|  |  |
| --- | --- |
| A picture of a winding road and trees  Senior Project  AI | Stock closing price prediction ...  Used ML & DL algorithms and neural networks  Hussein Issa |

**Table of Contents**

Introduction 1.1 Background  
1.2 Objectives  
1.3 Scope of the Project

Literature Review 2.1 Machine Learning in Finance  
2.2 Deep Learning Applications  
2.3 Stock Market Prediction Models

Methodology 3.1 Data Collection  
3.1.1 Apple Dataset  
3.1.2 Microsoft Dataset  
3.1.3 Google Dataset  
3.1.4 Cisco Dataset  
3.1.5 NBK Dataset  
3.1.6 Amazon Dataset  
3.2 Data Preprocessing  
3.3 Linear Regression Model  
3.4 LSTM Model in PyTorch  
3.5 LSTM Model in TensorFlow

Implementation 4.1 Environment Setup  
4.2 Code Structure  
4.3 Training the Models  
4.4 Evaluation

Results and Analysis 5.1 Linear Regression Results  
5.2 PyTorch LSTM Results  
5.3 TensorFlow LSTM Results  
5.4 Comparative Analysis

Discussion 6.1 Interpretation of Results  
6.2 Model Performance and Limitations  
6.3 Insights Gained

Conclusion 7.1 Summary of Findings  
7.2 Contributions to the Field  
7.3 Future improvement

1. **Introduction**

In the rapidly evolving landscape of technology and finance, the integration of machine and deep learning algorithms has become increasingly pivotal for data analysis and decision-making processes. This senior project embarks on a comprehensive exploration and implementation of various machine learning and deep learning techniques, including Linear Regression, Long Short-Term Memory (LSTM) models using both PyTorch and TensorFlow frameworks.

The primary focus of this project is to leverage these advanced algorithms for predicting and analyzing financial data from prominent companies such as Apple, Microsoft, Google, Cisco, National Bank of Kuwait (NBK), and Amazon. By employing a diverse range of datasets from these industry giants, the project aims to uncover meaningful insights, trends, and patterns in the stock market.

The significance of this project lies in its practical application of cutting-edge technologies to address real-world challenges in the financial domain. By utilizing machine learning models, we aim to enhance the accuracy of stock price predictions, enabling investors and financial analysts to make more informed decisions. The comparative analysis of LSTM implementations in PyTorch and TensorFlow offers a unique perspective on the strengths and nuances of these frameworks, contributing valuable insights to the broader field of deep learning.

This report provides a detailed account of the methodology, implementation, and results obtained throughout the course of the project. Each section delves into the specific aspects of the algorithms used, the datasets employed, and the implications of the findings. The ultimate goal is to showcase the practical utility of machine and deep learning in the financial sector and to contribute to the growing body of knowledge in this interdisciplinary field.

Through this project, we aspire to demonstrate the power of artificial intelligence in transforming traditional financial analysis methods, paving the way for more informed decision-making and strategic investments in an ever-changing global market.

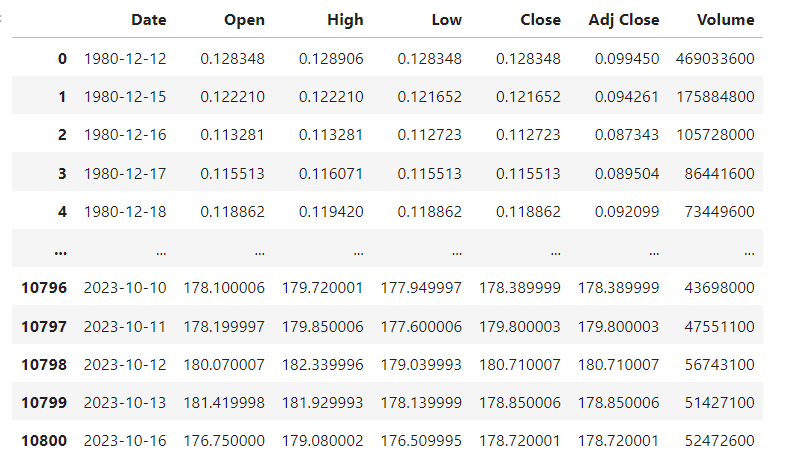
1. **Importance**

Machine learning (ML) has emerged as a transformative tool in the field of finance, revolutionizing traditional approaches to data analysis and decision-making. The literature on machine learning in finance emphasizes its applications in risk management, fraud detection, algorithmic trading, and credit scoring. Researchers have explored various ML techniques such as decision trees, support vector machines, and ensemble methods to model complex financial systems. Noteworthy studies highlight the effectiveness of ML in predicting market trends, optimizing portfolio management, and enhancing financial decision processes. The intersection of machine learning and finance continues to evolve, with a focus on addressing challenges like market volatility and data non-linearity.

