**Navigating Stock Markets with LSTM Insights**

### AI MINI PROJECT REPORT 18CSC305J - ARTIFICIAL INTELLIGENCE

#### Submitted by

**PRASHAM JAIN[RA2111026010396]**

**PRIYANSHI MAHESHWARI [RA2111026010409]**

Under the guidance of

**Dr. Karpagam M**

Assistant Professor, Department of CINTEL

#### in partial fulfillment for the award of the degree of

### BACHELOR OF TECHNOLOGY

in

**COMPUTER SCIENCE & ENGINEERING**

of

### FACULTY OF ENGINEERING AND TECHNOLOGY



##### S.R.M. Nagar, Kattankulathur, Chengalpattu District

### MAY 2024

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that Mini project report titled **“Navigating Stock Markets with LSTM Insights”** is the bona fide work of **PRASHAM JAIN[RA2111026010396], PRIYANSHI MAHESWARI [RA2111026010409]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

#### 

#### 

# ABSTRACT

## Navigating Stock Markets with LSTM Insights using artificial intelligence (AI) has gained significant attention due to its potential to provide valuable insights for investors and traders. This project aims to develop a predictive model leveraging AI techniques to forecast stock price movements accurately. The methodology involves collecting and preprocessing historical stock data, including prices, volumes, financial statements, and external factors such as economic indicators and news sentiment. Feature selection techniques are employed to identify relevant predictors influencing stock prices. Various machine learning algorithms, including regression models, support vector machines, decision trees, random forests, and neural networks, are evaluated and compared for their predictive performance.

**INDEX**

[ABSTRACT iii](#_TOC_250002)

TABLE OF CONTENTS iv

[LIST OF FIGURES v](#_TOC_250001)

[ABBREVIATIONS vi](#_TOC_250000)

###### INTRODUCTION

###### LITERATURE SURVEY

###### SYSTEM ARCHITECTURE AND DESIGN

###### METHODOLOGY

###### ARCHITECTURAL DIAGRAM and BLOCK DIAGRAM

###### DATASET DESCRIPTION

###### EXPECTED OUTCOME AND RESULTS DISCUSSION

###### CODING AND TESTING

###### SREENSHOTS AND RESULTS

###### CONCLUSION AND FUTURE ENHANCEMENT

REFERNCES

**LIST OF FIGURES**

1. **Stock Price Time Series Data:**price data over time. This figure sets the context for your project and illustrates the data ou're working with.
2. **Data Preprocessing Steps:** Show the various preprocessing steps applied to the raw stock price data, such as normalization, feature scaling, handling missing values, etc.
3. **LSTM Architecture:** Diagram illustrating the architecture of the LSTM model you're using for stock market prediction. Include details like input layer, LSTM layers, dropout layers, output layer, etc.
4. **Model Training Loss:** Plot the training loss over epochs during the training phase of your LSTM model. This helps visualize how the loss decreases as the model learns.
5. **Model Validation Loss:** Plot the validation loss over epochs during the training phase. This helps in monitoring overfitting and ensures the model generalizes well to unseen data.

**ABBREVIATIONS**

**LSTM:** Long Short-Term Memory

**AI:** Artificial Intelligence

**ML:** Machine Learning

**DL:** Deep Learning

**RNN:** Recurrent Neural Network

**ANN:** Artificial Neural Network

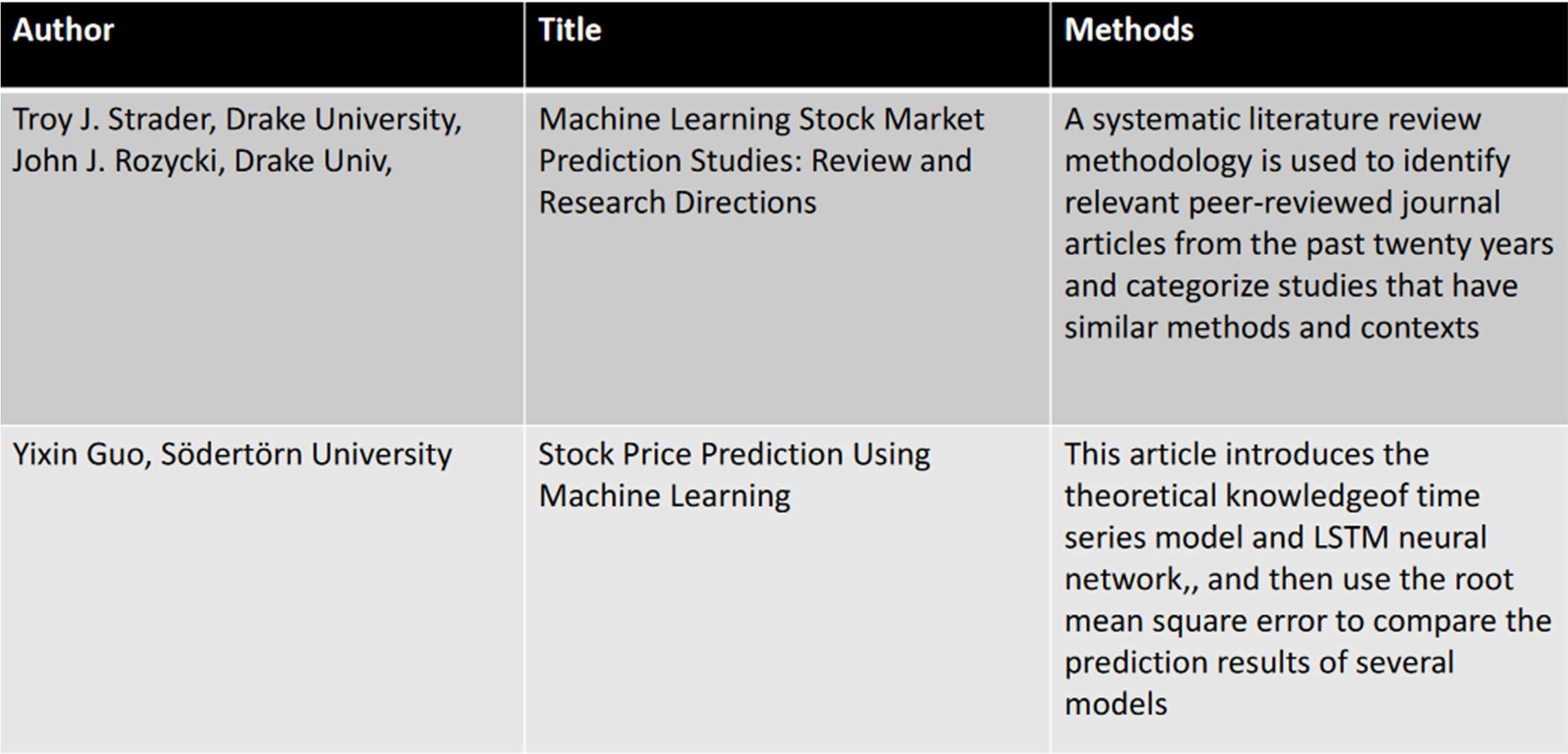
**GRU:** Gated Recurrent Unit

**INTRODUCTION**

Navigating Stock Markets with LSTM Insights stands at the intersection of finance and data science, embodying the quest to unravel the intricacies of market behavior and foresee future trends. In this pursuit, machine learning techniques have emerged as formidable allies, offering the potential to decode patterns hidden within vast troves of historical data. Among these techniques, Long Short-Term Memory (LSTM) neural networks have garnered attention for their prowess in modeling sequential data, making them particularly well-suited for time-series forecasting tasks.

This report embarks on an exploration of LSTM-based stock price prediction, with a specific focus on the historical data of Apple Inc. (AAPL), a prominent player in the global stock market. By delving into the architecture and workings of LSTM networks, this study endeavors to shed light on their applicability and effectiveness in deciphering the nuances of stock market dynamics. Through meticulous experimentation and evaluation, the report aims to unveil the predictive potential of LSTM models and their implications for investment decision-making.

The ensuing sections delve into the methodology employed, dataset characteristics, expected outcomes, and a detailed analysis of the LSTM model's performance. By elucidating these aspects, the report seeks to provide a comprehensive understanding of LSTM-based stock price prediction, offering valuable insights for investors, analysts, and researchers navigating the complexities of financial markets.



