**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 3](#_TOC_250002)

TABLE OF CONTENTS 4

[LIST OF FIGURES 5](#_TOC_250001)

[ABBREVIATIONS 5](#_TOC_250000)

###### INTRODUCTION

###### LITERATURE SURVEY

###### SYSTEM ARCHITECTURE AND DESIGN

###### METHODOLOGY

###### 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

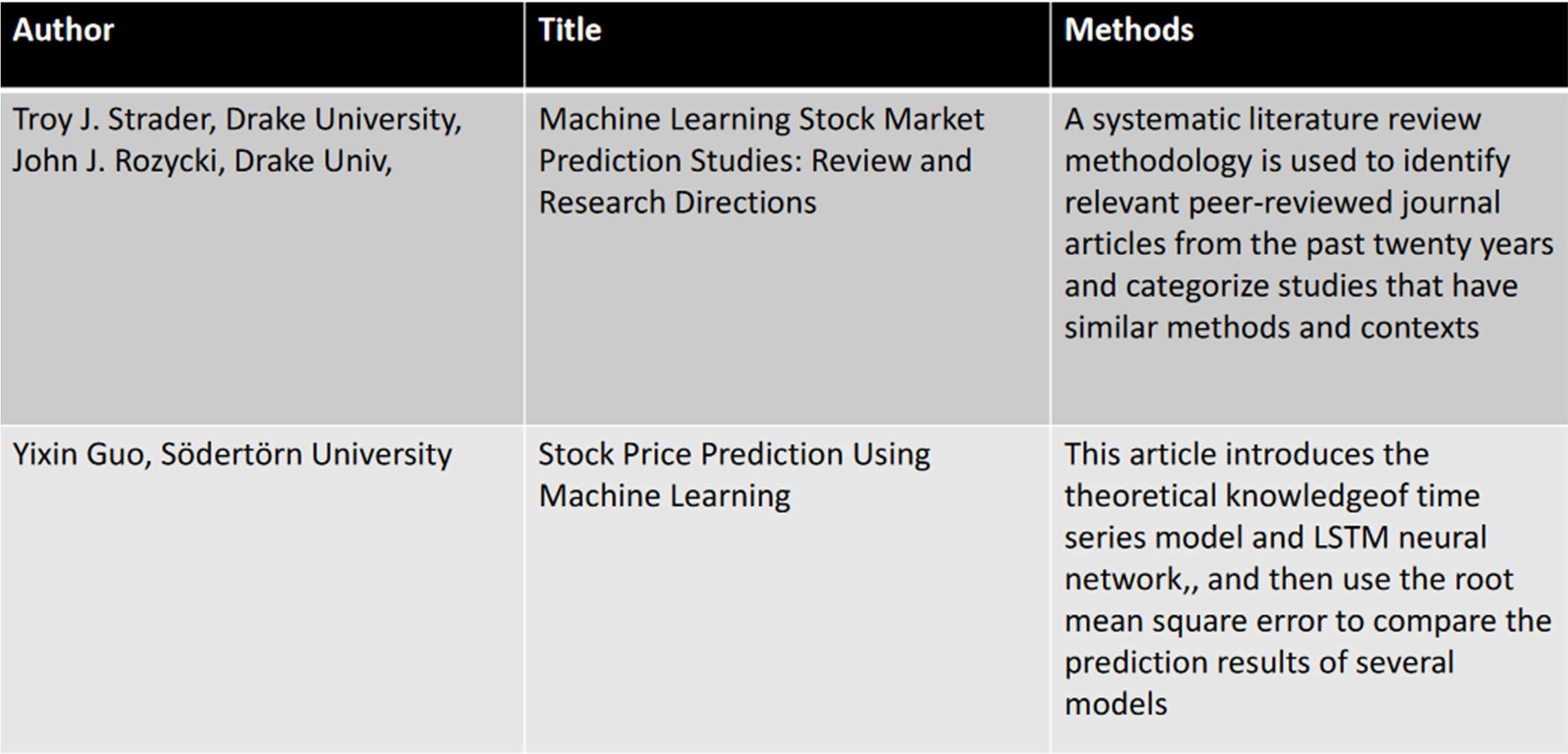
**MSE:** Mean Square error

**INTRODUCTION**

##### In the ever-evolving landscape of financial markets, the ability to predict stock movements accurately has long been the Holy Grail for investors and analysts alike. With the advent of Artificial Intelligence (AI) and machine learning technologies, the dream of developing a reliable stock market predictor has come closer to reality than ever before.

