



Cairo university
Faculty of computers and artificial intelligence
Operations research and decision support department

**Predicting and evaluating the risk
for the stock market**

The Graduation Project Submitted to The Faculty of Computers
and Artificial Intelligence, Cairo University
In Partial Fulfillment of the Requirements
for the Bachelor Degree
in
Operations Research and Decision Support

**Under supervision of
Dr. Ghada tolan**

**Cairo University
JULY,2024**



Predicting and evaluating the risk for the stock market

Basmalla Zakarya	20210598
Nada Mamdouh	20200593
Mohanad Arafa	20200563
Sherif Yasser	20200764
Mohamed Ahmed	20200423

Supervised By

Dr. Ghada Tolan

ABSTRACT

The prediction of stock market data is a complex and widely studied field in financial economics. Traditionally, investors and analysts have relied on historical data and fundamental analysis to forecast future stock prices. With the advent of advanced computing technologies and the proliferation of data, more sophisticated techniques, such as machine learning and deep learning, have been employed to improve the accuracy of these predictions. These methods leverage vast amounts of historical market data, along with other economic indicators, to identify patterns and trends that might not be discernible through traditional analysis.

In this project, we implemented a machine learning-based approach to predict stock market prices. The methodology involved the collection of historical stock market data from various financial databases, including daily stock prices, trading volumes, and relevant economic indicators. Data preprocessing steps were performed to handle missing values, normalize data, and engineer features that could enhance the predictive model's performance. We experimented with several machine learning algorithms, including linear regression, decision trees, and neural networks, to determine the most effective model. Hyperparameter tuning and cross-validation techniques were employed to optimize the models and prevent overfitting.

The final phase of the project focused on testing and validating the predictive models. We utilized tools such as Python's Scikit-learn and TensorFlow libraries for model development and evaluation. The models were tested on out-of-sample data to assess their predictive accuracy. The results indicated that our models could successfully capture market trends and provide reasonably accurate stock price forecasts. This project not only demonstrates the potential of machine learning in financial predictions but also highlights the importance of rigorous testing and validation in achieving reliable results. The achievements of this project pave the way for more advanced and automated stock market prediction systems, contributing to more informed and strategic investment decisions.

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine / original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade, and it may result in withdrawal of our bachelor's degree

Group members:

Name

Signature

Nada Mamdouh Mohamed

Basmalla Zakaria sliem

Mohanad Arafa

Sherif Yasser

Mohamed Ahmed

Table of Contents

ABSTRACT	3
DECLARATION	4
LIST OF FIGURES	7
CHAPTER 1	8
INTRODUCTION	8
1.1. Introduction	9
1.2. Problem domain	9
1.2.1. Challenges in Stock Market Prediction:	10
1.3. Problem statement	12
1.4. Objectives.....	12
1.5. Resources required.....	12
1.5.1. Data resources.....	12
1.5.2. Software resources	13
1.5.3. Hardware resources	13
1.6. Development methodology.....	13
1.7. Project report	14
1.7.1. Chapter 1	14
1.7.2. Chapter 2	14
1.7.3. Chapter 3	14
1.7.4. Chapter 4	15
1.7.5. Chapter 5	15
1.7.6. Chapter 6	15
CHAPTER 2	16
BACKGROUND/EXISTING WORK	16
2.1. Stock Market Prediction Using Machine Learning (LSTM):	17
2.1.1. overview	17
2.1.2. Regression Based Model and LSTM on 9 Lakh Data Records	17
2.1.3. LSTM Model and Results.....	17
2.1.4. conclusion:	18
2.2. Analysis and prediction of stock market trends using deep learning:	18
2.2.1. Overview	18
2.2.2. Methods	19
2.2.3. Conclusion and future work	19

2.3. Prediction and Analysis of Corporate Financial Risk Assessment Using Logistic Regression Algorithm in Multiple Uncertainty Environment:	20
2.3.1. introduction:	20
2.3.2. Methodology	20
2.3.3. Results	21
2.3.4. Conclusion	21
2.4. Prediction model for stock market analysis:	22
2.4.1. project overview	22
2.4.2. Algorithms used	22
CHAPTER 3.....	24
DATA Analysis	24
3.1. Data collection:	25
3.2. Data preprocessing:	26
3.2.1. Data Cleaning	26
3.3. Exploratory data analysis (EDA):	26
3.4. Data visualization	29
CHAPTER 4.....	38
METHODOLOGY	38
4.1. Algorithm used	39
4.2. The implementation	39
4.3. The results	41
4.4. Model Evaluation	42
CHAPTER 5.....	43
USER INTERFACE	43
CHAPTER 6.....	46
CONCLUSION AND FUTUURE WORK.....	46
6.1 CONCLUSION	47
6.1.1. Project features and limitations	47
6.2 FUTURE WORK	48
APPENDICES.....	50
Appendix – I: Analysis of Stock Market	51
Appendix – II: Prediction of Financial Risk and Early Warning Model	60
REFERENCES	62

LIST OF FIGURES

Figure 1 Sample from the data	25
Figure 2 Summary statistic for the EGAL company	27
Figure 3 Summary statistic for the price column to all the 5 companies	28
Figure 4 The highest prices at the companies	29
Figure 5 Closing Prices at each company	30
Figure 6 Largest Daily Return for each company	30
Figure 7 MFPC Daily Return	31
Figure 8 AXPB Daily Return	31
Figure 9 EGAL Daily Return	32
Figure 10 ABUK Daily Return	32
Figure 11 ESRS Daily Return	32
Figure 12 Expected Risk and Return for each company	33
Figure 13 Correlation Matrix of Companies	34
Figure 14 Cumulative Trading volume Over Time	34
Figure 15 EGAL Candlestick Chart with volume	35
Figure 16 ABUK Candlestick Chart with volume	35
Figure 17 ESRS Candlestick Chart with volume	36
Figure 18 AXPB Candlestick Chart with volume	36
Figure 19 MFPC Candlestick Chart with volume	37
Figure 20 Actual Val Vs. Predicted Val	41
Figure 21 Home Page	44
Figure 22 Analysis Page	44
Figure 23 Prediction page	45

CHAPTER 1

INTRODUCTION

1.1. Introduction

In recent years, the urgency of developing robust early warning systems for financial risk assessment has been underscored by the increasing frequency and severity of financial crises. The global financial meltdown of 2008 served as a stark reminder of the devastating consequences that unchecked systemic risks can inflict on economies, businesses, and individuals worldwide. This wake-up call has spurred renewed efforts among governments, regulatory bodies, and financial institutions to bolster their capacity for early detection and mitigation of financial vulnerabilities. The recognition that proactive intervention can help avert or minimize the impact of crises has catalyzed a collective drive to enhance the predictive accuracy and timeliness of early warning systems. By harnessing advances in data analytics, machine learning, and risk modeling techniques, stakeholders seek to fortify the resilience of financial markets and institutions.

The primary objective of this project is to empower investors and financial institutions with actionable insights to proactively manage risks and safeguard investments. By detecting early warning signals, users can make informed decisions, adjust investment strategies, and implement risk mitigation measures before significant losses occur. Additionally, the project aligns with regulatory requirements, promoting transparency and stability in financial markets.

Through a systematic approach encompassing data collection, model development, validation, and deployment, this project seeks to create a comprehensive early warning system that enhances market resilience and fosters confidence among investors. By leveraging cutting-edge technology and analytical tools, the "Early Warning" project aims to usher in a new era of proactive risk management, ultimately contributing to the stability and sustainability of global financial markets.

1.2. Problem domain

The stock market is a complex and highly dynamic environment where prices of stocks fluctuate based on numerous factors, including company performance, economic indicators, geopolitical events, and market sentiment. Accurate predictions of stock prices can lead to substantial financial gains for investors and institutions. However, due to the inherent volatility and the multitude of influencing factors, predicting stock prices with high accuracy remains a formidable challenge.

Investors and traders face significant difficulties in making informed investment decisions due to the unpredictable nature of stock price movements. Traditional methods of financial analysis, such as technical and fundamental analysis, often fail to capture the intricate patterns and rapid changes in the market. This results in suboptimal investment decisions, increased risk, and potential financial losses.

Moreover, the sheer volume of available data, including historical prices, trading volumes, financial statements, and news articles, presents an overwhelming challenge for individual investors and even institutional analysts to process and analyze effectively.

This project aims to address these challenges by developing an advanced predictive model for stock prices using state-of-the-art machine learning techniques. The model will integrate a wide range of data sources, apply sophisticated algorithms, and provide accurate, real-time predictions to assist investors in making data-driven decisions.

1.2.1. Challenges in Stock Market Prediction:

1. Market Volatility

- Stock markets are highly volatile, with prices fluctuating rapidly due to various factors.
- High volatility makes it difficult to predict short-term movements accurately.
- Political events, economic announcements, and natural disasters can cause sudden market shifts.

2. Non-linearity and Complexity

- Stock markets exhibit non-linear behavior and complex interactions among various factors.
- Traditional linear models often fail to capture these complexities, necessitating advanced machine learning techniques.
- The relationship between interest rates, inflation, and stock prices is complex and non-linear.

3. Noise and Irregularities in Data

- Stock market data contains a significant amount of noise and irregular patterns.
- Noise can obscure underlying trends and patterns, leading to inaccurate predictions.
- Market rumors and false information can cause temporary price changes unrelated to actual market conditions.

4. High Dimensionality

- Stock market prediction involves a large number of variables, including historical prices, trading volumes, and macroeconomic indicators.
- Managing and processing high-dimensional data requires sophisticated algorithms and computational resources.
- Incorporating global economic indicators and company-specific news increases the dimensionality of the dataset.

5. Overfitting

- Overfitting occurs when a predictive model learns the noise in the training data rather than the actual underlying patterns.
- Models that overfit the training data perform poorly on unseen test data, leading to unreliable predictions.
- A model that perfectly fits historical data may fail to generalize to future data.

6. Data Availability and Quality

- The availability and quality of historical data can vary, impacting the reliability of predictions.
- Incomplete or poor-quality data can lead to biased models and inaccurate forecasts.
- Missing data points or errors in historical records can skew the model's training process.

7. Model Interpretability

- Many advanced machine learning models, such as deep neural networks, are often considered "black boxes" with limited interpretability.
- Lack of interpretability makes it challenging to understand how predictions are made, which can be problematic for gaining trust from stakeholders.
- Financial analysts may be reluctant to use a model they cannot fully understand or explain.

8. Real time processing

- Predicting stock prices in real-time requires fast data processing and model updating capabilities.
- Real-time processing imposes technical challenges related to data latency and computational efficiency.
- High-frequency trading systems rely on real-time predictions to execute trades within milliseconds.

9. External Factors

- External factors such as geopolitical events, regulatory changes, and technological advancements can influence stock markets.
- Predicting the impact of these external factors is complex and often beyond the scope of traditional models.
- Unexpected regulatory changes can lead to sudden market reactions that are difficult to anticipate.