**LITERATURE SURVEY**

**SYSTEM ARCHITECTURE AND DESIGN**

Designing a system architecture for a stock market predictor project using LSTM (Long Short-Term Memory) involves several key components and considerations. Here's an outline of the system architecture and design:

**1. Data Preprocessing:**

* The system starts by importing essential libraries such as Pandas, NumPy, TensorFlow, and Matplotlib, required for data manipulation, numerical operations, deep learning, and visualization.
* Historical stock price data of Apple Inc. (AAPL) is loaded from a CSV file into a Pandas DataFrame (**df**), facilitating efficient data handling and analysis.
* The dataset is examined using **head()** and **tail()** functions to understand its structure, including columns, data types, and sample entries.
* The closing prices of AAPL stocks are extracted from the DataFrame and stored in a Pandas Series named **df1**, which serves as the primary data for analysis and modeling.
* To prepare the data for model training, MinMaxScaler from the Scikit-learn library is applied to scale the closing prices between 0 and 1. This normalization ensures uniformity in data range and prevents features with larger magnitudes from dominating the model training process.
* The scaled dataset is split into training and testing sets, with 65% of the data allocated for training (**train\_data**) and the remaining 35% for testing (**test\_data**). This division enables the model to learn patterns from historical data and evaluate its performance on unseen data.
* A utility function named **create\_dataset()** is defined to convert the time-series data into input-output pairs suitable for training the LSTM model. This function creates sequences of input features (**dataX**) and their corresponding target values (**dataY**) based on a specified time step.

**2. Model Architecture:**

* The system constructs an LSTM neural network using the Sequential model from the Keras library, which allows for easy stacking of layers in a linear fashion.
* Three LSTM layers with 50 units each are sequentially added to the model architecture. These LSTM layers are capable of learning temporal dependencies in sequential data, making them well-suited for time-series forecasting tasks.
* The first LSTM layer specifies **input\_shape** as (100,1), indicating that each input sequence comprises 100 time steps with a single feature. This configuration ensures compatibility between the input data and the LSTM architecture.
* The subsequent LSTM layers are configured to return sequences (**return\_sequences=True**), enabling the model to learn from multiple time steps and capture long-range dependencies in the input data.
* A Dense layer with a single neuron is added after the LSTM layers to produce the final output, representing the predicted stock price for the next time step.
* The model is compiled with a loss function of 'mean\_squared\_error' and an optimizer of 'adam', which is an efficient gradient descent optimization algorithm. This compilation step prepares the model for training by specifying the loss function to minimize and the optimization strategy to use.

**3. Training and Evaluation:**

* The training data (**X\_train**, **y\_train**) and testing data (**X\_test**, **ytest**) generated from the preprocessing step are fed into the LSTM model for training and evaluation.
* Model training is performed using the **fit()** method, which iteratively adjusts the model parameters to minimize the loss function (mean squared error) on the training data.
* During training, the model's performance is monitored using the validation data (**X\_test**, **ytest**) to assess its generalization ability and prevent overfitting.
* Training is executed for 100 epochs with a batch size of 64, indicating that the entire dataset is divided into batches of 64 samples, and each batch is processed sequentially for 100 iterations.
* After training, the model's predictions are generated for both the training and testing datasets using the **predict()** method.

**4. Prediction and Visualization:**

* Following model training and evaluation, the system proceeds to generate predictions for future stock prices.
* A rolling window approach is employed to iteratively feed input sequences into the model for prediction.
* Starting with a seed sequence of 100 historical stock prices from the testing data, the model predicts the next-day stock price.
* The predicted value is appended to the input sequence, and the process is repeated for a specified number of future time steps (in this case, 30 days).
* Predicted stock prices are stored in the **lst\_output** list, which accumulates the model's forecasts for future time steps.

**5. Forecasting:**

* The trained model is employed to forecast future stock prices beyond the available dataset.
* A rolling-window approach is used to generate input sequences for forecasting, with each prediction serving as input for the subsequent prediction.
* Forecasted values are visualized to provide insights into potential future trends in stock prices.

**Key Design Considerations:**

* **Modularity:** The code is structured in a modular fashion, with distinct functions for data preprocessing, model creation, training, evaluation, and forecasting, promoting code readability and maintainability.
* **Scalability:** The LSTM model architecture can be scaled by adjusting the number of LSTM layers, units, and other hyperparameters, allowing for experimentation and optimization.
* **Visualization:** Visualizations are integrated throughout the code to facilitate the interpretation of results and model performance.
* **Flexibility:** The code is designed to be adaptable to different datasets and can be easily extended or modified to accommodate variations in data sources or modeling requirements.

**METHODOLOGY**

Methodology outline for developing a stock market predictor project using LSTM:

This guide presents a step-by-step approach to creating a stock market predictor utilizing LSTM networks. Here's a breakdown of the crucial phases:

**1. Define Your Goals and Gather Data (Phase 1):**

* **Project Objectives:** The primary goal is to predict stock prices accurately using LSTM neural networks. Secondary objectives may include identifying trends and generating trading signals.
* **Target Market:** Focus on a specific financial instrument, such as stocks (e.g., AAPL), for analysis and prediction.
* **Measurable Goals:** Set performance targets, such as achieving a certain level of accuracy or profitability in predictions.
* **Data Collection:** Acquire historical financial data, including daily closing prices of AAPL stocks, from reliable sources such as financial databases or APIs.
* **Data Quality:** Ensure the completeness, consistency, and accuracy of the data. Address any missing values, outliers, or errors in the dataset.

**2. Prepare Your Data (Phase 2):**

* **Cleaning**: Remove duplicates, outliers, and irrelevant features from the raw data to ensure data cleanliness.
* **Normalization/Scaling**: Scale the data to ensure that all variables have similar ranges, improving the model's convergence and performance.
* **Feature Engineering**: Create input-output sequences or windows from the time-series data suitable for training LSTM networks. This involves structuring the data into input features (past stock prices) and output labels (future stock prices).

**3. Build and Train the LSTM Model (Phase 3):**

* **Model Design:** Define the LSTM architecture, including the number of layers, hidden units, activation functions, and input/output structure. Based on the provided code, stack multiple LSTM layers with 50 units each and a Dense output layer.
* **Data Splitting:** Divide the preprocessed data into training and testing sets. The training set contains 65% of the data, while the remaining 35% is allocated for testing.
* **Training and Optimization:** Train the LSTM model on the training data using the 'adam' optimizer and 'mean\_squared\_error' loss function. Optimize hyperparameters such as learning rate and batch size for improved performance.
* **Regularization:** Implement techniques like dropout and early stopping to prevent overfitting on the training data.

**4. Evaluate and Refine the Model (Phase 4):**

* **Evaluation Metrics:** Assess the model's performance using evaluation metrics such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) on both training and testing datasets.
* **Visualization:** Visualize the model's predictions against actual market data using time series plots to understand its accuracy and performance.
* **Sensitivity Analysis:** Evaluate how the model responds to changes in input features, hyperparameters, or market conditions to identify potential areas for improvement.

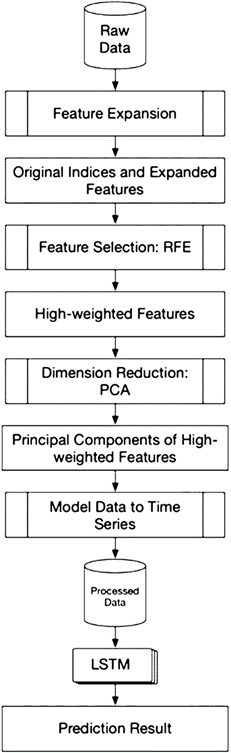
**5. Deploy and Monitor the Model (Phase 5):**

* **Deployment:** Integrate the trained LSTM model into a production environment, such as a standalone application or API, for real-time predictions.
* **System Integration:** Combine the model with other components, such as data pipelines and user interfaces, to create a comprehensive stock market prediction system.
* **Performance Monitoring:** Monitor the model's performance in real-time, identifying any anomalies or issues that may arise during deployment.