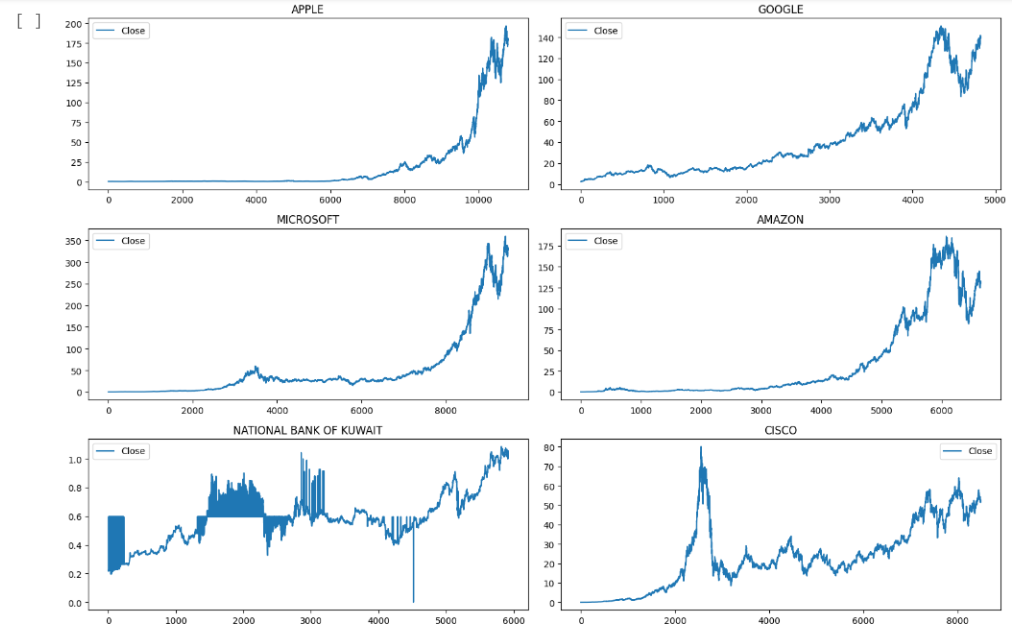
Deep learning, a subset of machine learning, has demonstrated remarkable success in handling complex and high-dimensional data, making it particularly appealing for financial modeling. The application of deep learning in finance involves neural networks with multiple layers, capable of learning intricate patterns and representations from raw data. This section delves into the literature exploring the diverse applications of deep learning in finance, including but not limited to time series forecasting, risk management, and algorithmic trading. Researchers have investigated various architectures, such as recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks, showcasing their effectiveness in capturing temporal dependencies and improving predictive accuracy. The section aims to provide insights into the advancements and challenges associated with integrating deep learning techniques into financial modeling.

Machine Learning Models: Regression-based models, support vector machines (SVM), and ensemble methods like Random Forests have been applied to predict stock prices based on historical data and relevant features.