1.3. Problem statement

Stock market prediction involves forecasting the future values of financial markets based on historical data and various analytical techniques. It is a domain that combines finance, economics, data science, and machine learning. Accurate predictions can lead to significant financial gains, making this an attractive but complex area of study.

1.4. Objectives

The proposed model aims to predict the Financial Risk associated with stock investments. By Analyzing Historical data, this will help investors make informed decisions.

- Collecting data with high quality to train the model.
- Create robust machine learning models capable of predicting stock prices based on historical data.
- Extract relevant features from stock market data, considering factors like trading volume, price trends.
- Assess model performance using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and accuracy.

1.5. Resources required

1.5.1. Data resources

1. Historical Stock Market Data

- Historical prices, trading volumes, and other stock market-related data.
- Financial data providers like Yahoo Finance, Google Finance, Bloomberg, and investing.com.
- Access to comprehensive and up-to-date historical data for multiple stocks over a significant period.

2. Economic and Financial Indicators

- Data on interest rates, inflation rates, GDP growth, unemployment rates, etc.
- Government databases, central banks, World Bank, International Monetary Fund (IMF).
- Access to relevant economic indicators that can influence stock market movements.

3. Company-specific Information

- Financial statements, earnings reports, news articles, and press releases related to specific companies.

- Company websites, SEC filings, financial news websites, databases like EDGAR.
- Access to detailed and timely company-specific data.

1.5.2. Software resources

1. Data preprocessing and analysis tools
 - Python, R, SQL, Pandas, NumPy, Scikit-learn.
 - Libraries and frameworks for data cleaning, preprocessing, and exploratory data analysis.
 - Access to programming environments and necessary libraries.
2. Machine Learning Frameworks
 - TensorFlow, Keras, PyTorch, Scikit-learn.
 - Frameworks to develop and train machine learning models.
 - Installation and familiarity with machine learning frameworks.
3. Visualization Tools
 - Matplotlib, Seaborn, Plotly, Tableau. Tools for visualizing data and model predictions.
 - Access to visualization libraries and software for creating insightful charts and graphs.

1.5.3. Hardware resources

Computers with powerful CPUs and large RAM to handle data processing and model training. Sufficient computing power to process large datasets and train complex models efficiently.

1.6. Development methodology

- Data Collection: Gather historical stock data from reliable sources like investing.com for the companies AXPB, ABBK, ESRS, EGAL and MFPC.
- Data Preprocessing: Clean, normalize, handle missing values and descriptive statistics.
- Exploratory data analysis: Check for correlations between stock prices and other variables.
- Model Selection: Experiment with algorithm LSTM.
- Model Training and Tuning: Train models on historical data
- Model Evaluation: Validate predictions against data.
- Documentation: Maintain clear documentation throughout the project.

1.7. Project report

1.7.1. Chapter 1

This chapter introduces the project, which focuses on predicting stock market prices using machine learning techniques. It provides a comprehensive overview of the project's aim to leverage historical market data and economic indicators to enhance the accuracy of stock price forecasts. The chapter begins with the Problem Statement, highlighting the challenges of stock market volatility and the limitations of traditional forecasting methods. It emphasizes the need for sophisticated computational approaches to handle large data volumes and uncover meaningful patterns. The Objectives section outlines the primary goals of the project, including data collection and preprocessing, implementing and comparing various machine learning algorithms, optimizing models, and evaluating their performance. The Resources Required section details the necessary resources, such as data sources, computational tools, hardware, literature, and development environments, essential for completing the project. The Development Methodology section describes the structured approach followed, including phases like data collection and preprocessing, model implementation, model evaluation, analysis and interpretation, and documentation and reporting.

1.7.2. Chapter 2

This chapter provides a comprehensive review of the literature related to stock market prediction using machine learning. It aims to contextualize the current project within the broader research landscape by examining existing methodologies, findings, and technological advancements in the field. The review is structured to highlight significant studies, identify gaps in the literature, and justify the need for the current research.

1.7.3. Chapter 3

This chapter provided a comprehensive overview of the data used in the project, including the processes of data collection, preprocessing, and visualization. The steps taken to clean and prepare the data, as well as the techniques used to visualize it, were detailed. This foundational work ensures that the data is reliable, clean, and ready for model development and analysis in the subsequent chapters.

1.7.4. Chapter 4

This chapter details the process of building a predictive model for stock market prices using Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) particularly well-suited for time series data. It covers the implementation steps, the evaluation methodology, and the results of the model using Root Mean Squared Error (RMSE) as the primary evaluation metric. The results demonstrated the model's effectiveness and highlighted areas for potential improvement in future work.

1.7.5. Chapter 5

This chapter has summarized the project's development, highlighting key features and limitations of the stock market prediction website. It has outlined specific recommendations for future work, focusing on enhanced data integration, advanced model development, and user interface improvements. By addressing these areas, the project can continue to evolve, providing more accurate predictions and a better user experience.

1.7.6. Chapter 6

This chapter provides a comprehensive overview of the project, summarizing its key features and limitations. It discusses the accomplishments achieved through the development and evaluation of the stock market prediction model and highlights specific enhancements that were beyond the scope of the project but could be pursued in future work. The recommendations for future work are based on the findings and conclusions drawn from the extensive research and development carried out over the past year.

CHAPTER 2

BACKGROUND/EXISTING WORK

2.1. Stock Market Prediction Using Machine Learning (LSTM):

2.1.1. overview

In the paper “Using machine learning algorithms on prediction of stock price”, Li-PangChen investigated the performances of different machine learning methods that include LSTM, CNN and SVR in predicting the future stock price on four stocks chosen from Yahoo Finance. The datasets are extracted from the historical stock data chosen from Apple, Mastercard, Ford and ExxonMobil, these are time-depend data that has variables of open, close, daily high, daily low, adjusted close and volumes as explained in the part 3.1.1. Chen uses datasets that are much larger from the last one, it includes data from January 1st 2002 till March 11th 2020 to ensure the variations will allow the models to perform at their real value with no exceptions. Before building the model, Chen also carefully investigated the properties of the data that includes distribution, variability clustering, linear correlation and long-range dependence, which also helps greatly with the model-Ing part later on. It is always necessary to investigate and visualize the datasets before doing any practical modeling work because it will ensure the data is trainable and does not have any features that might influence the performance of the models and the accuracy of the results. Most importantly, the Mean Absolute Percentage Error (MAPE) issued to evaluate the performance and accuracy of each model.

2.1.2. Regression Based Model and LSTM on 9 Lakh Data Records

In the Paper “Stock Market Prediction Using Machine Learning” written by Ishita Parmaret al., supervised machine learning methods Regression-based model and LSTM are employed to forecast stock market prices using a dataset of 9 lakh records obtained from Yahoo Finance. The dataset represents the stock prices at different time intervals each day for one company and it includes variables of open, close, low, high and volume. As the variable names suggest, open, close, high and low are the direct stock bid prices at different times and volume means the total number of shares traded during a specific period of time. The dataset was used for simulation purpose only and thus only the data of one company was extracted. The dataset was divided into training and testing sets for the machine learning models to use and predict.

2.1.3. LSTM Model and Results

The Long Short-Term Memory is an advanced recurrent neural network (RNN) that is capable of learning long-term dependencies rather than only current and recent information. This is a wide-spread and effective method used in stock market prediction because an accurate prediction in the stock market heavily relies on the long-term history data of the

market and this is exactly what long-term short memory is designed for. LSTM controls its error by holding information of older stages that will make the results more accurate. In general, LSTM deals with the problem of vanishing gradient caused by the processing of large amount of data. Moreover, it contains a remembering cell that deals with long-term propagation very well. Parmer applied a model that stacked two LSTM layers with an output value of 256, they took care of the overfitting and efficiency problems by making 0.3 of the total nodes frozen during the training. The compiling process involves a mean square cost function to hold the error and accuracy is at stake during this process. This prediction is compared to the actual value and it measures the real trend of the stock market data when time passes. The resulting train score is 0.00106 MSE and test score is 0.00875 MSE.

2.1.4. conclusion:

Financial markets provide an essential platform for financial transactions and investments to happen and the it allows people to have the opportunity of making their investment grow. Therefore, stock market prediction is essential in helping the investors make correct and profitable decisions. The attempt of this paper is to give a systemic review of the modern machine learning methods that are commonly used in the stock market prediction process by analyzing two academic papers that includes the building, testing and analyzing of LSTM, Regression-based model, CNN and SVM. LSTM performed better than the Regression-based model when training on a relatively small dataset, SVM perform the best among SVM, CNN and LSTM in predicting future prices based on large stock datasets and the combination of CNN and LSTM also produces better results. It is recommended that people try to choose suitable data sets with different time spans and investigate the characteristics of the data before building a model.

2.2. Analysis and prediction of stock market trends using deep learning:

2.2.1. Overview

The paper proposes a progressive conclusion on the application of recurrent neural networks in stock price forecasting. We have also used random forest classifier to factor in the sudden fluctuations in stock prices which are derivatives of any abnormal events. Machine learning and deep learning strategies are being used by many quantitative hedge funds to increase their returns. Finance data belongs to time series data. A time series is a series of data points indexed in time. The nonlinearity and chaotic nature of the data can be combated using recurrent neural networks which are effective in tracing relationships between historical data and using it to predict new data.

Historical data in this context is time series data from the past. It is one of the most important and the most valuable parts for speculating about future prices. Long short-term memory (LSTM) is capable of capturing the most important features from time series data

and modelling its dependencies. Building a good and effective prediction system can help investors and traders to get a glimpse of the future direction of the stock and accordingly help them mitigate risk in their respective portfolios. The results obtained by our approach are accurate up to 97% for the values predicted using historic data and 67% for the trend prediction using news headlines.

2.2.2. Methods

1. Value prediction using historical data

We are taking opening price, closing price, highest price, lowest price and volume of shares traded of a specific stock on a particular day to predict the next day's stock prices. The output will be received in the form of the next day's open, close, high and low prices of a particular stock. LSTM layers are being used with timesteps of 120 days. Four hidden layers are being used. Each layer contains 75 nodes where edges are initialized with random weights. The output layer contains four nodes which give us our four predicted values. 'Adam' optimizer (Its function is to update the weights of the neural network during training) is used for minimization of the loss function. It is better than the rest of the adaptive technique, and it rectifies every problem that is faced in other optimization techniques such as vanishing learning rate, slow convergence or high variance in the parameter updates which leads to fluctuating loss function. After exploring other loss function, 'mean squared error' is found to be the most appropriate loss function for this system. Dense layer connection will be used with a dropout rate of 20% to avoid overfitting.