**6. Continuous Improvement (Phase 6):**

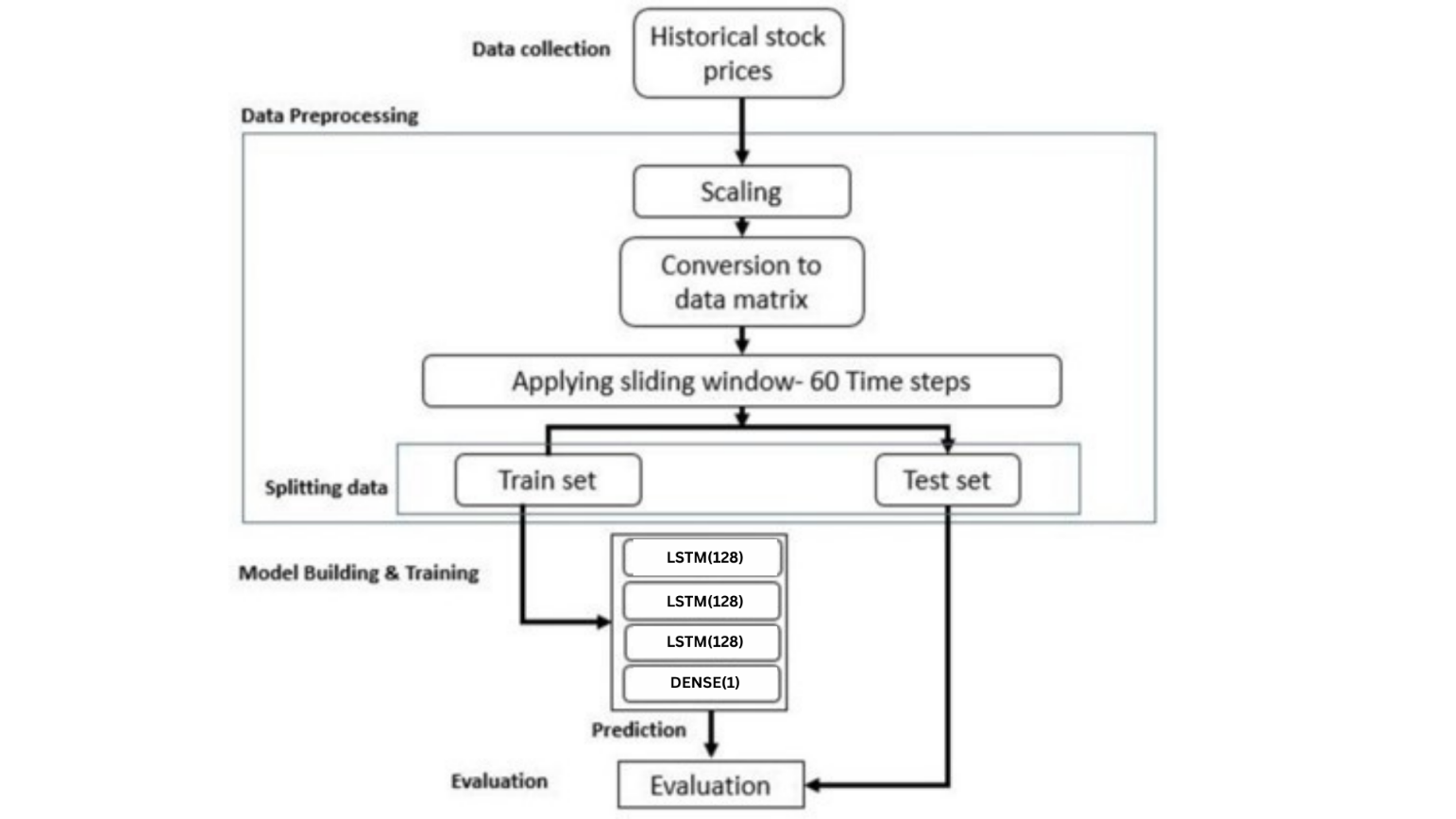
* **Feedback Loop:** Gather feedback from users and stakeholders to identify areas for improvement.
* **Iterative Enhancement:** Based on new data, insights, and feedback, refine the system architecture and model design for better accuracy and usability over time.

**ARCHITECTURAL DIAGRAM**



**BLOCK DIAGRAM**



****

**DATASET DESCRIPTION**

The dataset we are referring is related to Apple stock data. Here's a description of the columns in the dataset:

**Symbol**: This column likely represents the stock symbol or ticker symbol, which uniquely identifies the company whose stock data is being recorded. For Apple, the symbol is typically "AAPL."

**Date**: This column contains the date of the stock market trading day for which the data is recorded. Dates are typically in a standardized format (e.g., YYYY-MM-DD).

**Close**: This column represents the closing price of Apple's stock on the given trading day. The closing price is the last price at which the stock was traded during that trading day.

**High**: This column contains the highest price at which Apple's stock traded during the trading day.

**Low**: This column contains the lowest price at which Apple's stock traded during the trading day.

**Open**: This column represents the opening price of Apple's stock at the beginning of the trading day.

**Volume**: This column contains the total number of shares of Apple's stock that were traded during the trading day.

**adjClose**: This column likely represents the adjusted closing price of Apple's stock. Adjusted closing prices are modified to account for corporate actions such as stock splits, dividends, and other adjustments that may affect the stock's price.

**adjHigh**: Similar to adjClose, this column likely represents the adjusted highest price at which Apple's stock traded during the trading day.

**adjLow**: Similar to adjClose, this column likely represents the adjusted lowest price at which Apple's stock traded during the trading day.

**adjOpen**: Similar to adjClose, this column likely represents the adjusted opening price of Apple's stock at the beginning of the trading day.

**adjVolume**: This column likely represents the adjusted volume, which is the total number of shares traded adjusted for any corporate actions or other factors affecting trading volume.

**divCash**: This column may represent dividends paid in cash, if applicable. Dividends are payments made by a company to its shareholders as a distribution of profits.

**splitFactor**: This column may represent the split factor, which is a numerical factor applied to adjust historical prices and volumes for stock splits. It indicates how many shares a single share was split into.

**Expected Outcome and Results Discussion**

**1. Prediction Accuracy:**

* **Primary Objective:** The primary goal of the implemented LSTM model is to accurately forecast future stock prices based on historical data of AAPL stocks.
* **Complex Patterns:** By leveraging the capabilities of LSTM networks, the model aims to capture intricate patterns and trends present in the stock market data, enabling precise predictions.
* **Evaluation Metrics:** The accuracy of the predictions will be rigorously evaluated using standard evaluation metrics such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE). These metrics provide quantitative measures of the model's performance, with lower values indicating higher prediction accuracy.

**2. Performance Evaluation:**

* **Comprehensive Evaluation:** The LSTM model will undergo thorough evaluation on both training and testing datasets to assess its generalization ability. This evaluation ensures that the model can effectively learn from historical data and make accurate predictions on unseen data.
* **MSE/RMSE Analysis:** Evaluation metrics such as MSE and RMSE will be computed to quantify the discrepancy between the model's predictions and actual stock prices. A lower MSE/RMSE value signifies better prediction accuracy, while higher values may indicate overfitting or inadequate model performance.
* **Visualization Techniques:** Time series plots will be employed to visually compare the model's predictions against actual stock prices. This qualitative assessment provides valuable insights into the model's performance and its ability to capture underlying trends in the data.

**3. Model Robustness:**

* **Evaluation under Varying Conditions:** The robustness of the LSTM model will be tested by evaluating its performance under different market conditions and time periods. This ensures that the model can adapt to diverse market dynamics and trends.
* **Sensitivity Analysis:** Sensitivity analysis will be conducted to examine how changes in input features, hyperparameters, or training data impact the model's predictions. Understanding the model's sensitivity to various factors is crucial for enhancing its stability and reliability.
* **Adaptability:** The model's ability to adapt to evolving market dynamics and trends is essential for its practical utility. A robust and adaptable model can provide reliable predictions across different market scenarios.