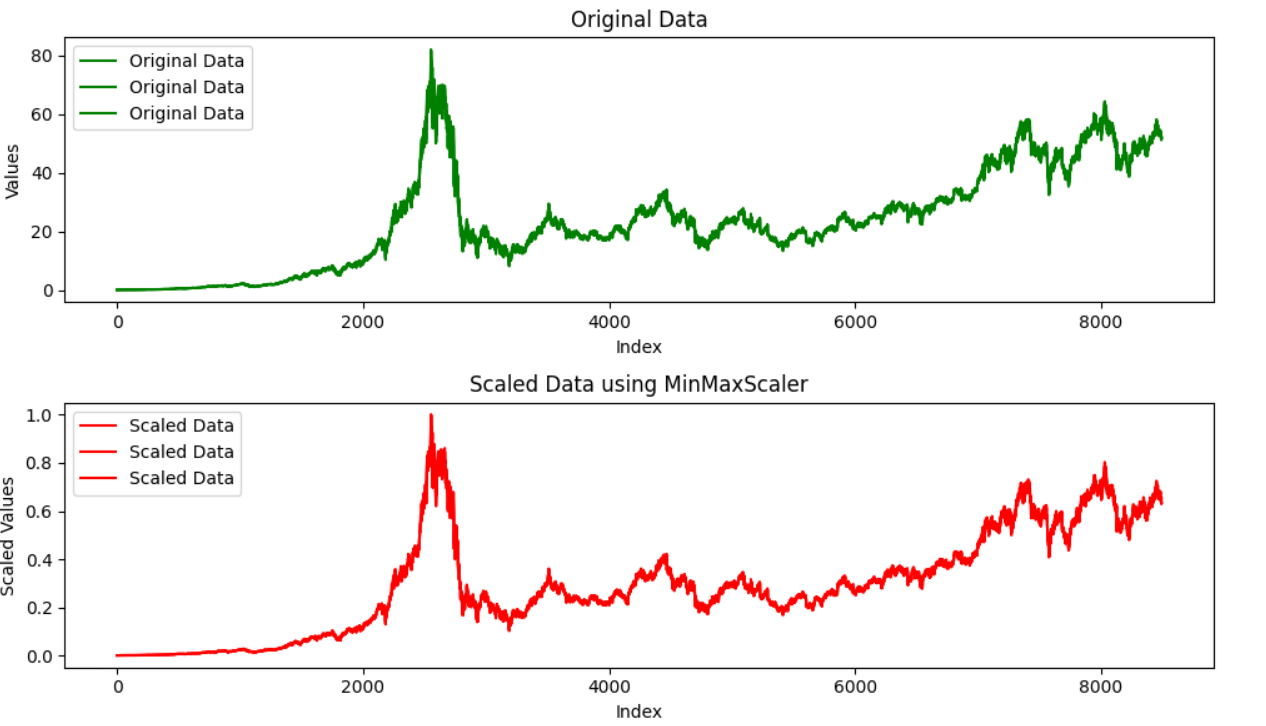
Deep Learning Models: Recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks are increasingly popular for modeling sequential data and have shown promise in capturing complex patterns in stock price movements.

1. Data 

Financial datasets play a crucial role in the development and evaluation of predictive models. For this project, we collected data from prominent technology and financial companies to ensure a diverse representation of the market globally like from America , Kuwait etc. ...



Before feeding the data into our models, a crucial preprocessing step was undertaken. This involved cleaning the datasets, handling missing values, and normalizing the features to ensure consistency and reliability. Time-series data required careful treatment, including the creation of lag features to capture temporal dependencies. The preprocessed data sets the foundation for accurate and meaningful model training.



A computer code with text

Description automatically generated with medium confidence

A screenshot of a computer code

Description automatically generated

Linear regression serves as a fundamental baseline model for predicting stock prices. We implemented a linear regression model to establish a benchmark for comparison with more complex models. The model leverages historical stock prices, trading volumes, and other relevant financial indicators as features. The training process involves optimizing the model parameters to minimize the difference between predicted and actual stock prices.

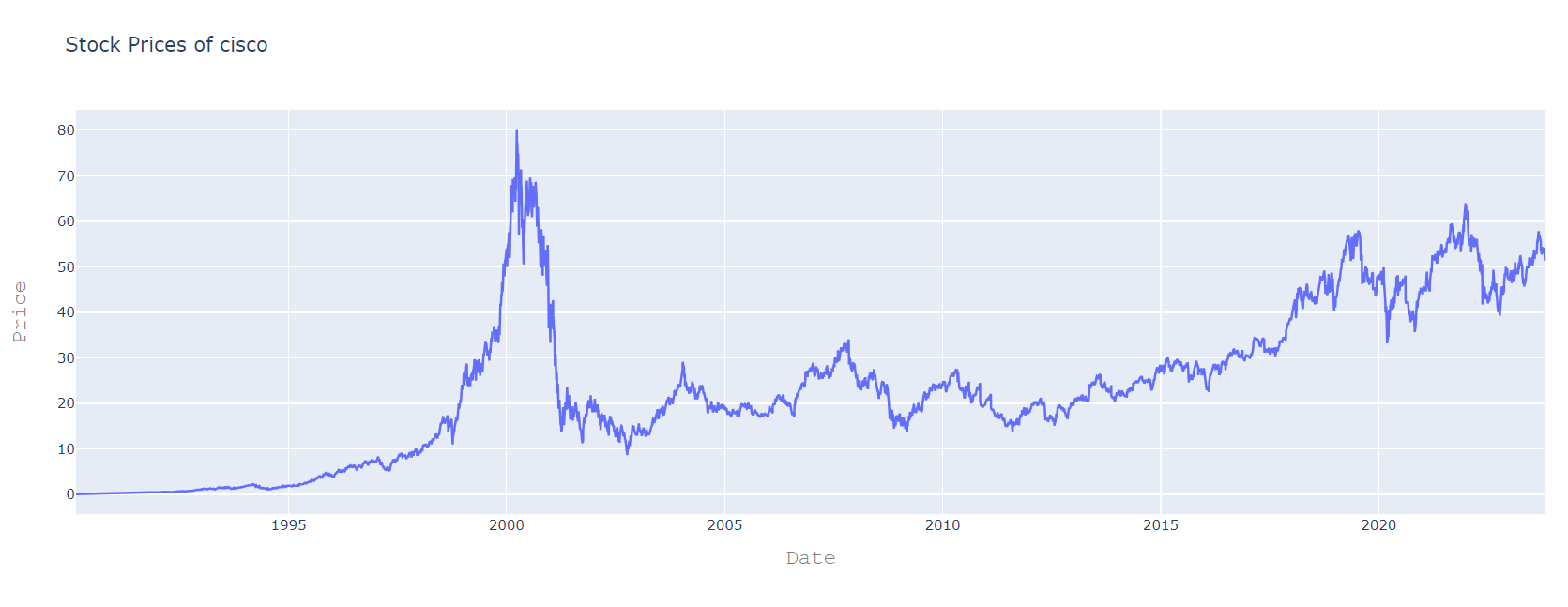
Long Short-Term Memory (LSTM) networks in PyTorch offer a powerful tool for capturing sequential dependencies in time-series data. We implemented an LSTM model to exploit the temporal patterns present in our financial datasets. The PyTorch framework facilitated efficient model training, enabling the network to learn and adapt to the nuanced dynamics of the stock market.