2. Trend Prediction Using News Headlines

This module will use sentiment analysis on news headlines for the stock whose next day's value is to be predicted. The news headlines are extracted from the web and the data is pre-processed and cleaned. After that, each headline is classified as positive, negative and neutral. Then, the feature extracting is done using N-Gram and analyte data is trained using random forest classifier as. The input for this module is the news headlines of the stock and labels to predictive the value of the stock will increase or decrease on the next day. The label for the current day is calculated by subtracting the open value of current day from the close value of the previous day, if the resulting value is negative then the label will be '1' and if the resulting value is positive or zero then the label will be '0'.

2.2.3. Conclusion and future work

The proposed system is accurately following the pattern of the stock prices where the predicted values are in close vicinity to the actual market values. We have used the data set of SBIN stock and achieved an accuracy of 97%, but the sudden fluctuations in the price of many stocks which are derived from abnormal events cannot be factored in the prediction system which is trained on past values of the respective stock prices. For such anomalies, sentimental analysis of the event is required.

We have implemented the system by analyzing the news which is available for any specific stock to reduce the deviation of the predicted prices from the actual prices. After adjusting the system and factoring in the sentimental analysis, we have achieved an accuracy of 67.39%. RNN is robust in modelling complex relationships between parameters but it is not good enough to find its application in live trading system. For future work, adding more dimensionality to the data set can further improve the accuracy of prediction. Global trends and events can be used to further improve the prediction of the stock's direction for a longer timeline.

2.3. Prediction and Analysis of Corporate Financial Risk Assessment Using Logistic Regression Algorithm in Multiple Uncertainty Environment:

2.3.1. introduction:

Under the environment of economic improvement and market opening, listed companies are developing rapidly and have broad prospects, but they are also under heavy pressure and fierce competition [1]. Enterprises are the most basic element of social production and the main force to promote economic improvement. Whether their improvement is healthy or not is not only related to their own destiny but also related to the national economy and national peace and security. The assumption of sustainable operation in classical finance theory is being relaxed in the face of the pressure from growing market competitiveness in the information economy, and the uncertainty faced by businesses is growing. Due to financial crises, it is not uncommon for businesses to experience problems or even file for bankruptcy. Therefore, at any point in its business process, every organization must take into account early warning of financial crisis and collapse. It should begin to act as soon as anomalous indicators are discovered in order to prevent or lessen the harm to the business. The low level of company financial management forms a sharp contrast with the high requirements of companies for financial management. It is an urgent task to establish a scientific financial management mode and improve the level financial management. At the same time, the operating status of the company will ultimately be reflected by financial indicators. Therefore, improving the financial management level can directly improve the overall management level of companies. It can be predicted that in the near future, most companies will gradually enter the stage financial oriented corporate governance. In this economic environment, company financial management will become the core issue of corporate governance.

2.3.2. Methodology

The prediction of financial risk management is examined in this work using the logistic regression model. Particularly in the disciplines of medicine, social research, processing biological data, and other areas, logistic regression analysis is frequently utilized as an efficient data processing technique. The choice of the regression model, cost function, and

likelihood function in logistic regression analysis is frequently correlated with a particular probability distribution or probability model. The original data can be suitably altered during data processing, with the aim of as closely aligning the model with the real dose-response relationships possible. In addition, the covariate distribution's skewness and kurtosis coefficients are not very high to guarantee the stability of convergence. Model construction requires a high level of expertise, particularly cumulative model construction. The right model must be chosen and reasonably built in accordance with the actual questions and the unique conditions of the data in the application, all while having a thorough understanding of the theoretical background, characteristics, and probability assumptions of the model. Regression analysis must also use the effective cost function or likelihood functions ensure a reasonable regression effect.

2.3.3. Results

Generally, the so-called early warning system is a system that informs a person, an organization, or even a country in advance of the dangerous situation, and should pay attention to the genes that may cause a crisis and take preventive measures in advance. Differentially warning indicators have different priorities and different prediction conclusions. Therefore, financial early warning indicators should be effectively classified according to the nature of the field to be monitored, and important indicators that can best reflect the characteristics of financial activities should be listed separately. Only by ensuring that the financial personnel can grasp the information in the first time can the overall timeliness of financial risk early warning be ensured, so that there can be enough time to evaluate the risk and select the best treatment method. Enterprise financial risk early warning system is a system engineering and one of the subsystems of management early warning system. The financial risk early warning system needs to establish the corresponding early warning organization system and early warning analysis methods to help realize the management functions of decision-making, organization, and control and to realize the self-balance and self-improvement in the business process of companies.

2.3.4. Conclusion

Financial hazards are a constant concern for businesses. An extreme example of financial risk is a financial crisis. However, neither the occurrence nor the cause of a financial crisis happens overnight. It emerges gradually and keeps building up over time. Financial crisis arises when a company's financial risk reaches a specific level of accumulation. Any corporation may face financial risks in its financial management activities given the intense market rivalry.

The company will suffer financial losses or financial difficulties or possibly a financial crisis if it is unable to identify and effectively address the current and potential risks in a timely manner. This will start a domino effect that will result in unknown disasters for creditors, stockholders, and other stakeholders. The degree of financial risk for the organization can be quickly understood if it is assessed and handled quantitatively, allowing the company managers to make informed decisions. These needs can be satisfied

by the financial early warning system. This method is suited for widespread usage in practice because it has shown exceptional outcomes in study that are 16.24% better than those of the conventional method.

2.4. Prediction model for stock market analysis:

2.4.1. project overview

Predicting the Stock Market has been the bane and goal of investors since its existence. Everyday billions of dollars are traded on the exchange, and behind each dollar is an investor hoping to profit in one way or another. Entire companies rise and fall daily based on the behavior of the market. Should an investor be able to accurately predict market movements, it offers a tantalizing promise of wealth and influence. It is no wonder then that the Stock Market and its associated challenges find their way into the public imagination every time it misbehaves. The 2008 financial crisis was no different, as evidenced by the flood of films and documentaries based on the crash. If there was a common theme among those productions, it was that few people knew how the market worked or reacted. Perhaps a better understanding of stock market prediction might help in the case of similar events in the future.

Despite its prevalence, Stock Market prediction remains a secretive and empirical art. Few people, if any, are willing to share what successful strategies they have. A chief goal of this project is to add to the academic understanding of stock market prediction. The hope is that with a greater understanding of how the market moves, investors will be better equipped to prevent another financial crisis. The project will evaluate some existing strategies from a rigorous scientific perspective and then move onto its objective of providing a mathematical model for prediction.

The idea at the base of this project is to build a model to predict financial market's movements. The forecasting algorithm aims to foresee whether tomorrow's exchange closing price is going to be lower or higher with respect to today. Next step will be to develop a trading strategy on top of that, based on our predictions, and back-test it against a benchmark.

2.4.2. Algorithms used

1. Logistic regression

Logistic regression is an alternative method to use other than the simpler Linear Regression. Linear regression tries to predict the data by finding a linear – straight line – equation to model or predict future data points. Logistic regression does not look at the relationship between the two variables as a straight line. Instead, Logistic regression uses the natural logarithm function to find the relationship between the variables and uses test data to find the coefficients. Logistic regression uses the concept of odds ratios to calculate the

probability. This is defined as the ratio of the odds of an event happening to its not happening.

2. Random Forest

Random Forest is a statistical algorithm that is used to cluster points of data in functional groups. When the data set is large and/or there are many variables it becomes difficult to cluster the data because not all variables can be taken into account, therefore the algorithm can also give a certain chance that a data point belongs in a certain group. This is how the clustering takes place.

3. Support Vector Machine

the dataset (not the training set) is used to predict which tree in the forests makes the A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVMs are more commonly used in classification problems. Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set. Hyperplane is a line that linearly separates and classifies a set of data. Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplanes possible, while still being on the correct side of it. So when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it. The distance between the hyperplane and the nearest data point from either set is known as the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly.

CHAPTER 3

DATA Analysis

3.1. Data collection:

Data Collection is one of the most important steps in creating an accurate machine learning model. This involves gathering historical data on stock market of Egyptian companies from (investing.com) like prices, volumes.

We collected the data of 5 companies that are:

ABUK (Abu Qir fertilizers), MFPC (Misr fertilizers production co), EGAL (Egypt aluminum), AXPB (Alexandria pharmaceuticals), ESRS (Ezz steel)

Here's a sample from the data

Date	Price	Open	High	Low	Vol#	Change %
4/24/2019	123.01	128.9	128.9	122	320	-0.0461
5/2/2019	123.01	123.1	123.1	123.1	10	0
5/5/2019	123.01	127.5	129	127.5	100	0
5/7/2019	122.16	123.11	123.11	122.15	13250	-0.0069
5/12/2019	122	122	122	122	1210	-0.0013
5/13/2019	122	121	121	121	200	0
5/14/2019	122	120.01	120.01	116	50	0
5/16/2019	110.02	110.01	111	110	1850	-0.0982
5/19/2019	110.66	110.02	112	106.2	380	0.0058
5/20/2019	110.66	113	113	111	150	0
5/21/2019	110	110	110.01	110	1100	-0.006
5/22/2019	110.01	110	110.05	110	720	0.0001
5/23/2019	111.02	111.1	111.1	111	1030	0.0092
5/27/2019	111	111	111	111	540	-0.0002
5/30/2019	111.16	111.25	111.25	111.15	540	0.0014
6/9/2019	114.11	111.31	121.96	110	11030	0.0265
6/10/2019	111.92	112.52	112.52	111.02	2950	-0.0192
6/11/2019	111.5	111.01	112	111	6470	-0.0038
6/12/2019	108.49	114	114	101.02	2360	-0.027
6/13/2019	105.21	108	108	104.99	8280	-0.0302

Figure 1 Sample from the data

After collecting data, we need to understand what data means and here's an explanation

Price: A stock's price indicates its current value to buyers and sellers.

Open: It is the price at which the financial security opens in the market when trading begins.

High: Today's high refers to a security's intraday highest trading price. It is represented by the highest point on a day's stock chart.

Low: Today's low is the lowest price at which a stock trades over the course of a trading day.

Volume: A stock's volume is the number of shares traded in a given period.

Change: If the closing price is higher than the opening price, the Day's Change is positive, indicating a gain in the stock's value. **Negative Change:** Conversely, if the closing price is lower than the opening price, the Day's Change is negative, signifying a loss in the stock's value.

3.2. Data preprocessing:

Data preprocessing is a crucial stage in the development of efficient machine learning models.

And there are several important techniques utilized for data preprocessing in financial risk assessment model:

3.2.1. Data Cleaning

The first stage in data pre-processing is cleansing the data. This process identifies and corrects errors, duplicate, and missing values in the dataset.

- **Dealing with Duplicates:** Identify and remove duplicate records.
- **Imputing missing values** is important for ensuring that the machine learning model can use all the data available. Missing values can occur in the datasets due to data entry errors or because some information is not recorded. To impute these missing values, we can use mean, median, or mode imputation.
- **Outliers** are data points that deviate significantly from the rest of the data. They may occur in the datasets due to errors in data collection. To handle outliers, statistical techniques like z-score analysis can be employed to identify these anomalous data points. Once identified, they can be removed from the dataset.

