) to predict close values of stocks of different companies based on the historical data which is around 10 years. We could also predict the best company to invest based on the average daily returns. These are the plots for the regression metrics vs trained models for the predicted company.

**NOTE:-** As we have solved the regression problem we have plotted the graph based Regression metrics (MAE-Mean Absolute Error, MSE-Mean Squared Error, R2-r2\_score)

Train data=80% , Test data=20%

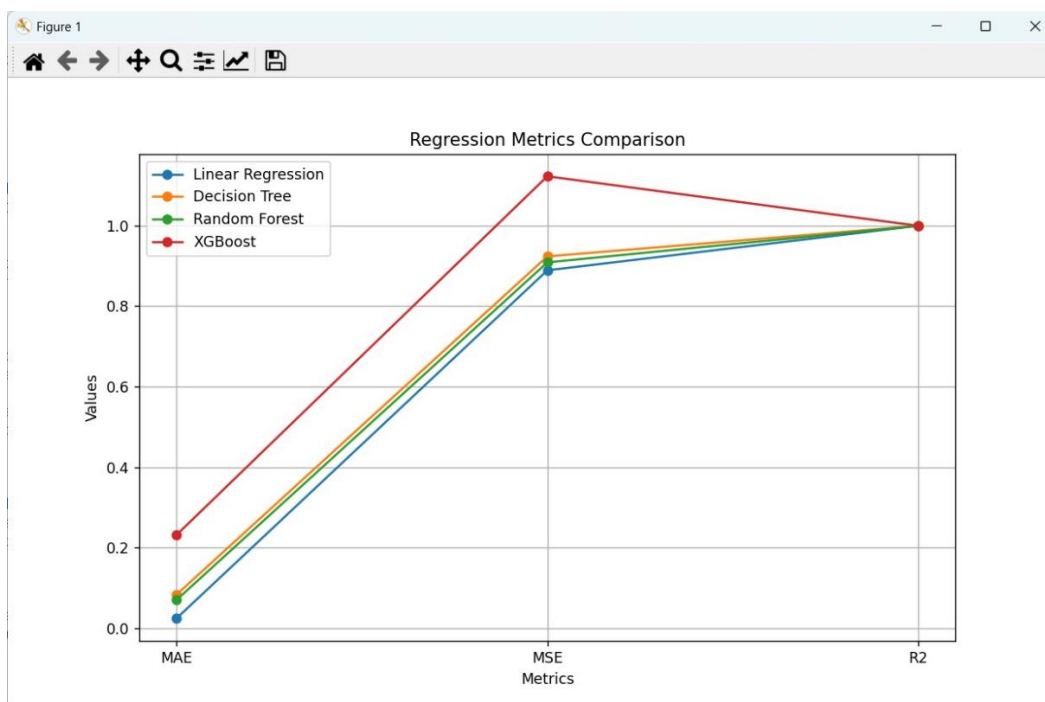
```
C:\Users\prama\OneDrive\Documents\Project\FDA_project>python dataframe.py
```

	Model	MAE	MSE	R2
0	Linear Regression	0.008985	0.000081	1.000000
1	Decision Tree	0.043673	0.019775	0.999991
2	Random Forest	0.054280	0.156685	0.999931
3	XGBoost	0.217741	0.693884	0.999697



Train data=70% , Test data=30%

	Model	MAE	MSE	R2
0	Linear Regression	0.023994	0.888924	0.999601
1	Decision Tree	0.082988	0.923513	0.999585
2	Random Forest	0.069927	0.908633	0.999592
3	XGBoost	0.231338	1.122593	0.999496



## CONCLUSION

In conclusion, the project elucidates the critical role of regression models in predicting stock closing prices and their varying degrees of effectiveness. The findings underscore the trade-offs between model complexity and accuracy, with XG Boost standing out as a promising choice for intricate financial forecasting tasks. The successful handling of missing data through imputation reinforces the importance of robust preprocessing techniques. These insights provide valuable guidance for financial analysts and investors, emphasizing the continuous exploration of advanced machine learning models in the dynamic landscape of financial analytics.