**4. Discussion of Findings:**

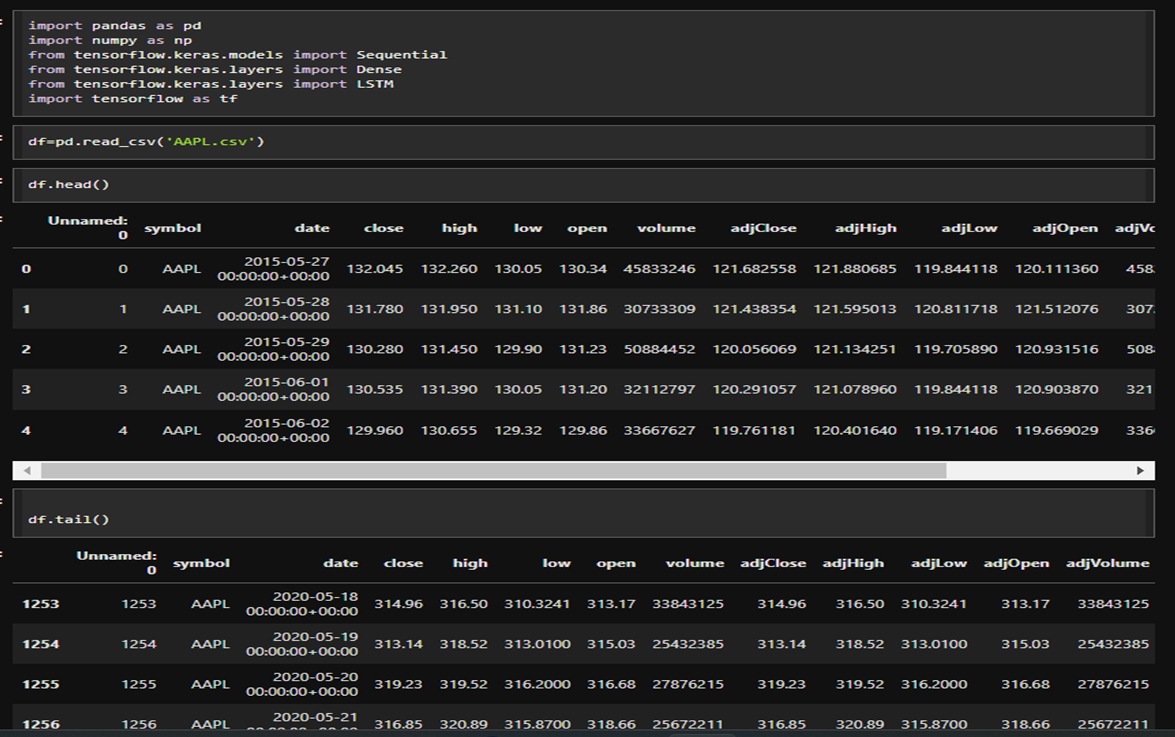
* **Insights from Predictions:** Insights gained from the model's predictions will be thoroughly analyzed to discern underlying market trends and dynamics. These insights can provide valuable information for making informed investment decisions.
* **Factors Influencing Performance:** Potential factors influencing the model's performance, such as data quality, model architecture, and training methodology, will be discussed in detail. Understanding these factors is crucial for optimizing the model's performance and addressing any limitations.
* **Comparative Analysis:** Comparative analysis may be conducted to compare the performance of the LSTM model with alternative forecasting methods or benchmarks. This comparative assessment provides valuable context for interpreting the model's performance and identifying areas for improvement.

**5. Future Directions:**

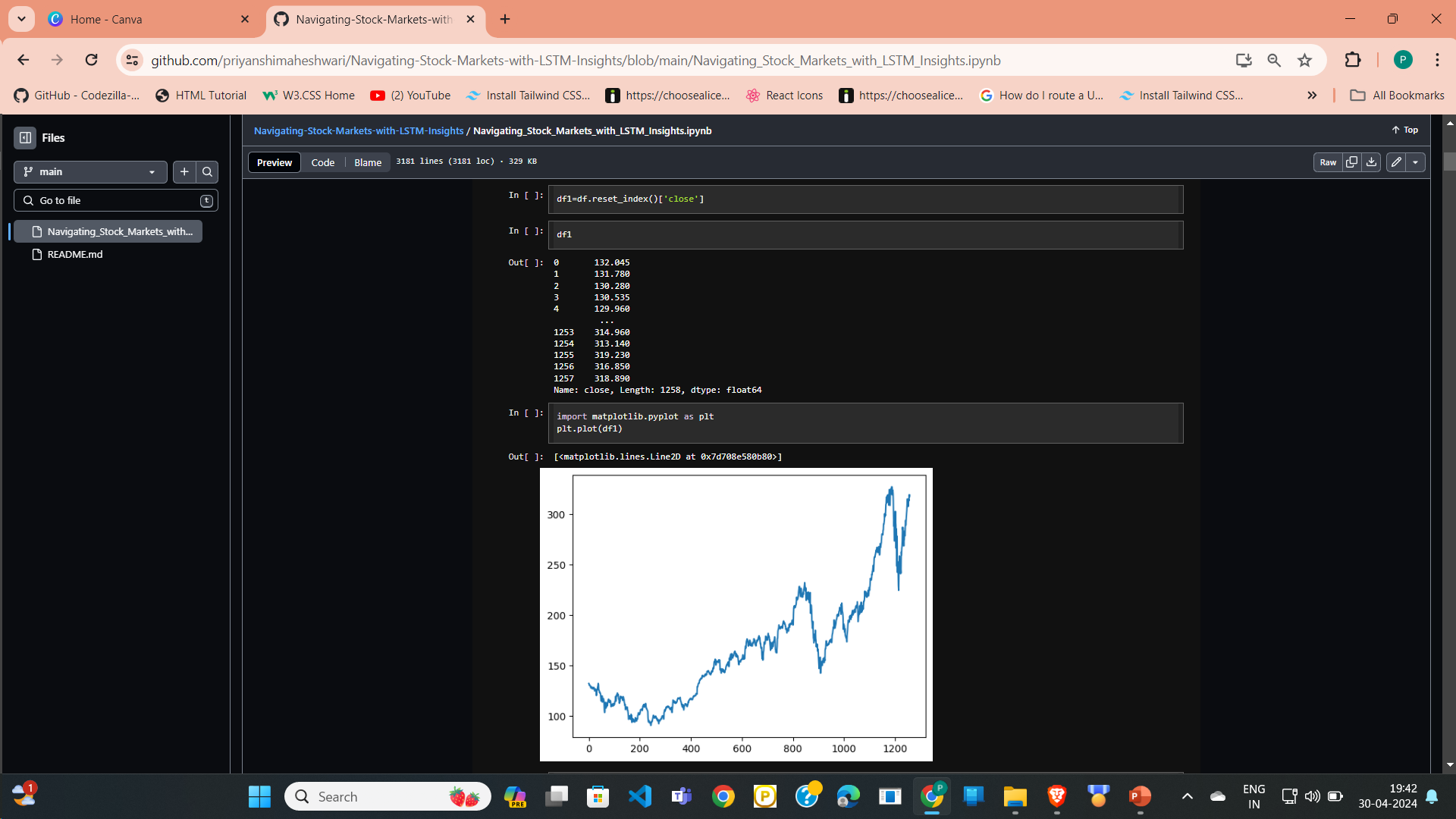
* **Areas for Further Exploration:** Based on the findings, areas for further exploration and improvement will be identified. This may include incorporating additional features, refining the model architecture, or exploring alternative algorithms to enhance prediction accuracy.
* **Continuous Iteration:** The LSTM model serves as a foundation for future enhancements and iterations. Continuous iteration and refinement are essential for developing more sophisticated and accurate stock market prediction models that can effectively adapt to changing market conditions.
* **Long-Term Vision:** The ultimate goal is to develop a robust and reliable stock market predictor that can provide actionable insights for investors and stakeholders. Continuous innovation and improvement are essential for realizing this long-term vision.