3.3. Exploratory data analysis (EDA):

Exploratory data analysis (EDA) is an important step as it allows us to gain insight into the data that allows us to understand our data and can be used to make predictions.

For example, we made a summary statistic for the EGAL company

	<i>price</i>	<i>open</i>	<i>high</i>	<i>low</i>	<i>vol</i>
Mean	27.57	27.62	28.38	26.95	297687.12
Standard Error	0.68	0.68	0.71	0.66	11793.36
Median	17.89	17.86	18.58	17.51	163740.00
Mode	13.98	12.30	14.20	13.00	1270000.00
Standard Deviation	23.60	23.63	24.61	22.94	410910.33
Sample Variance	557.13	558.41	605.54	526.09	168847295405.04
Kurtosis	4.73	4.72	4.99	4.59	22.06
Skewness	2.20	2.20	2.25	2.17	3.94
Range	122.72	123.41	128.34	117.24	4147500.00
Minimum	6.98	6.29	7.66	6.29	2500.00
Maximum	129.70	129.70	136.00	123.53	4150000.00
Sum	33471.47	33534.81	34448.44	32720.63	361392160.00
Count	1214.00	1214.00	1214.00	1214.00	1214.00

Figure 2 Summary statistic for the EGAL company

From this statistic we can conclude these observations:

1. Mean and Median Values:
 - The mean values for the price, open, high, and low columns are higher than the median values, indicating a positive skew in the data distribution.
 - The mean volume is significantly higher than the median volume, suggesting the presence of outliers with very high values.
2. Standard Deviation and Variance:
 - High standard deviations and variances for all columns indicate substantial variability within the data sets.
 - Volume has an exceptionally high variance and standard deviation, further supporting the presence of significant outliers.
3. Mode Values:
 - The mode for price, open, high, and low is relatively lower than the mean and median, indicating that the most frequently occurring values are on the lower end of the spectrum.
4. Range and Extremes:
 - There is a wide range in the data, especially for volume, which has the largest range (4,147,500).

And another summary statistic for the price column to all the 5 companies

	MFPC	AXPH	EGAL	ABUK	ESRS
count	1306.000000	1306.000000	1306.000000	1306.000000	1306.000000
mean	16.270658	132.786233	27.490084	31.582129	22.248185
std	17.435374	30.332721	23.570095	19.883627	21.469735
min	3.030000	89.730000	6.980000	11.710000	4.440000
25%	7.050000	107.710000	12.930000	19.870000	9.905000
50%	9.810000	134.080000	17.740000	22.285000	12.955000
75%	16.072500	140.130000	32.917500	40.810000	24.890000
max	88.510000	246.920000	129.700000	106.700000	92.000000

Figure 3 Summary statistic for the price column to all the 5 companies

From these statistics we can obtain important information about the price for each company

- **MFPC:** The data ranges widely from 3.03 to 88.51 with a mean around 16.27, indicating a right-skewed distribution.
The standard deviation is quite high, suggesting significant variability.
- **AXPH:** This column shows values mostly clustered around a mean of 132.79, with values ranging from 89.73 to 246.92.
The data shows moderate variability with a standard deviation of 30.33.
- **EGAL:** The values range from 6.98 to 129.70 with a mean of 27.49. The distribution likely has a long tail on the higher end.
The high standard deviation of 23.57 indicates considerable spread in the data.
- **ABUK:** The data points range from 11.71 to 106.70 with a mean around 31.58. This suggests the distribution is spread out.
A standard deviation of 19.88 also reflects significant variability.
- **ESRS:** The values range from 4.44 to 92.00 with a mean of 22.25. The distribution is likely right-skewed.
A standard deviation of 21.47 indicates considerable spread.

3.4. Data visualization

Data visualization helps us to display the data in meaningful insights that make sense and can be studied and analyzed easily, and it provides a better way to compare between values, which will support better decision-making, and here are our important visualizations we make out in the project:

This is a histogram to visualize the highest price at the companies over the past 5 years, from this graph we find that AXPB Company have the highest price over the years and ESRS company has the lowest.



Figure 4 The highest prices at the companies

And this is a line chart for each company to see in details how the prices change over the years, here we can see that AXPB with the highest prices have a significant changes over the years, from 2020 to 2022 the price start to increase but after this period we can see that the prices dropped down again and then it has the highest prices by the year 2024.



Figure 5 Closing Prices at each company

A histogram to see the maximum daily return for each company and we see that MFPC has the highest, ESRS and AXPB equally have the lowest:

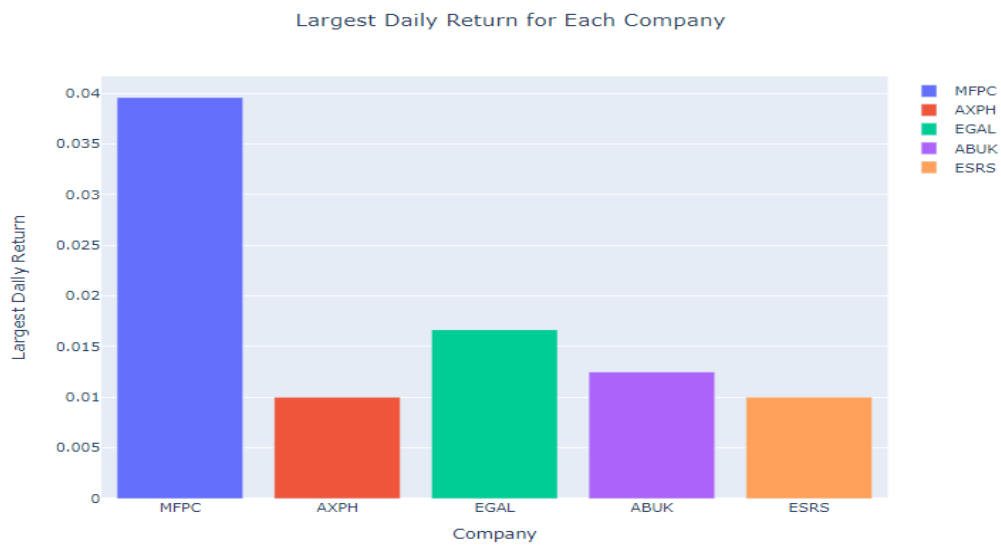


Figure 6 Largest Daily Return for each company

-And now we can see the change of the daily return over the years for each company

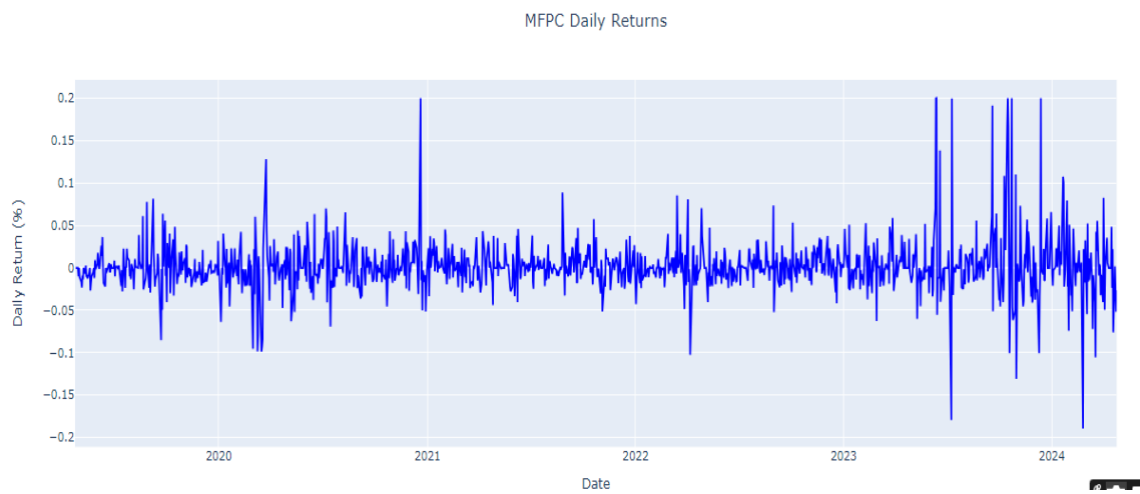


Figure 7 MFPC Daily Return

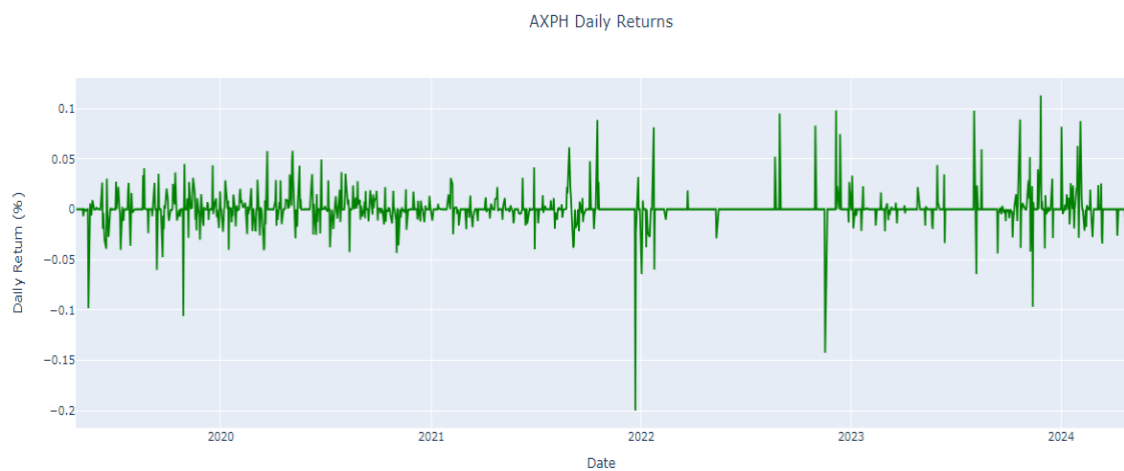


Figure 8 AXPH Daily Return



Figure 9 EGAL Daily Return

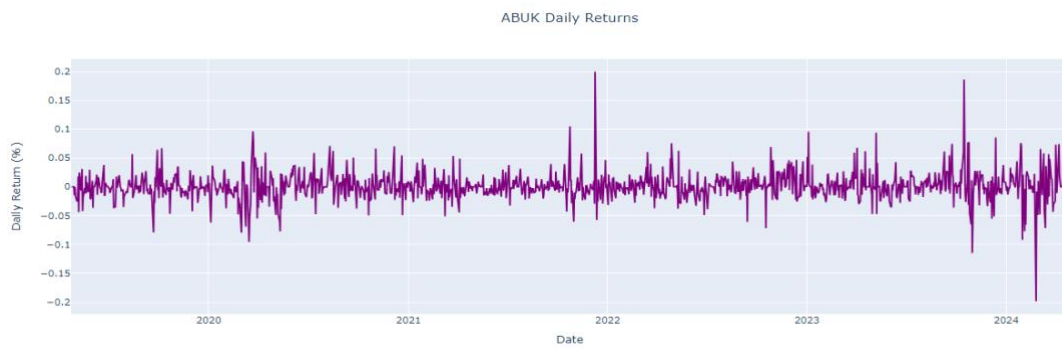


Figure 10 ABUK Daily Return



Figure 11 ESRS Daily Return

Then to choose the most appropriate company to build our model and train the data, we need to calculate the expected risk and return for each company and start to compare:



Figure 12 Expected Risk and Return for each company

Risk: It calculates the standard deviation (std) of daily returns for each company, representing the volatility or risk.