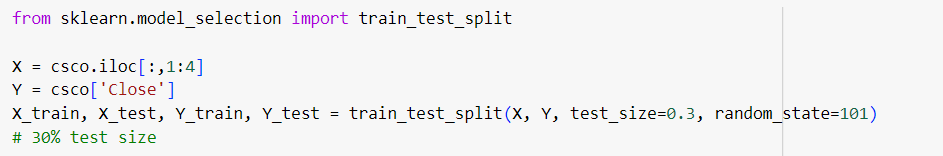
In parallel, we implemented an LSTM model using TensorFlow, another widely used deep learning framework. This provided an opportunity for comparative analysis between PyTorch and TensorFlow implementations. The TensorFlow LSTM model followed a similar structure to the PyTorch model, emphasizing the flexibility and transferability of deep learning architectures across frameworks.

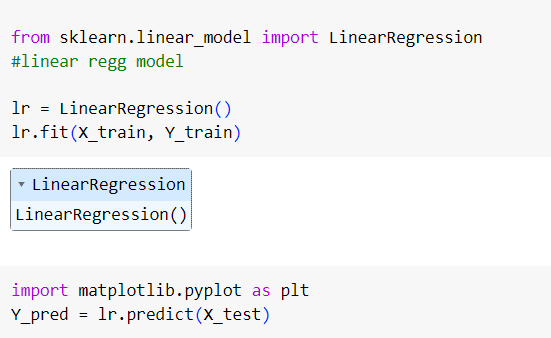
1. Implementation

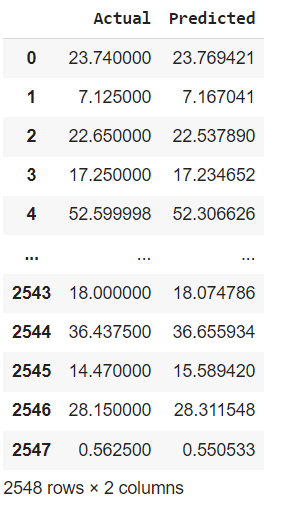
Used google colab mainly plus the jupyter notebook to implement the codes using python.

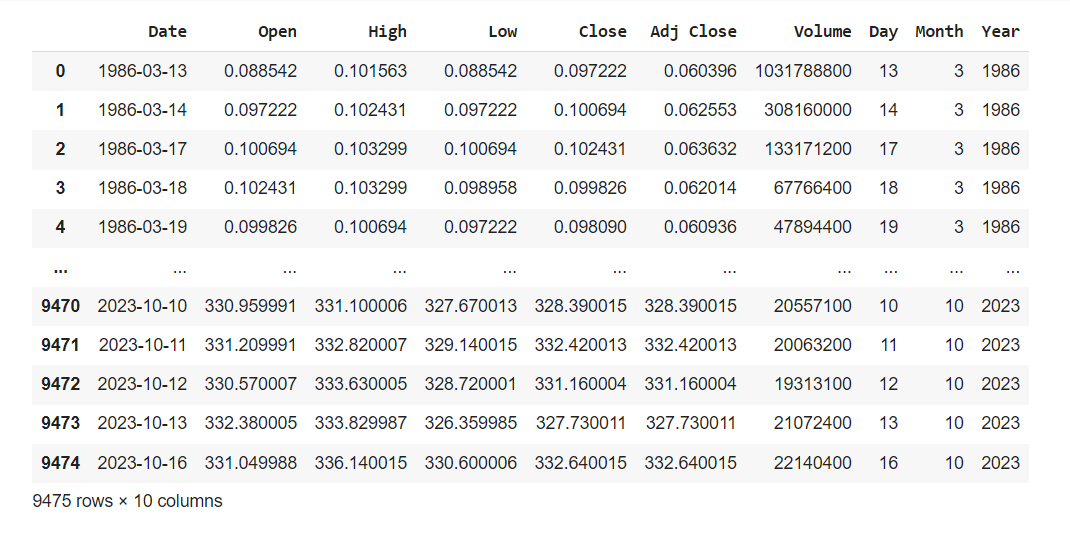
Starting with linear regression + knn Regressor + Random Forest Regressor :