**CODING AND SCREENSHOT**

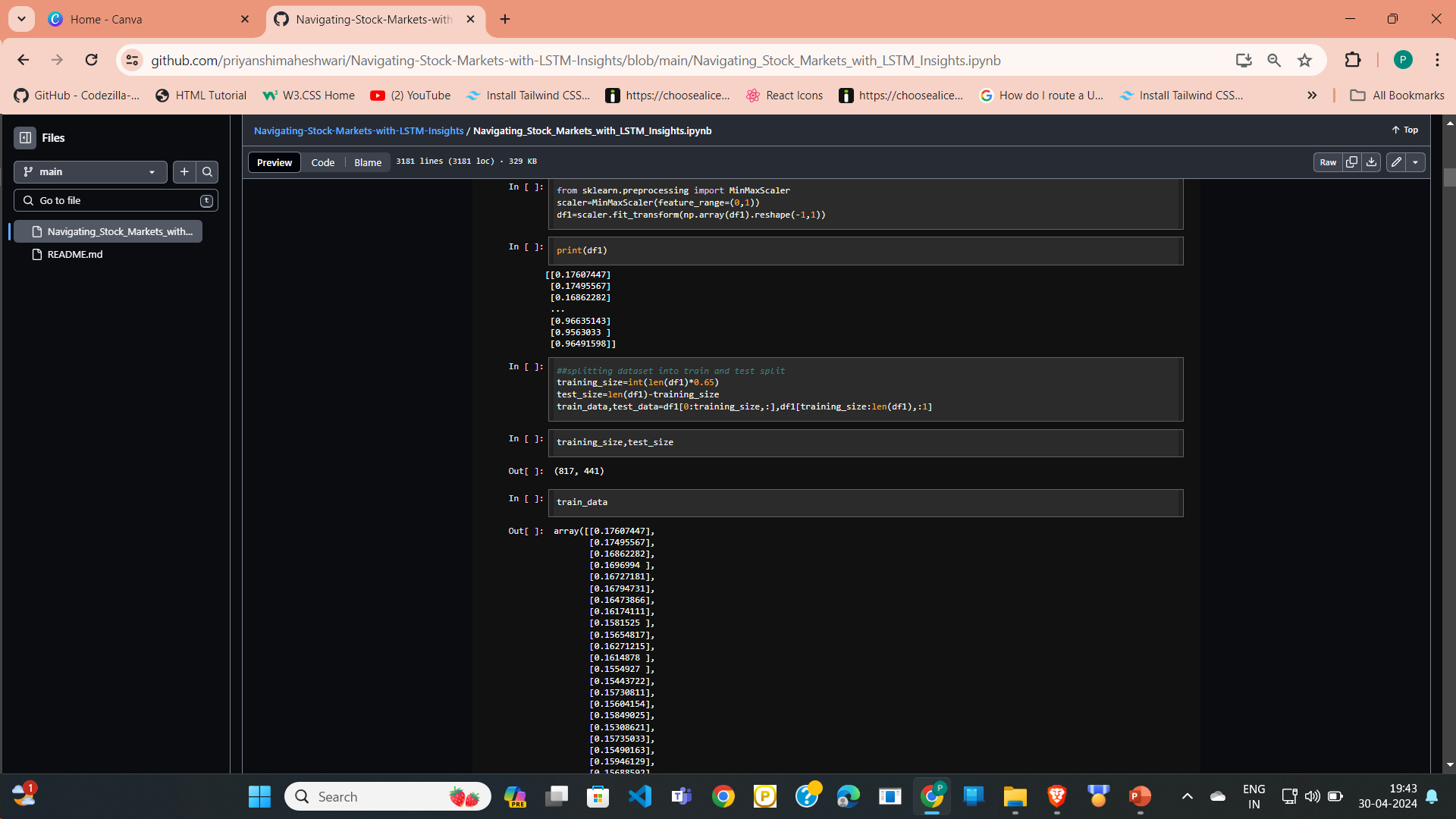
1. **Import all the libraries and data file**

****

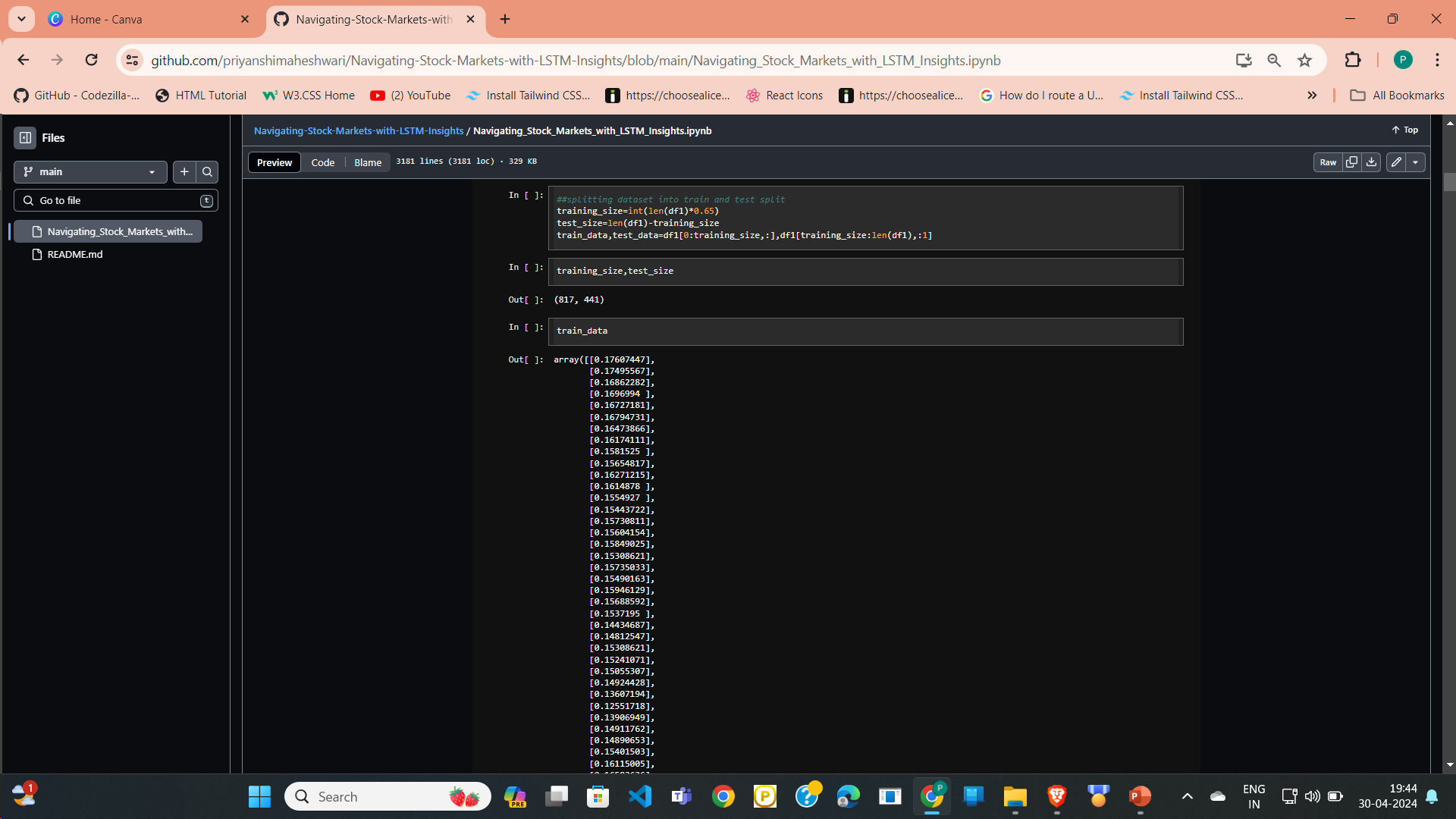
**2. Using closing values for our experimentation of time series with LSTM and plotting the graph for it**