Expected Return: It computes the mean of daily returns for each company, indicating the average return expected from the stock.

From this graph we see that AXPB has the lowest risk and return but EGAL has the highest risk and return, so we choose to build our model for EGAL company to make an early warning prediction.

Then Calculate the correlation matrix to determine the relationship between the stock prices of different companies to know how the stock prices of various companies move in relation to each other.

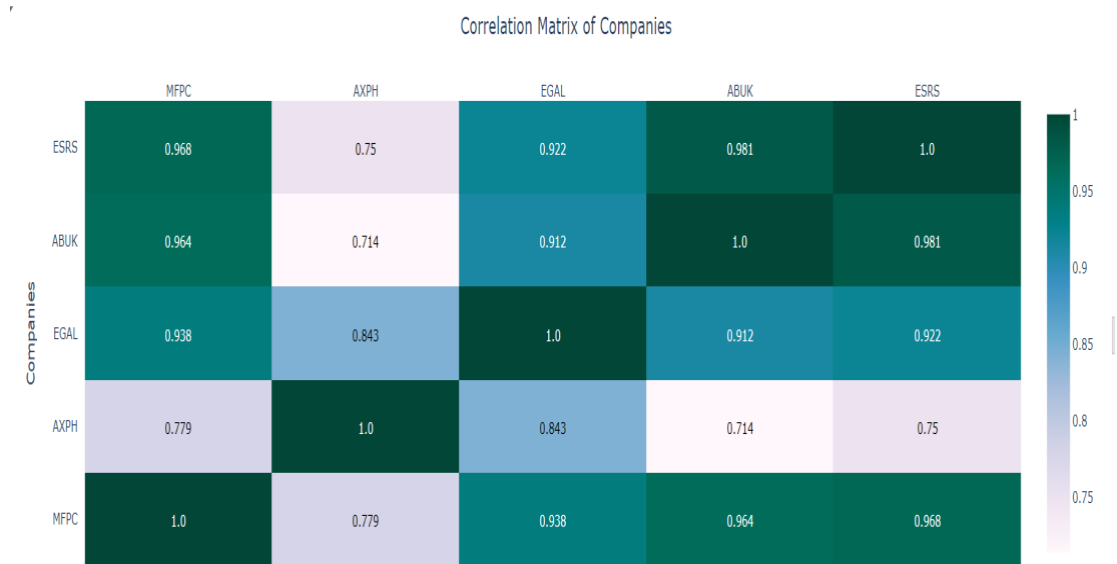


Figure 13 Correlation Matrix of Companies

To analyze the trading volume trends of different companies over time by calculating and visualizing their cumulative trading volumes.

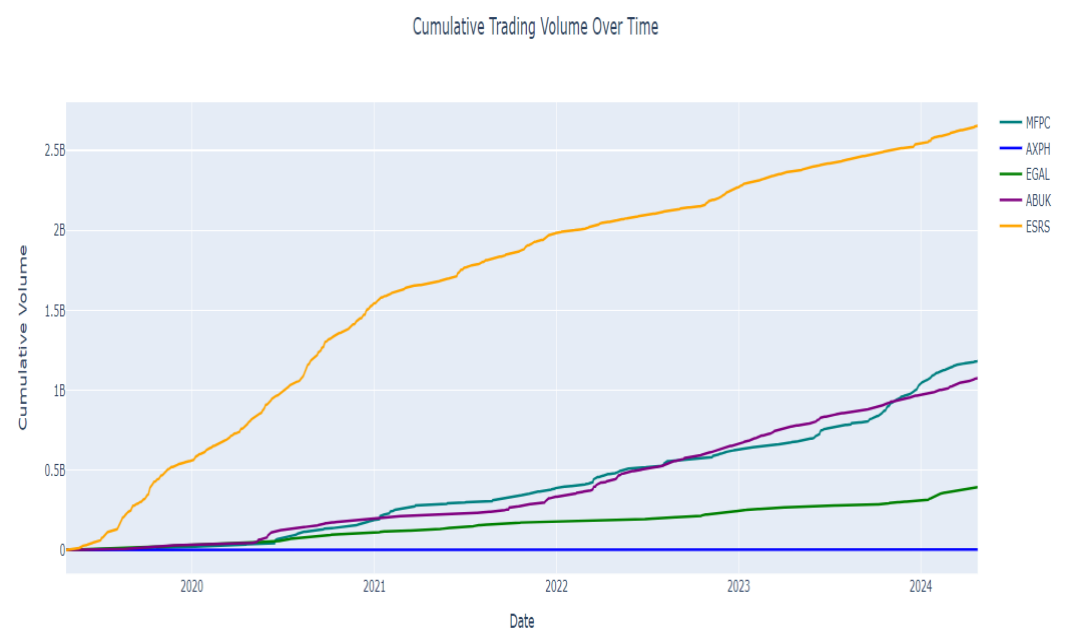


Figure 14 Cumulative Trading volume Over Time

So, we can see which companies were traded more frequently.

- MFPC and ESRS had higher trading volumes at different times.
- EGAL had the lowest trading volume.
- AXPB and ABUG had moderate trading volumes with some fluctuations.

Then visualize the stock price movements of a financial instrument and trading volume for EGAL using a candlestick chart and a volume bar chart.



Figure 15 EGAL Candlestick Chart with volume

The Candlestick in ABUG:



Figure 16 ABUK Candlestick Chart with volume

The Candlestick in ESRS:



Figure 17 ESRS Candlestick Chart with volume

The Candlestick in AXPB:



Figure 18 AXPB Candlestick Chart with volume

The Candlestick in MFPC:



Figure 19 MFPC Candlestick Chart with volume

☐ **Price Movements:**

- **Green Candlesticks:** These indicate days when ABUK's stock price increased, it means that the stock price significantly rose that day.
- **Red Candlesticks:** These indicate days when ABUK's stock price decreased. A long red candlestick means a significant drop in price.

☐ **Trading Volume:**

- **High Volume:** Days with high trading volumes are represented by taller bars. This usually coincides with significant price movements. For example, a tall volume bar on a day with a long candlestick (green or red) suggests that many shares were traded and there was strong market activity.
- **Low Volume:** Shorter bars indicate lower trading volumes, suggesting less trading activity.

☐ **Combined Analysis:**

- **Price Increase with High Volume:** When you see green candlesticks with high volume bars, it indicates strong buying interest and investor confidence in ABUK.
- **Price Decrease with High Volume:** Red candlesticks with high volume suggest strong selling pressure, possibly due to negative news or investor concerns.
- **Price Movements with Low Volume:** When significant price changes are accompanied by low trading volume, it may indicate less conviction in the price movement, as fewer investors are participating.

CHAPTER 4

METHODOLOGY

The stock price prediction model utilizes a Long Short-Term Memory (LSTM) neural network to forecast future stock prices based on historical data. The process begins with collecting and preprocessing historical stock prices, ensuring the data is clean and scaled appropriately. A sequence of past stock prices is then generated to serve as input for the LSTM model. The model is designed with multiple LSTM layers, followed by dense layers to output the predicted stock price. After training the model on a portion of the data, predictions are made on the test set, and the model's performance is evaluated using metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The final step involves visualizing the predicted stock prices against the actual prices to assess the model's accuracy. This approach demonstrates the potential of LSTM networks in capturing temporal dependencies in stock price data, leading to more accurate predictions.

4.1. Algorithm used

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to overcome the vanishing gradient problem in traditional RNNs, allowing them to capture long-term dependencies more effectively. Here are some key points about LSTMs:

- **Structure:** LSTMs contain memory cells that maintain a cell state and three gates: input gate, forget gate, and output gate. These gates regulate the flow of information into and out of the memory cell, enabling LSTMs to retain information over long sequences.
- **Long-term Dependencies:** LSTMs are capable of learning and remembering over long sequences, which is beneficial for tasks requiring context over time, like speech recognition and language translation.
- **Applications:**
 - Time Series Prediction: Predicting stock prices, weather forecasting, etc.
 - Natural Language Processing: Language translation, sentiment analysis, text generation.
 - Speech Recognition: Converting speech to text and vice versa.
- **Training:** LSTMs are trained using backpropagation through time (BPTT), where gradients are computed across all time steps to adjust the model parameters.

4.2. The implementation

We built a machine learning model that predicts stock prices based on historical data. The LSTM model is particularly suited for this task due to its ability to capture temporal dependencies. Data preprocessing, model training, prediction, evaluation, and visualization are all crucial steps to ensure that the model performs well and provides meaningful predictions.

1. Start by importing the necessary libraries for data manipulation, visualization, and machine learning.

2. Load and Prepare Data, Extract the 'Price' column and convert the Data Frame to a NumPy array, we focus on the 'Price' column because it is the target variable we want to predict. Converting the Data Frame to a NumPy array facilitates numerical computations.
3. Split Data into Training and Testing Sets, we split the data to ensure that we have a separate set for training the model and for testing its performance. Using 80% of the data for training and 20% for testing is a common practice.
4. Normalize the data to ensure all values are between 0 and 1. Scaling the data helps improve the performance and convergence speed of the neural network by ensuring that all input values are within a similar range.
5. Generate sequences of past stock prices to use as input features for the LSTM model. LSTM networks are designed to capture temporal dependencies in sequential data. By creating sequences of past prices, we allow the model to learn patterns and dependencies over time. The LSTM layers are used to capture the temporal patterns in the stock prices. Dense layers are added to map the LSTM outputs to the final prediction. The model is compiled with the Adam optimizer and mean squared error loss function for regression.
6. Fit the model on the training data. Training the model involves feeding the training data to the LSTM network and adjusting the weights to minimize the loss function. The number of epochs and batch size can be tuned based on model performance and computational resources.
7. Prepare Test Data. To make predictions, the test data must be prepared in the same way as the training data. This ensures consistency and allows the model to use the same input structure it was trained on. Use the trained model to make predictions on the test data and inverse transform the scaled predictions.
8. Visualizing the results helps to intuitively understand how well the model is performing. The plot shows the actual vs. predicted prices, making it easy to see any deviations.