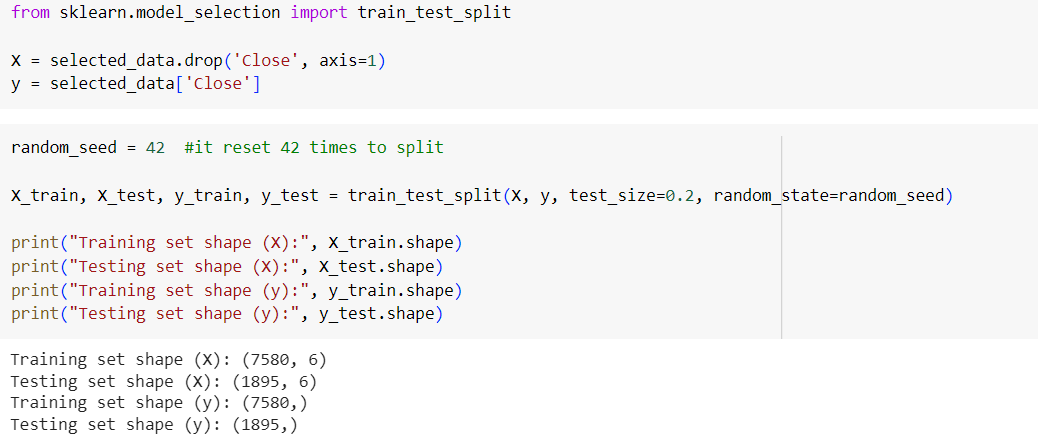
Here I split the data into training and testing parts I took the training part 70% after many tries with different values.

fitting the data into the model. Then predicting it using the x test.

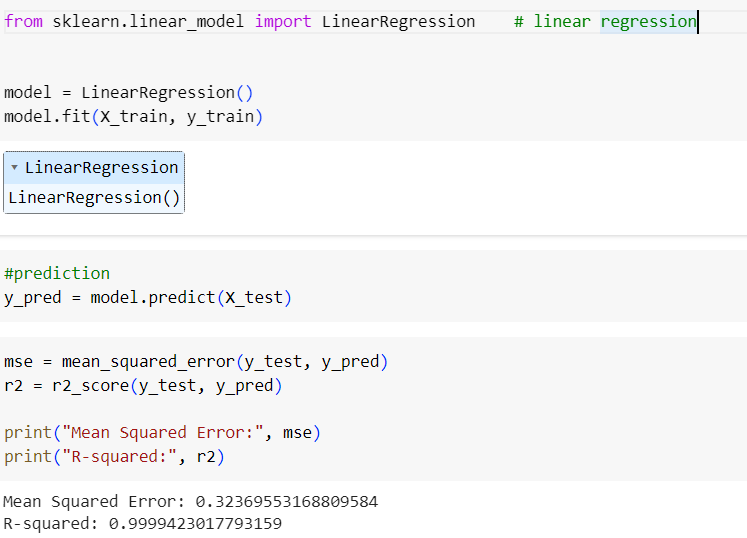
here is a sample of predicted values with their original values after training.