****

1. **Normalizing the Data**



**4. *Creating X\_TRAIN AND Y\_TRAIN DATA STRUCTURES***

****

## **5. Convert an array of values into a dataset matrix and reshaping it**

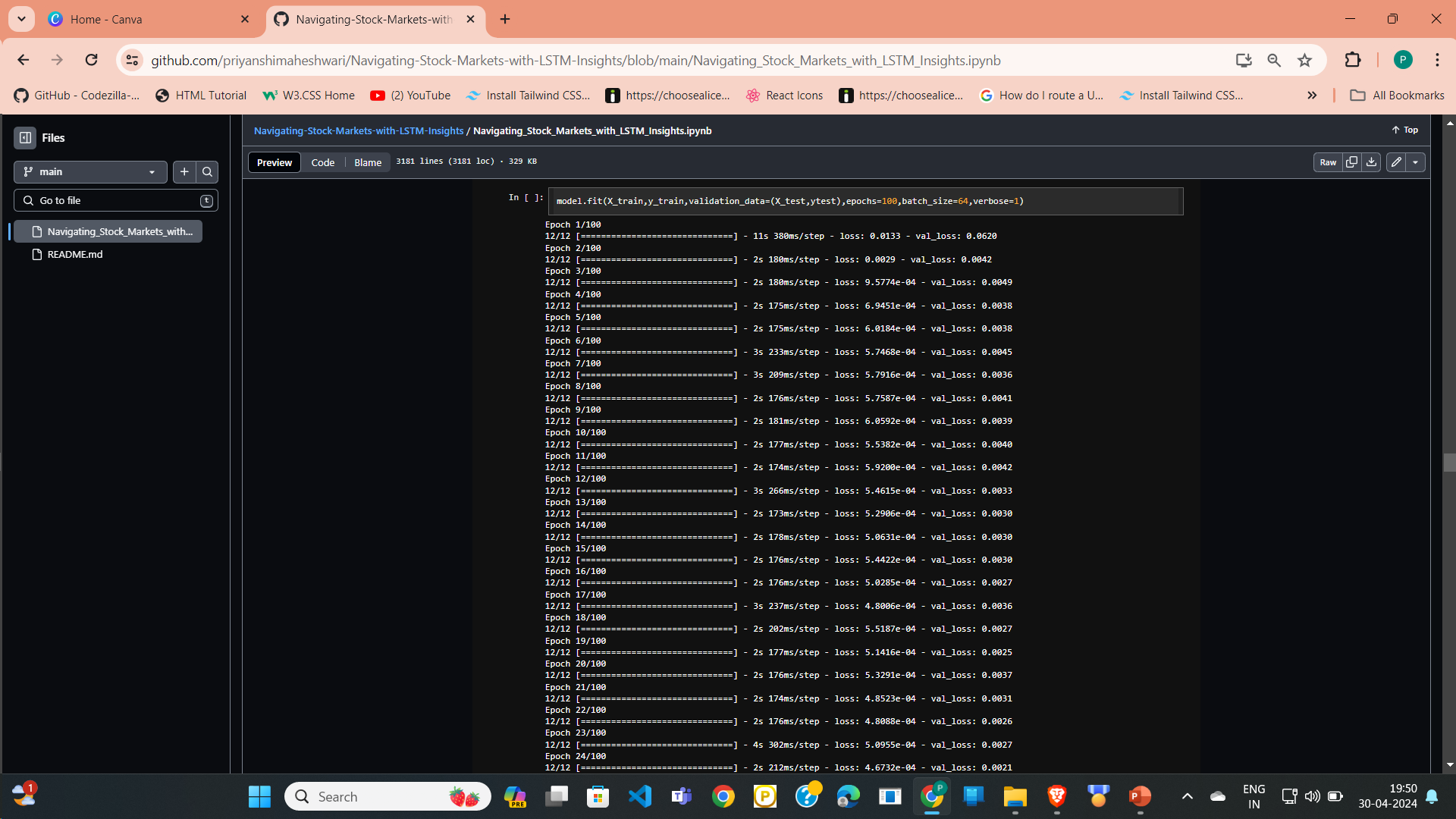
## 

## 

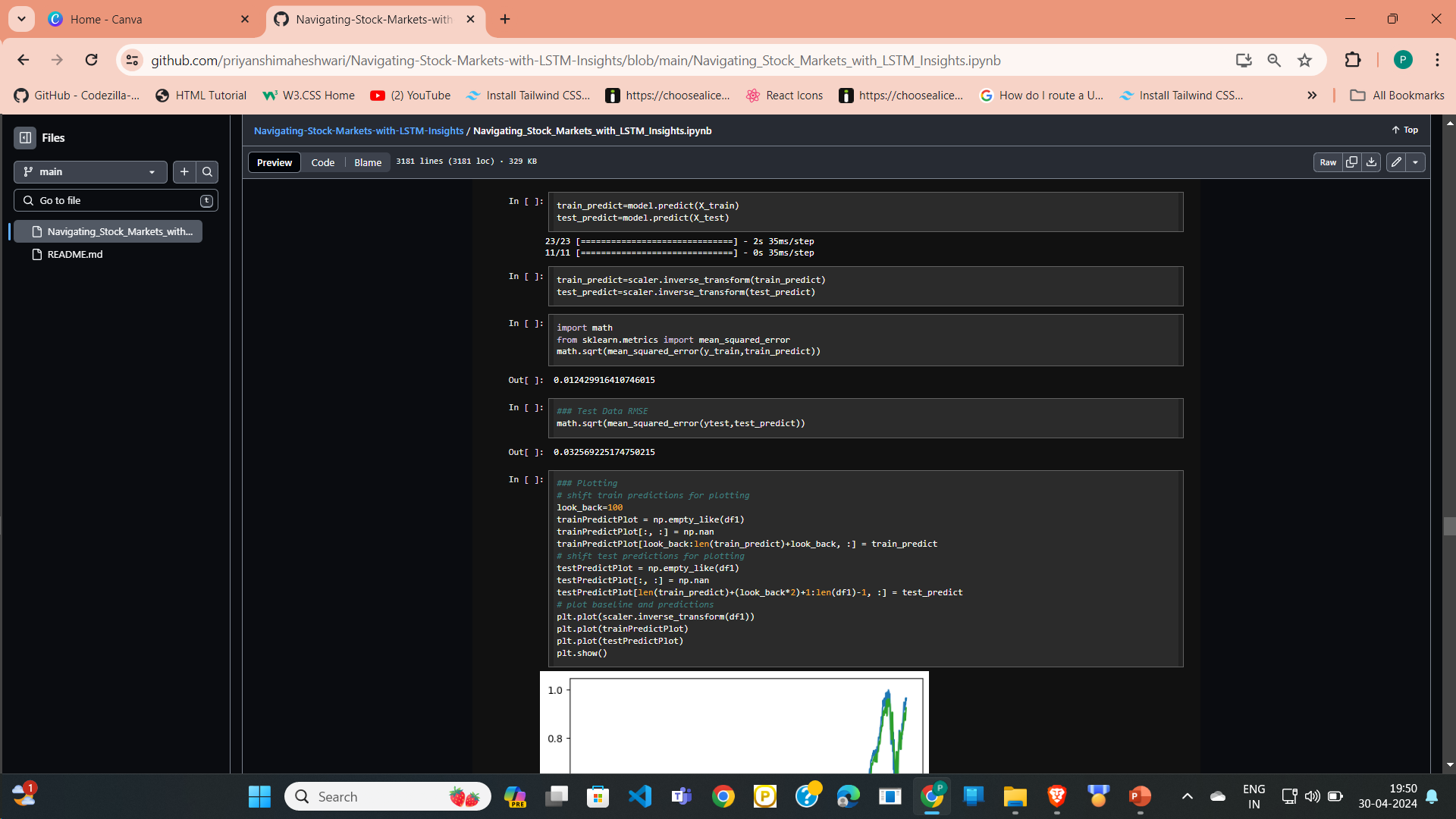
## **6. Building the lstm models with lstm and dense layer**