4.3. The results

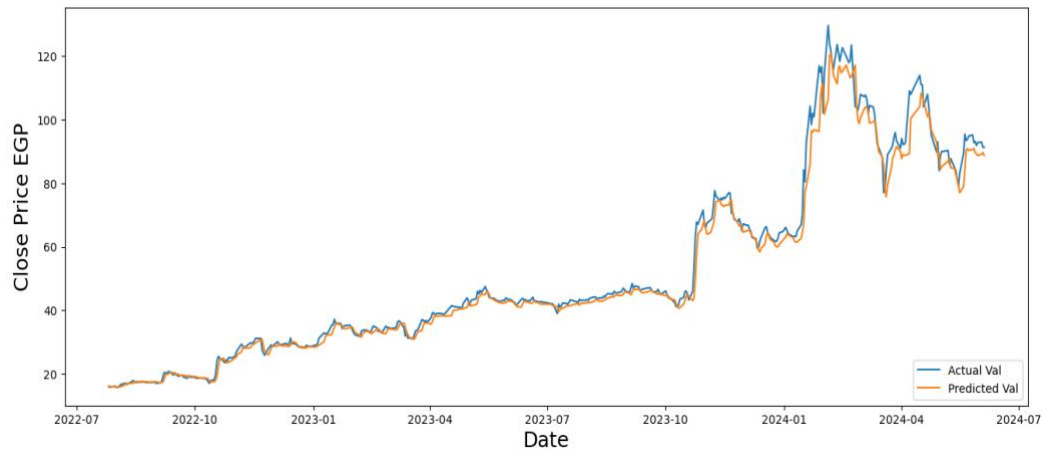


Figure 20 Actual Val Vs. Predicted Val

- The predicted stock prices closely follow the actual stock prices over the entire time period. Both the actual and predicted values capture the general upward and downward trends, indicating that the model is effective at understanding the overall market movements. The model demonstrates high accuracy during periods of relatively stable stock prices (from July 2022 to November 2023). During more volatile periods, such as from January 2024 to March 2024, the predictions still track the overall trend but exhibit some deviations from the actual prices.
- There are slight lags in the model's predictions compared to actual values, particularly noticeable during rapid changes in stock prices. This lag is typical in time series predictions and indicates that the model requires a few time steps to adjust to sudden changes
- The model performs well during periods of gradual changes and moderate volatility. During extreme volatility, the model still captures the direction of price movements but shows some discrepancies in the magnitude. The model appears to handle seasonal effects and periodic fluctuations in stock prices, indicating that it effectively captures underlying temporal patterns.

4.4. Model Evaluation

- The Root Mean Squared Error (RMSE) serves as a vital metric for evaluating the predictive accuracy of our Long Short-Term Memory (LSTM) model. RMSE measures the average magnitude of errors between the predicted stock prices and the actual stock prices. A lower RMSE indicates that our model's predictions are closely aligned with the actual market movements, which is essential for accurate risk assessment and early warning signals in financial markets.
- The RMSE is calculated in several steps. First, we create the testing dataset by scaling the stock price data and then segmenting it into sequences that the LSTM model can process. Specifically, we use the last 60 days of historical stock prices to predict the next day's price. These sequences are used to generate predicted prices, which are then compared to the actual stock prices. The squared differences between the predicted and actual prices are averaged to compute the Mean Squared Error (MSE). Taking the square root of the MSE gives us the RMSE, which translates the error metric back to the original units of stock prices, making it more interpretable.
- In our financial risk assessment project, a lower RMSE signifies that our model can accurately predict stock price movements, which is crucial for issuing reliable early warning signals about potential risks. Accurate predictions allow stakeholders to make informed decisions to mitigate financial risks. Conversely, a higher RMSE would suggest larger prediction errors, indicating the need for model improvements or additional data refinement. Presenting the RMSE value underscores the model's effectiveness in forecasting stock prices and its reliability in providing early warnings for financial risk management.
- By obtaining the RMSE value, we gain valuable insight into the model's overall performance. It helps identify how well the model generalizes to new, unseen data. If the RMSE is low, it indicates that the model performs well not just on training data but also on real-world data, enhancing its credibility for practical applications in financial risk management.
- An RMSE value of approximately 3.582 indicates the average magnitude of prediction error in our model, requiring further context-specific evaluation to determine its suitability for financial risk assessment applications.

CHAPTER 5

USER INTERFACE

Development of a website with a user-friendly interface that includes three main pages: Home, Analysis, and Prediction.

First the home page: it consists of the companies in Egypt with their history and when it was founded, and a search bar to write the name of the company you need and find it easily.

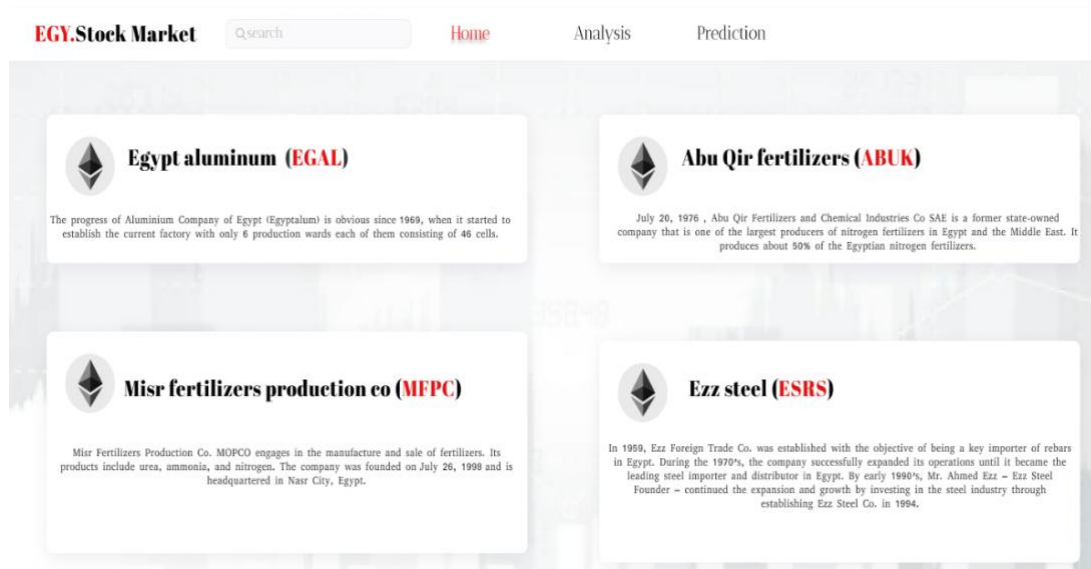


Figure 21 Home Page

Analysis page: this page to see the insights and analysis of the company you are searching for.

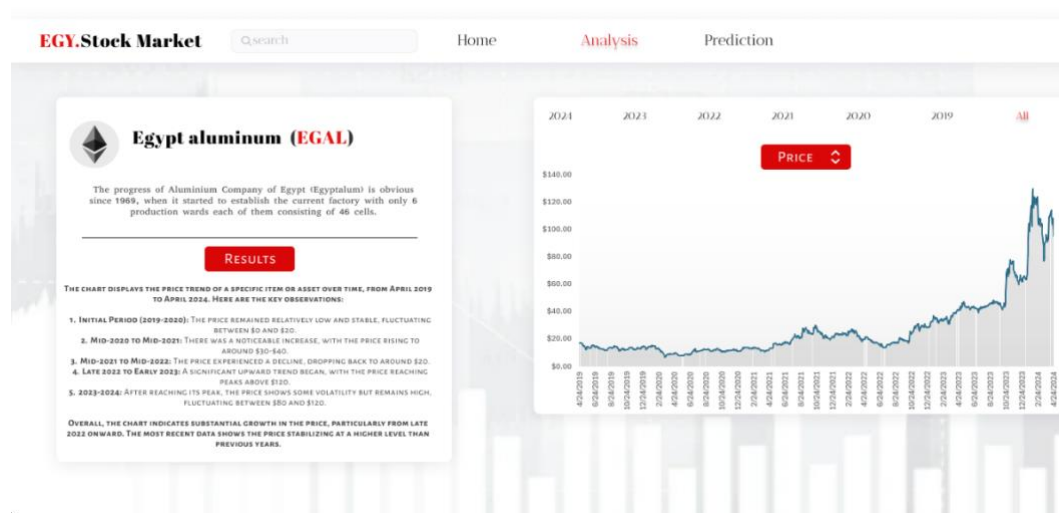


Figure 22 Analysis Page

Prediction page: to see the prediction results for the company.

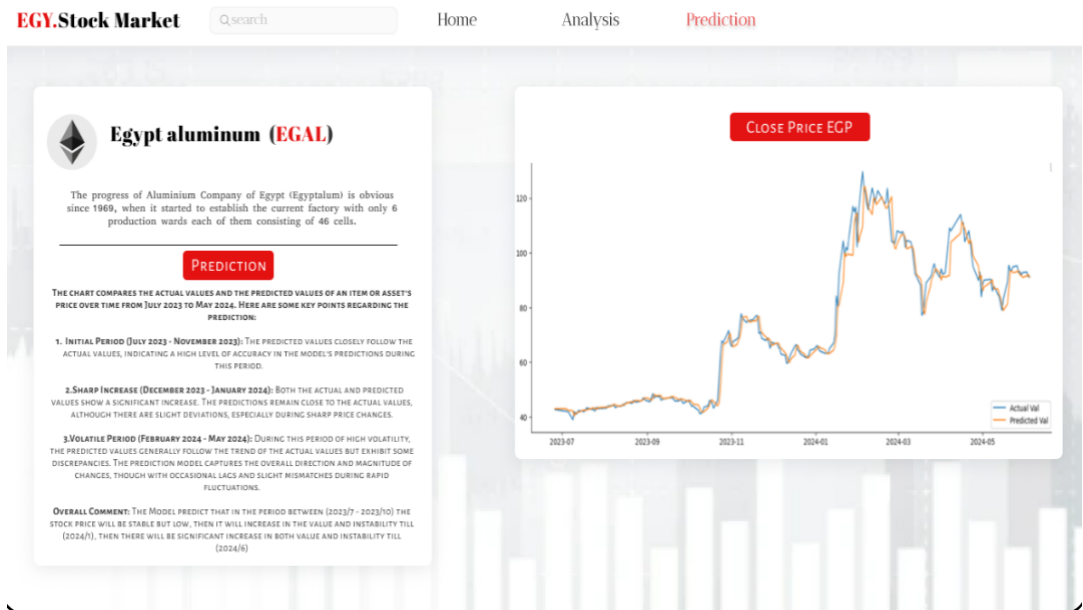


Figure 23 Prediction page

CHAPTER 6

CONCLUSION AND FUTUURE WORK

6.1 CONCLUSION

we developed a financial risk assessment early warning system using machine learning techniques to analyze stock data from five Egyptian companies. By leveraging historical stock prices and volumes, we trained an LSTM model to predict future stock prices and assess potential financial risks. Our analysis included visualizing price trends, trading volumes, and calculating key metrics such as daily returns and correlation matrices.