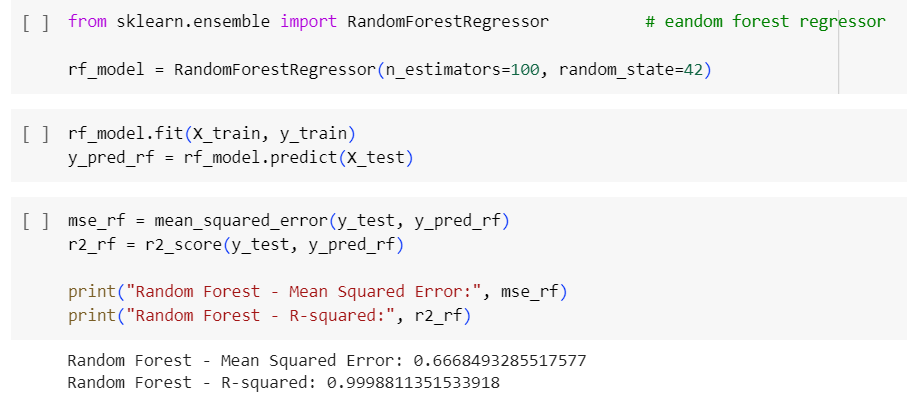
In Microsoft dataset I made the date column into 3 columns to specify each feature as input.



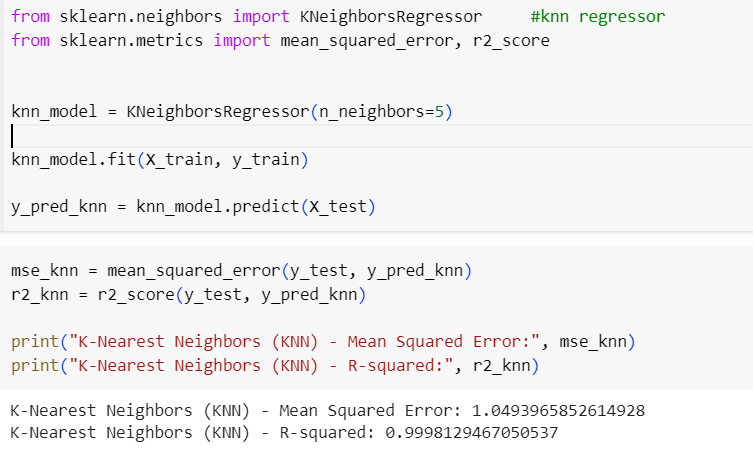
Splitting using the sklearn library.

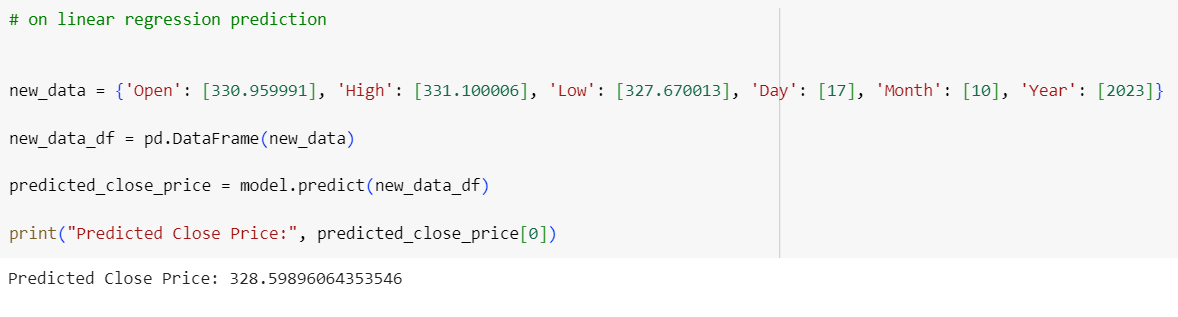
1st model:

2nd model:

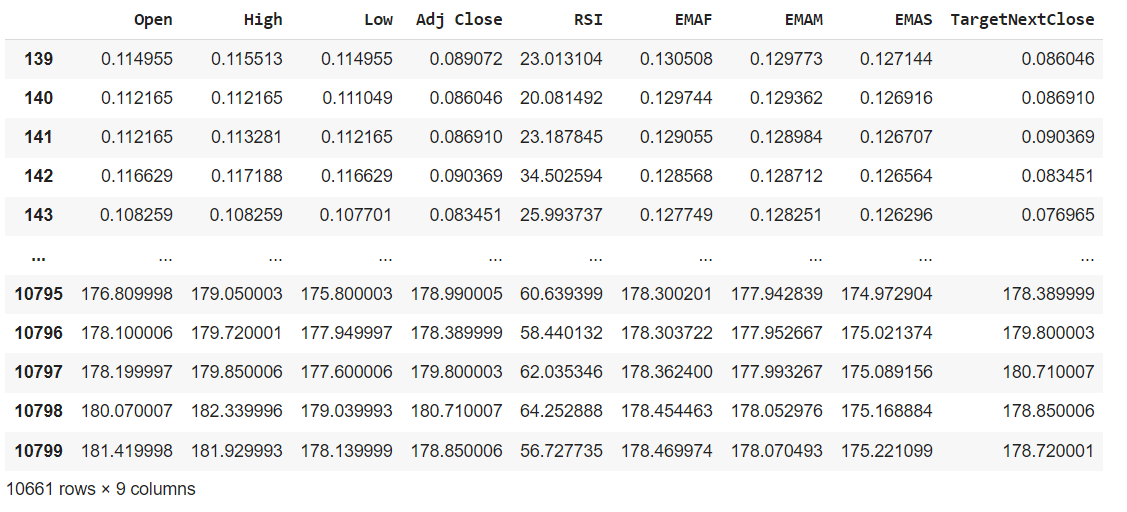


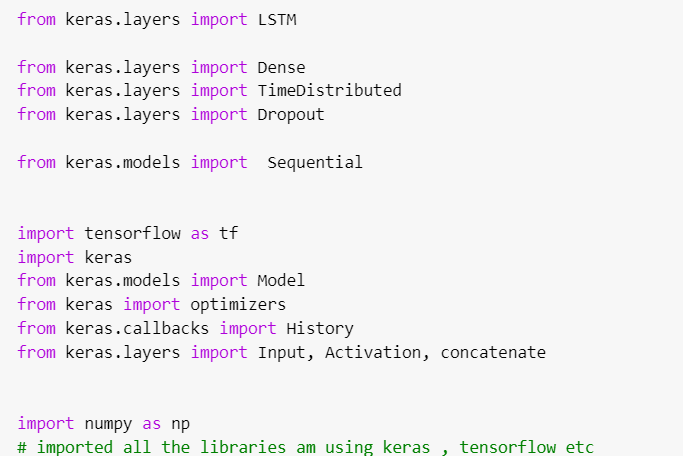
3rd model:

predict new data next day :



with LSTM TensorFlow Framework :

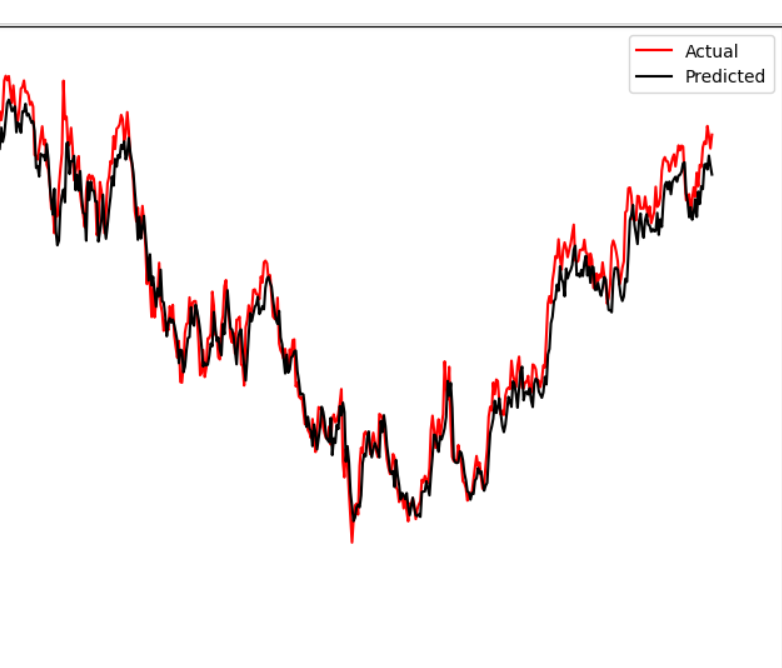
added useful columns like moving averages and price speed and added the target of next day in each before day.



A screenshot of a computer

Description automatically generated

Training the model with the lstm layers and backcandels and trying different number of epochs.