## 

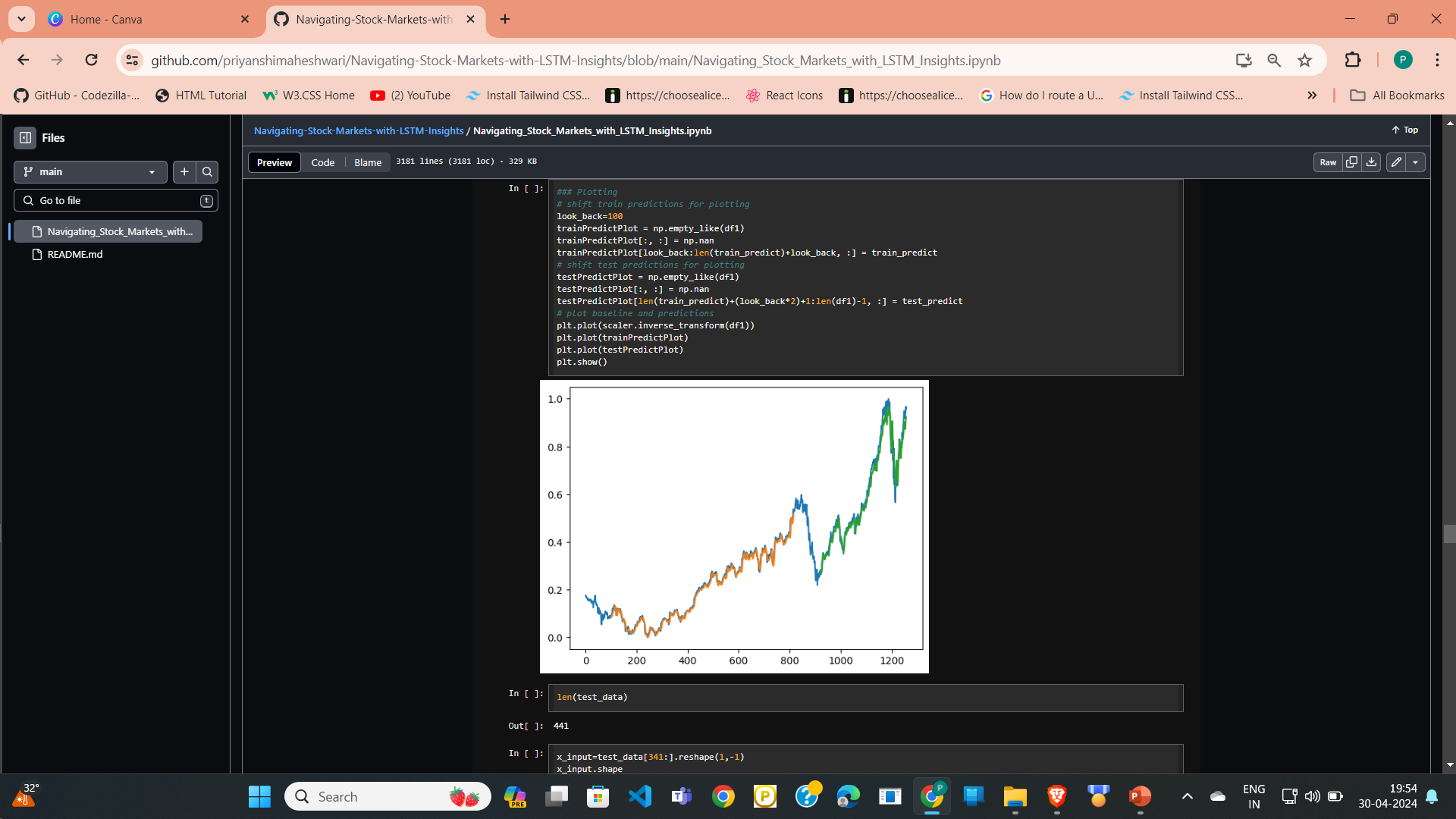
## **7. Fit the model**



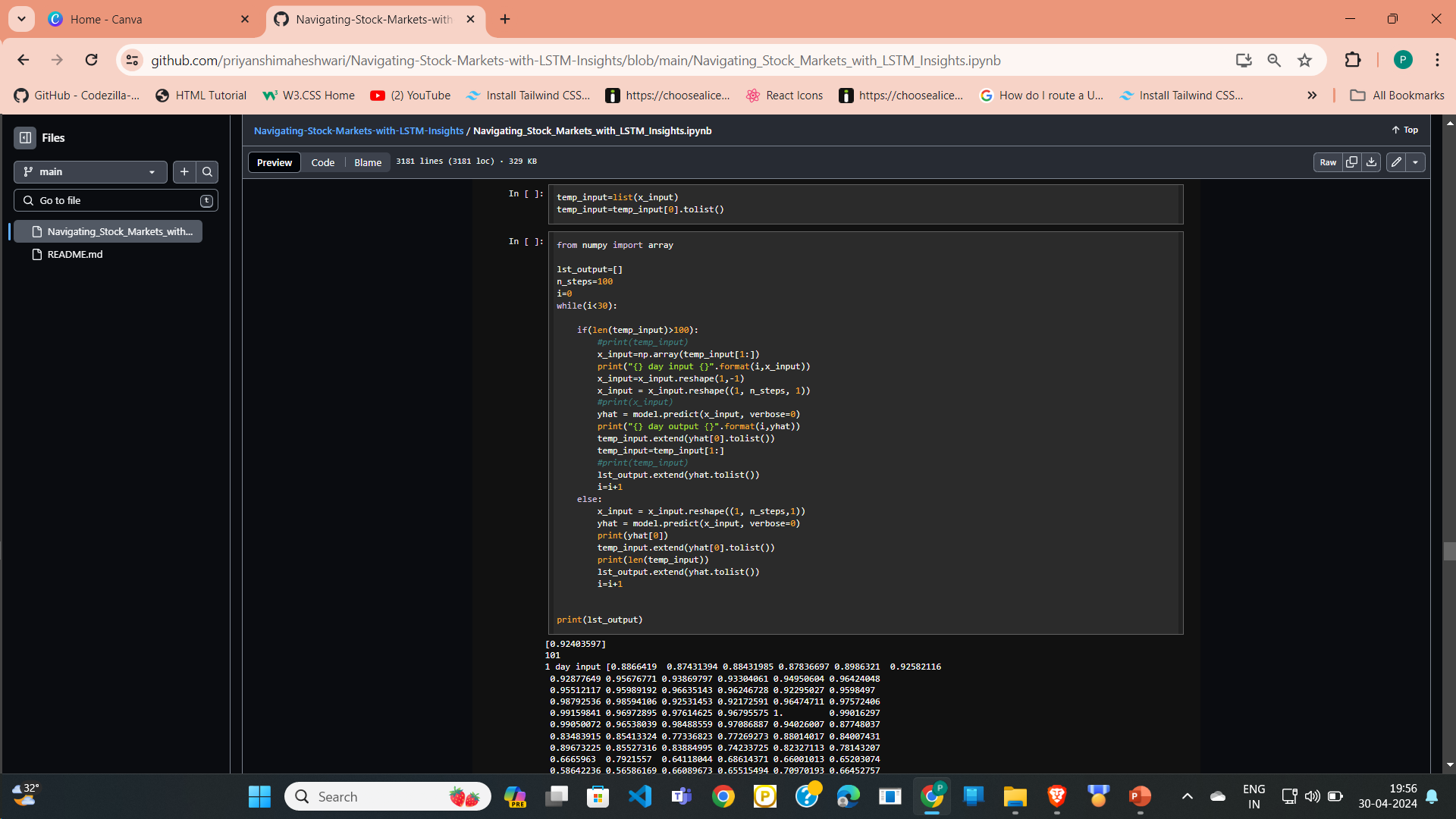
**8. Making prediction on both train and test data and Invert the scaling and calculate mse for both train and test data**