The model's performance was evaluated using the Root Mean Squared Error (RMSE), with a resulting value of approximately 3.582. This value indicates the average prediction error, suggesting that while the model demonstrates a reasonable level of accuracy, there is room for improvement. Further refinements and context-specific evaluations are necessary to enhance the model's predictive capabilities and ensure its reliability for financial risk assessments.

6.1.1. Project features and limitations

FEATURES

1. **Data Integration:** The project successfully integrated a variety of data sources, including historical stock prices, trading volumes, and economic indicators, to create a robust dataset for model training and evaluation.
2. **Implement model:** We implemented machine learning algorithm for stock price prediction.
3. **Predictive Accuracy:** The final models demonstrated the ability to capture market trends and provide reasonably accurate stock price forecasts, showcasing the potential of machine learning in financial predictions.
4. **Model evaluation:** By obtaining the RMSE value, we gain valuable insight into the model's overall performance. It helps identify how well the model generalizes to new, unseen data. If the RMSE is low, it indicates that the model performs well not just on training data but also on real-world data, enhancing its credibility for practical applications in financial risk management.

LIMITATIONS

1. **Data Quality and Availability:** The accuracy of the models was constrained by the quality and completeness of the available data. Missing values and inconsistencies in the data posed challenges during preprocessing.
2. **Market Volatility:** The models sometimes struggled to predict sudden market changes and high volatility periods, which are inherently difficult to forecast.
3. **Feature Selection:** While various features were engineered and included in the models, the selection process could be further refined to enhance predictive performance.
4. **Computational Resources:** Training complex models, especially neural networks, required significant computational power, which limited the extent of hyperparameter tuning and model experimentation.

6.2 FUTURE WORK

- **Enhanced Data Collection and Integration**

Future work could involve integrating additional data sources, such as social media sentiment analysis, news headlines, and global economic indicators. This could provide a more comprehensive view of the factors influencing stock prices and potentially improve model accuracy.

- **Advanced Model Architectures**

Exploring more sophisticated model architectures, such as ensemble methods and hybrid models that combine the strengths of different algorithms, could lead to better predictive performance. For instance, integrating Long Short-Term Memory (LSTM) networks with traditional machine learning models might capture both short-term fluctuations and long-term trends more effectively.

- **Robustness to Market Conditions**

Future work could focus on improving the models' robustness to different market conditions, such as periods of high volatility or economic downturns. This might involve training the models on various market scenarios and incorporating stress-testing techniques to evaluate performance under extreme conditions.

- **Real-time Prediction and Adaptation**

Developing a real-time prediction system that continuously updates and adapts to new data could enhance the practical applicability of the models. Implementing a feedback loop where the model learns from its prediction errors and adjusts its parameters accordingly could further improve accuracy.

- **User Interface and Usability**

Developing a user-friendly interface like (a website) that allows investors to interact with the predictive models and customize their inputs could increase the project's practical value. Visualization tools that display model predictions alongside historical data and other relevant metrics could aid in decision-making.

- **Ethical and Regulatory Considerations**

Considering ethical and regulatory implications of using machine learning in stock market predictions is crucial. Future enhancements could include implementing measures to ensure model transparency, fairness, and compliance with financial regulations.

This project laid a solid foundation for stock market prediction using machine learning, demonstrating the potential and challenges of this approach. By addressing the limitations and building on the identified future work areas, more accurate and reliable predictive systems can be developed, ultimately contributing to more informed and strategic investment decisions.

APPENDICES

Appendix – I: Analysis of Stock Market

✓ Import Libraries

```
[2] import pandas as pd
import numpy as np
from google.colab import files
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import plotly.figure_factory as ff
import matplotlib.pyplot as plt
import seaborn as sns
```

✓ Upload Excel Files

```
[3] uploaded = files.upload()

file_name = list(uploaded.keys())[0]
```

✓ Read Excel Sheet 'Closing Price' For Companies ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']

```
[8] sheet_name = 'Price' # Specify the sheet name

df = pd.read_excel(file_name, sheet_name=sheet_name)
df
```

✓ Descriptive Statistics

```
df.drop('Date',axis=1).describe()
```

✓ Show Figure Compare Closing Price For 5 Companies

```
[ ] # Create a figure for the histograms
fig = go.Figure()

# Loop through each company and add a histogram trace
for column in ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']:
    fig.add_trace(go.Histogram(x=df['Date'], y=df[column], name=column, histfunc='max', opacity=0.7))

# Update layout
fig.update_layout(
    title='Histogram of Highest Prices Each Companies',
    xaxis_title='Date',
    yaxis_title='Price',
    bargate='overlay', # Overlay histograms for better comparison
    bargate=0.5,
    title_x=0.5
)

# Show the plot
fig.show()
```

✓ Show Figure Outliers in Closing Prices For 5 Companies

```
fig = px.box(df, y=['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS'], title='Box Plot for Financial Data')

fig.update_layout(
    title='Box Plot for Financial Data',
    yaxis_title='Values',
    xaxis_title='Variables',
    title_x=0.5
)

fig.show()
```

✓ Show Figure to Changes Of Closing Prices For 5 Companies

```
fig = make_subplots(rows=2, cols=3, subplot_titles=['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS'])

# Add traces for each company
fig.add_trace(go.Scatter(x=df['Date'], y=df['MFPC'], mode='lines', name='MFPC'), row=1, col=1)
fig.add_trace(go.Scatter(x=df['Date'], y=df['AXPH'], mode='lines', name='AXPH'), row=1, col=2)
fig.add_trace(go.Scatter(x=df['Date'], y=df['EGAL'], mode='lines', name='EGAL'), row=1, col=3)
fig.add_trace(go.Scatter(x=df['Date'], y=df['ABUK'], mode='lines', name='ABUK'), row=2, col=1)
fig.add_trace(go.Scatter(x=df['Date'], y=df['ESRS'], mode='lines', name='ESRS'), row=2, col=2)

# Update x-axis and y-axis labels for each subplot
for i in range(1, 3):
    for j in range(1, 4):
        fig.update_xaxes(title_text="Date", row=i, col=j)
        fig.update_yaxes(title_text="Close Price", row=i, col=j)

# Update layout
fig.update_layout(height=800, width=1300, title_text="Line Chart For Closing Prices", title_x=0.5)

# Show the figure
fig.show()
```

✓ Show Figure The Largest Daily Return For 5 Companies

```
# Calculate daily returns as percentage change
for column in ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']:
    df[f'{column}_Daily_Return'] = df[column].pct_change()

# Find the largest daily return for each company
largest_daily_returns = {column: df[f'{column}_Daily_Return'].max() for column in ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']}

# Create a bar chart using Plotly
fig = go.Figure()

for company, largest_return in largest_daily_returns.items():
    fig.add_trace(go.Bar(
        x=[company],
        y=[largest_return],
        name=company
    ))

# Update layout
fig.update_layout(
    title='Largest Daily Return for Each Company',
    xaxis_title='Company',
    yaxis_title='Largest Daily Return',
    height=600,
    width=800,
    title_x=0.5
)

# Show the figure
fig.show()
```

✓ Show Figure The Largest Risk & Expected Return For 5 Companies

```
[ ] # Calculate daily returns as percentage change
for column in ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']:
    df[f'{column}_Daily_Return'] = df[column].pct_change()

# Calculate risk (standard deviation) and expected return (mean) for each company
risk_return_data = {
    'Company': [],
    'Risk': [],
    'Expected_Return': []
}

for column in ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']:
    risk_return_data['Company'].append(column)
    risk_return_data['Risk'].append(df[f'{column}_Daily_Return'].std())
    risk_return_data['Expected_Return'].append(df[f'{column}_Daily_Return'].mean())

# Convert to DataFrame
risk_return_df = pd.DataFrame(risk_return_data)

# Create a scatter plot using Plotly
fig = go.Figure()
```

```

for index, row in risk_return_df.iterrows():
    fig.add_trace(go.Scatter(
        x=[row['Expected_Return']],
        y=[row['Risk']],
        mode='markers+text',
        name=row['Company'],
        text=row['Company'],
        textposition='top center'
    ))

# Update layout
fig.update_layout(
    title='Risk vs Expected Return',
    xaxis_title='Expected Return',
    yaxis_title='Risk',
    height=600,
    width=800,
    title_x=0.5
)

# Show the figure
fig.show()

```

▼ Show Figure Daily Returns For Each Company

```

[ ] # Calculate daily returns for each company
for column in ['HFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']:
    df[f'{column}_Daily_Return'] = df[column].pct_change() # Calculate daily returns as percentage change

colors = {
    'HFPC': 'blue',
    'AXPH': 'green',
    'EGAL': 'red',
    'ABUK': 'purple',
    'ESRS': 'orange'
}

# Create separate Plotly figures for each company's daily returns with specified colors
figs = []

for column in ['HFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']:
    fig = go.Figure()
    fig.add_trace(go.Scatter(x=df['Date'], y=df[f'{column}_Daily_Return'], mode='lines', name=f'{column} Daily Return', line=dict(color=colors[column])))
    fig.update_layout(
        title=f'{column} Daily Returns',
        xaxis_title='Date',
        yaxis_title='Daily Return (%)',
        legend_title='Company',
        title_x=0.5,
        width=800,
        height=400
    )
    figs.append(fig)

# Show plots
for fig in figs:
    fig.show()

```

▼ Show Figure Pair Plot with Regression Lines for Companies Data

```
[ ] tech_rets = df[['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']].pct_change().dropna()

# Convert to Plotly
fig = px.scatter_matrix(tech_rets, dimensions=['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS'], title='Pair Plot with Regression Lines for Company Data')
fig.update_layout(title_x=0.5)
fig.show()
```

[Show hidden output](#)

▼ Display Figure Correlation Matrix

```
companies = ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS'] # Adjust based on your column names

# Calculate correlation matrix
correlation_matrix = df[companies].corr()

# Create heatmap using Plotly Figure Factory
fig = ff.create_annotated_heatmap(
    z=correlation_matrix.values,
    x=companies,
    y=companies,
    colorscale='pubugn',
    showscale=True,
    annotation_text=correlation_matrix.round(3).values, # Display correlation values
)

fig.update_layout(
    title='Correlation Matrix of Companies',
    yaxis_title='Companies',
    title_x=0.5 # Center title horizontally
)

# Show plot
fig.show()
```

▼ Read Excel Sheet 'Volume' For Companies ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']

```
[ ] sheet_name2 = 'Vol.'

df = pd.read_excel(file_name, sheet_name=sheet_name2)
df
```

[Show hidden output](#)