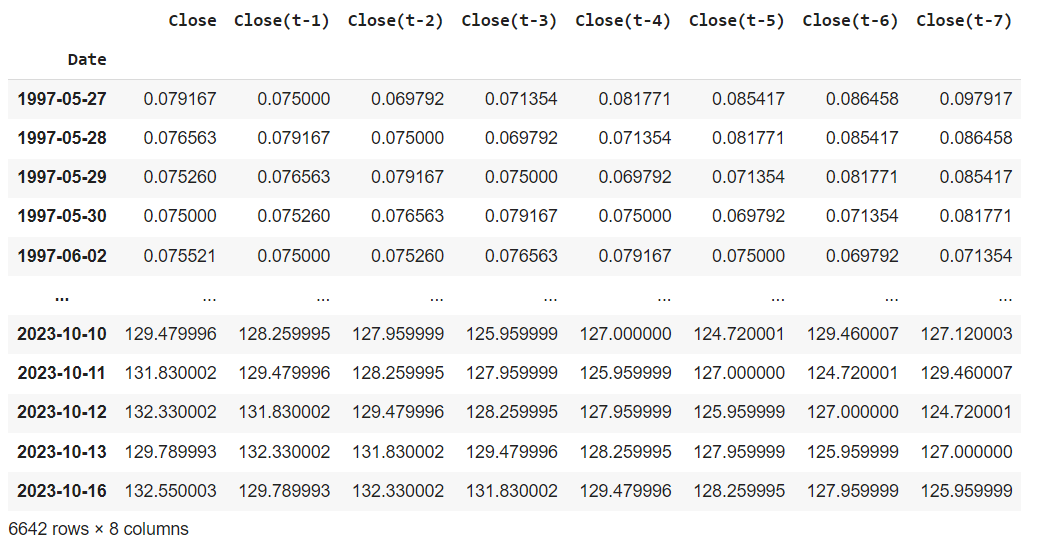
Plot of the results using matplotlib library.

A computer screen shot of a error

Description automatically generated

96% accuracy after training for hours and low mean errors so this model considered very good results.

with LSTM pyTorch Framework :

I will be predicting based on the 7 days before of the close price..

A screenshot of a computer code

Description automatically generated

Took around 95% of the data as training set

A screenshot of a computer program

Description automatically generated

each epoch should make a loop and optimixe it as the batch size..

A screenshot of a computer

Description automatically generatedA computer screen shot of a program

Description automatically generatedwonderful results

here I predicted the next day closing price with the 7 last days of closing price in the dataset and got the predicted value after inversing the minmax scaler.

Well!

The results obtained from our analysis provide valuable insights into the predictive capabilities of the implemented models. In interpreting the results, it is crucial to consider the performance of each model against the diverse set of financial datasets. The linear regression model, serving as our baseline, offers a straightforward interpretation. It highlights the influence of various financial indicators on stock prices, providing a foundation for understanding the impact of individual features.

The LSTM models in PyTorch and TensorFlow, leveraging the temporal dependencies in the data, exhibit nuanced performance. The interpretation involves deciphering the significance of learned patterns in the time series. Both models capture complex relationships, showcasing their ability to adapt to the intricate dynamics of the stock market. Understanding how these models weigh the importance of different features and time intervals is essential for practical application and decision-making.

Assessing the performance of the models reveals the strengths and limitations inherent in each approach. The linear regression model, while simplistic, sets a benchmark for straightforward interpretability. However, it may struggle to capture the nonlinear relationships and temporal dependencies present in financial time series.

The LSTM models, on the other hand, demonstrate superior performance in capturing complex patterns, especially during periods of market volatility. However, their black-box nature makes it challenging to interpret the exact features and patterns influencing predictions. Additionally, overfitting remains a concern, especially when dealing with limited datasets. Addressing these limitations is crucial for enhancing the reliability and robustness of predictive models in financial applications.

Through the course of this project, several key insights have emerged. The analysis of financial datasets from diverse sectors reveals that certain features, such as trading volumes and historical prices, consistently impact stock price movements. The LSTM models, particularly in PyTorch and TensorFlow, showcase the potential for deep learning to uncover intricate temporal dependencies, providing a nuanced understanding of market dynamics.

Furthermore, the comparative analysis between the frameworks offers insights into the practical considerations of choosing between PyTorch and TensorFlow for financial modeling. While both frameworks yield comparable results, the ease of use, flexibility, and community support may influence the choice depending on the specific project requirements.

Summary

I collected the data then I started to preprocess the data that I collected and see if there is missing values that can affect my data accuracy. I used features engineering like minmax scaler and standard, I created in some models new features based on the existed features that can help me in training the model, I splitted the data into training and testing parts I tried different numbers in splitting to see the accuracy. Then I trained the model by fitting the training parts in some models I used epochs and batches that used for training progress based on the loss that is happening , then predicted the target by applying the test part to it then comparing them, then I concluded the r2 score and the errors to see if the model is fitted well and plotted a lot of plots so we can conclude the result visually at first sight.

The problems I faced , the main ones are some values was having missing values and null rows and 0 rows so I had to deal with that and minmax scaler was hard to inverse it after the prediction part so I can see the real values. Also the data set features was only 4 so I had to create more feature to make the model more stronger.

The final result of the linear and lstm prediction , they all got an r2 score above 90% which considered very good models and we can depend on them

In future I would collect more specific databases that reach the yesterday date that can predict the real time now.