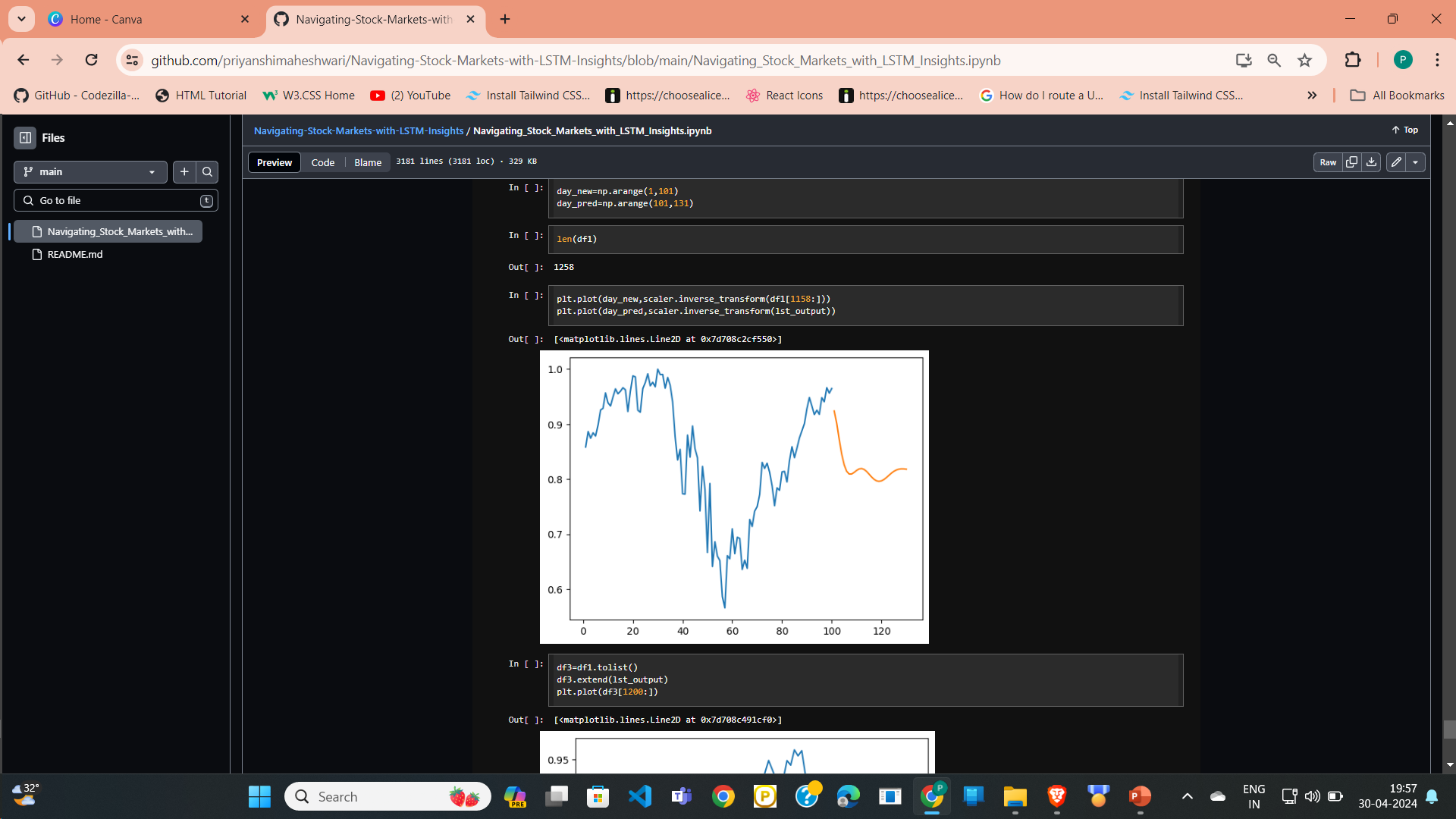
**9. Plotting the graph**

****

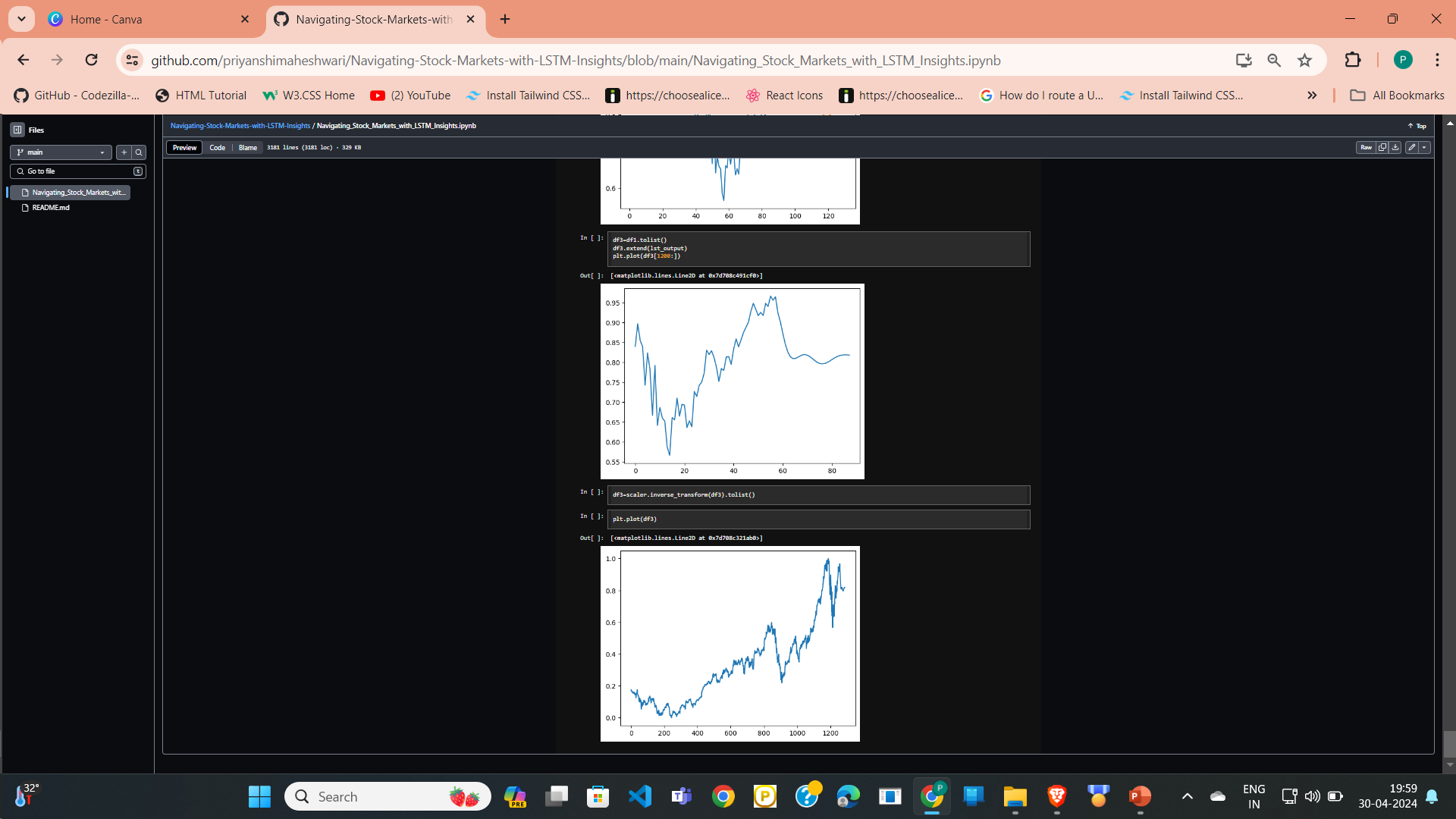
**10. Prepare input and generate prediction for next 30 days**

****

**11. Plotting graph for last 100 days and next 30 days and concating it**

****

**12. Plotting graph for last 100 days and next 30 days**

****

**CONCLUSION AND FUTURE ENHANCEMENTS**

## In conclusion, overcoming the limitations of existing methodologies in stock prediction requires a multifaceted approach that leverages alternative data sources, advanced modeling techniques, and rigorous evaluation methods. By incorporating diverse data streams, such as social media sentiment and consumer behavior, researchers can gain deeper insights into market dynamics and investor sentiment.

## Enhanced feature engineering, ensemble modeling, and deep learning architectures enable the extraction of complex patterns from financial data, leading to more accurate predictions.

**Future Enhancements**

## To further enhance the stock market predictor project and address its limitations, several future enhancements can be considered:

## Feature Engineering: Explore additional input features such as sentiment analysis of news articles, social media sentiment, economic indicators, or technical indicators to capture more nuanced market dynamics.

## Model Ensemble: Implement ensemble learning techniques to combine predictions from multiple models, including LSTM, convolutional neural networks (CNNs), or traditional statistical models, to improve predictive accuracy and robustness.

**REFERENCES**

1. https://colah.github.io/posts/2015-08-Understanding-LSTMs/
2. https://[www.sas.com/en\_us/insights/analytics/machine-](http://www.sas.com/en_us/insights/analytics/machine-) learning.html
3. https://groww.in/us-stocks/googl
4. https://[www.investopedia.com/terms/d/deep-learning.asp](http://www.investopedia.com/terms/d/deep-learning.asp)
5. https://[www.nasdaq.com/market-activity/stocks/google](http://www.nasdaq.com/market-activity/stocks/google)