▼ Show Figure Volume For Each Companies (Over Time)

```
# Compute cumulative volume for each company
df['MFPC_Cumulative'] = df['MFPC'].cumsum()
df['AXPH_Cumulative'] = df['AXPH'].cumsum()
df['EGAL_Cumulative'] = df['EGAL'].cumsum()
df['ABUK_Cumulative'] = df['ABUK'].cumsum()
df['ESRS_Cumulative'] = df['ESRS'].cumsum()

# Create a line chart for cumulative volumes
fig = go.Figure()

companies = ['MFPC', 'AXPH', 'EGAL', 'ABUK', 'ESRS']
colors = ['teal', 'blue', 'green', 'purple', 'orange'] # Custom colors for each company

for company, color in zip(companies, colors):
    fig.add_trace(go.Scatter(x=df['Date'], y=df[f'{company}_Cumulative'], mode='lines', name=f'{company}', line=dict(color=color)))

# Update layout
fig.update_layout(
    title='Cumulative Trading Volume Over Time',
    xaxis_title='Date',
    yaxis_title='Cumulative Volume',
    title_x=0.5 # Center title horizontally
)

# Show plot
fig.show()
```

▼ Upload all 5 Companies

```
[9] uploaded = files.upload()

file_name = list(uploaded.keys())[0]
```

 [Show hidden output](#)

Double-click (or enter) to edit


▼ Upload Sheet 'EGAL' Company

```
[ ] # Specify the sheet name
sheet_name = 'EGAL'

df = pd.read_excel(file_name, sheet_name=sheet_name)
df
```

 [Show hidden output](#)

▼ Descriptive Statistics

```
 df.drop('Date', axis=1).describe()
```

▼ Show Figure (EGAL) Candlestick Chart with Volume

```
 # Assuming df is your DataFrame containing the required columns
fig = make_subplots(specs=[[{"secondary_y": True}]]

# Add Candlestick trace
fig.add_trace(go.Candlestick(x=df['Date'],
                             open=df['Open'],
                             high=df['High'],
                             low=df['Low'],
                             close=df['Price'],
                             increasing_line_color='green',
                             decreasing_line_color='red',
                             name = 'Price'),
              secondary_y=False)

# Add Volume trace
fig.add_trace(go.Bar(x=df['Date'], y=df['Vol.'], name='Volume'), secondary_y=True)

# Update layout
fig.update_layout(
    title='EGAL Candlestick Chart with Volume',
    xaxis_title='Date',
    yaxis_title='Price',
    yaxis2_title='Volume',
    title_x=0.5 # Center title horizontally
)

# Show plot
fig.show()
```


✓ Upload Sheet 'ABUK' Company

```
▶ # Specify the sheet name
sheet_name = 'ABUK'

df = pd.read_excel(file_name, sheet_name=sheet_name)
df
```

✓ Descriptive Statistics

```
[ ] df.drop('Date', axis=1).describe()
```

✓ Show Figure (ABUK) Candlestick Chart with Volume

```
▶ # Assuming df is your DataFrame containing the required columns
fig = make_subplots(specs=[[{"secondary_y": True}]])

# Add Candlestick trace
fig.add_trace(go.Candlestick(x=df['Date'],
                             open=df['Open'],
                             high=df['High'],
                             low=df['Low'],
                             close=df['Price'],
                             increasing_line_color='green',
                             decreasing_line_color='red',
                             name = 'Price'),
              secondary_y=False)

# Add Volume trace
fig.add_trace(go.Bar(x=df['Date'], y=df['Vol.'], name='Volume'), secondary_y=True)

# Update layout
fig.update_layout(
    title='ABUK Candlestick Chart with Volume',
    xaxis_title='Date',
    yaxis_title='Price',
    yaxis2_title='Volume',
    title_x=0.5 # Center title horizontally
)

# Show plot
fig.show()
```

✓ Upload Sheet 'ESRS' Company

```
[ ] sheet_name = 'ESRS' # Specify the sheet name

df = pd.read_excel(file_name, sheet_name=sheet_name)
df
```

✓ Descriptive Statistics

```
[ ] df.drop('Date', axis=1).describe()
```

✓ Upload Sheet 'AXPH' Company

```
[ ] # Specify the sheet name
sheet_name = 'AXPH'

df = pd.read_excel(file_name, sheet_name=sheet_name)
df
```

✓ Descriptive Statistics

```
[ ] df.drop('Date', axis=1).describe()
```

```
[ ] # Assuming df is your DataFrame containing the required columns
fig = make_subplots(specs=[[{"secondary_y": True}]])

# Add Candlestick trace
fig.add_trace(go.Candlestick(x=df['Date'],
                             open=df['Open'],
                             high=df['High'],
                             low=df['Low'],
                             close=df['Price'],
                             increasing_line_color='green',
                             decreasing_line_color='red',
                             name = 'Price'),
              secondary_y=False)

# Add Volume trace
fig.add_trace(go.Bar(x=df['Date'], y=df['Vol.'], name='Volume'), secondary_y=True)

# Update layout
fig.update_layout(
    title='AXPH Candlestick Chart with Volume',
    xaxis_title='Date',
    yaxis_title='Price',
    yaxis2_title='Volume',
    title_x=0.5 # Center title horizontally
)

# Show plot
fig.show()
```

✓ Upload Sheet 'MFPC' Company

```
[ ] sheet_name = 'MFPC' # Specify the sheet name

df = pd.read_excel(file_name, sheet_name=sheet_name)
df
```

✓ Descriptive Statistics

```
[ ] df.drop('Date', axis=1).describe()
```

✓ Show Figure (MFPC) Candlestick Chart with Volume

```
▶ # Assuming df is your DataFrame containing the required columns
fig = make_subplots(specs=[[{"secondary_y": True}]])

# Add Candlestick trace
fig.add_trace(go.Candlestick(x=df['Date'],
                             open=df['Open'],
                             high=df['High'],
                             low=df['Low'],
                             close=df['Price'],
                             increasing_line_color='green',
                             decreasing_line_color='red',
                             name = 'Price'),
              secondary_y=False)

# Add Volume trace
fig.add_trace(go.Bar(x=df['Date'], y=df['Vol.'], name='Volume'), secondary_y=True)

# Update layout
fig.update_layout(
    title='MFPC Candlestick Chart with Volume',
    xaxis_title='Date',
    yaxis_title='Price',
    yaxis2_title='Volume',
    title_x=0.5 # Center title horizontally
)

# Show plot
fig.show()
```

Appendix – II: Prediction of Financial Risk and Early Warning Model

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Dense, LSTM

✓ 4.0s Python

df = pd.read_excel("/Users/mohanad/Desktop/EGAL Historical Data.xlsx")

print(df)

Python

df = df.rename(columns = {'Price': 'Price', ' Open': 'Open', ' High': 'High', ' Low': 'Low'})

Python

df['Date'] = pd.to_datetime(df['Date'])
df = df.sort_values(by = 'Date')

Python

df

Python

plt.figure(figsize=(16,6))
plt.plot(df['Date'], df['Price'])
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price EGP', fontsize=18)
plt.show()

Python

# Create a new dataframe with only the 'Close column
data = df.filter(['Price'])

# Convert the dataframe to a numpy array
dataset = data.values

# Get the number of rows to train the model on
training_data_len = int(np.ceil( len(dataset) * .80 ))

# Create a new dataframe with only the 'Date column and Price
predictions_df = pd.DataFrame({
    "Date": df['Date'][:training_data_len:],
    "Actual": dataset[:training_data_len:, 0]
})

training_data_len

Python

predictions_df

Python

# Scale the data
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)

scaled_data

Python

# Create the training data set
# Create the scaled training data set
train_data = scaled_data[0:int(training_data_len), :]
# Split the data into x_train and y_train data sets
x_train = []
y_train = []

for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])
    if i<= 61:
        print(x_train)
        print(y_train)
        print()

# Convert the x_train and y_train to numpy arrays
x_train, y_train = np.array(x_train), np.array(y_train)

# Reshape the data
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

Python
```

```
# Build the LSTM model
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape= (x_train.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model.fit(x_train, y_train, batch_size=32, epochs=40)
```

Python

```
# Create the testing data set
# Create a new array containing scaled values
test_data = scaled_data[training_data_len - 60: , :]
# Create the data sets x_test and y_test
x_test = []
y_test = dataset[training_data_len:, :]
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])

# Convert the data to a numpy array
x_test = np.array(x_test)

# Reshape the data
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1 ))

# Get the models predicted price values
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)

# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean((predictions - y_test) ** 2))
rmse
```

Python

```
predictions_df["Predicted"] = predictions
```

Python

```
predictions_df.head(10)
```

Python

```
plt.figure(figsize=(16,6))
plt.plot(predictions_df['Date'], predictions_df[['Actual', 'Predicted']],)
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price EGP', fontsize=18)
plt.legend(['Actual Val', 'Predicted Val'], loc='lower right')
plt.show()
```

Python

```
# Define threshold
threshold = 4 # change between predicted prices

# Iterate through predictions dataframe and identify triggers
predicted_change = predictions_df["Predicted"].diff() # Calculate difference between consecutive predicted prices
predicted_change = predicted_change.dropna() # Remove the first element

for index, price_change in predicted_change.items():
    if abs(price_change) > threshold:
        direction = "Increase" if price_change > 0 else "Decrease"
        print(f"Early Warning: Predicted Price significant (direction) on {predictions_df.loc[index, 'Date']}")
```

Python

REFERENCES

1. Stock Market Prediction Using Machine Learning, December 2022, DOI:10.2991/978-94-6463-030-5_47, License CC BY-NC 4.0 In book: Proceedings of the 2022 International Conference on Bigdata Blockchain and Economy Management (ICBBEM 2022) (pp.458-465), author: Qingyi Chen.
2. Analysis and Prediction of Stock Market Trends Using Deep Learning, January 2020, In book: Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019) (pp.521-531), Publisher: Springer, authors: Harshit Agarwal and Gaurav Jariwala.
3. Prediction and Analysis of Corporate Financial Risk Assessment Using Logistic Regression Algorithm in Multiple Uncertainty Environment, September 2022, Authors: Xinyue Li, Saisai Yan, Jiayi Lu and Yanqiu Ding.
4. Stock Market Forecasting using Machine Learning: Today and Tomorrow, July 2019, Conference: 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), Authors: Sukhman Singh, Tarun Kumar Madan, Jitendra Kumar and Ashutosh Kumar Singh.
5. Stock market prediction using deep learning algorithms, august 2021, authors: Somnath Mukherjee, Bikash Sadhukhan, Nairita Sarkar, and Debajyoti Roy.
6. Project Work on PREDICTION MODEL FOR STOCK MARKET ANALYSIS Under the supervision of Prof. Dinesh Kumar. November 2017, Authors: Raghav mohta, Piyush Tekchandani, and Sumit Adikane.