##### This project aims to harness the power of AI to build a robust and efficient stock market predictor that not only analyzes historical data but also adapts to real-time market dynamics. By leveraging advanced algorithms and data analytics techniques, our endeavor seeks to unlock valuable insights from vast volumes of financial data, enabling investors to make informed decisions and capitalize on market opportunities with greater precision.

##### In this introduction, we will delve into the significance of predicting stock market trends, explore the challenges inherent in traditional methods, and outline the objectives and methodology of our AI-driven approach. Additionally, we will highlight the potential impact of such a predictive tool on financial markets and the broader economy, paving the way for a new era of augmented intelligence in finance.



**LITERATURE REVIEW**

**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 Preparation:**

* **Gather Data:** Collect historical stock market data from reliable sources (APIs, databases, scraping).
* **Clean and Preprocess:** Address missing values, outliers, and inconsistencies. Normalize data for consistent scaling.
* **Feature Engineering:** Create sequences/windows from the time-series data for LSTM training.

**2. Building the LSTM Model:**

* **Design the Architecture:** Define the number of LSTM layers, hidden units, activation functions, and input/output structure.
* **Train, Split, and Evaluate:** Train the model on historical data. Split data into training, validation (for hyperparameter tuning), and testing sets. Evaluate performance with metrics like MAE, MSE, or RMSE.

**3. Refining and Improvement:**

* **Optimize Hyperparameters:** Adjust settings like learning rate, batch size, and dropout rate to improve accuracy.
* **Prevent Overfitting:** Implement techniques like early stopping or regularization to avoid overfitting the training data.
* **Sensitivity Analysis:** Assess how the model reacts to changes in features, hyperparameters, or market conditions.

**4. Deployment and Monitoring:**

* **Production Environment:** Deploy the trained model for real-world use (standalone app, API, trading platform).
* **Performance Tracking:** Monitor the model's performance in real-time, identifying anomalies and potential issues.

**5. Continuous Learning:**

* **Feedback Loop:** Gather user and stakeholder feedback 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.

**METHODOLOGY**

**Data Collection and Preprocessing**

1. **Data Collection:**
   * Gather historical stock price data from reliable sources like financial APIs databases of Apple stock from Internet
2. **Data Cleaning:**
   * Identify and handle missing values in the data. This might involve removing data points, imputation techniques, or carrying forward values depending on the nature of the missing data.
3. **Data Normalization:**
   * Normalize the data using Minimum scaling to bring all values between 0 and 1. This helps the LSTM model learn more effectively.

**Data Splitting**

Divide the collected data into two sets: training and testing. Split data in 65% for training and 35% for testing. This allows the model to learn from the majority of the data and evaluate its performance on unseen data.

**Sequence Creation**

1. **Function Definition:**
   * Create a function to generate sequences of data for the LSTM model.
2. **Sequence Definition:**
   * Each sequence should contain a fixed number of past data points (e.g., 100 days of closing prices) as input and the corresponding next day's price as the target output. This allows the LSTM to learn temporal dependencies in the data.

**Model Building**

1. **Model Initialization:**
   * Initialize a sequential model using a deep learning library like TensorFlow or Keras.
2. **LSTM Layers:**
   * Add LSTM layers to the model. These layers capture long-term dependencies in the data. The number of layers and units can be optimized through experimentation.
3. **Output Layer:**
   * Add a dense output layer with one neuron to predict the next day's closing price.

**Model Training**

1. **Compilation:**
   * Compile the model by specifying a loss function MSE that measures the difference between predicted and actual prices and an Adam optimizer that updates the model weights to minimize the loss.
2. **Training:**
   * Train the model on the training dataset for a specific no. of epochs (iterations).
3. **Validation:**
   * Evaluate the model's performance on the testing dataset after each epoch. This helps to avoid overfitting, where the model performs well on the training data but poorly on unseen data.

**Prediction**

1. **Prediction on Training and Testing Data:**
   * After training, use the trained model to predict stock prices for both the training and testing datasets.
2. **Inverse Scaling:**
   * Apply the inverse Min-Max scaling to the predicted values to obtain the actual stock price estimates.

**Evaluation**

1. **RMSE Calculation:**
   * Calculate the Root Mean Squared Error (RMSE) for both the training and testing datasets. RMSE measures the average magnitude of the errors between predicted and actual prices. A lower RMSE indicates better model performance.

**Visualization**

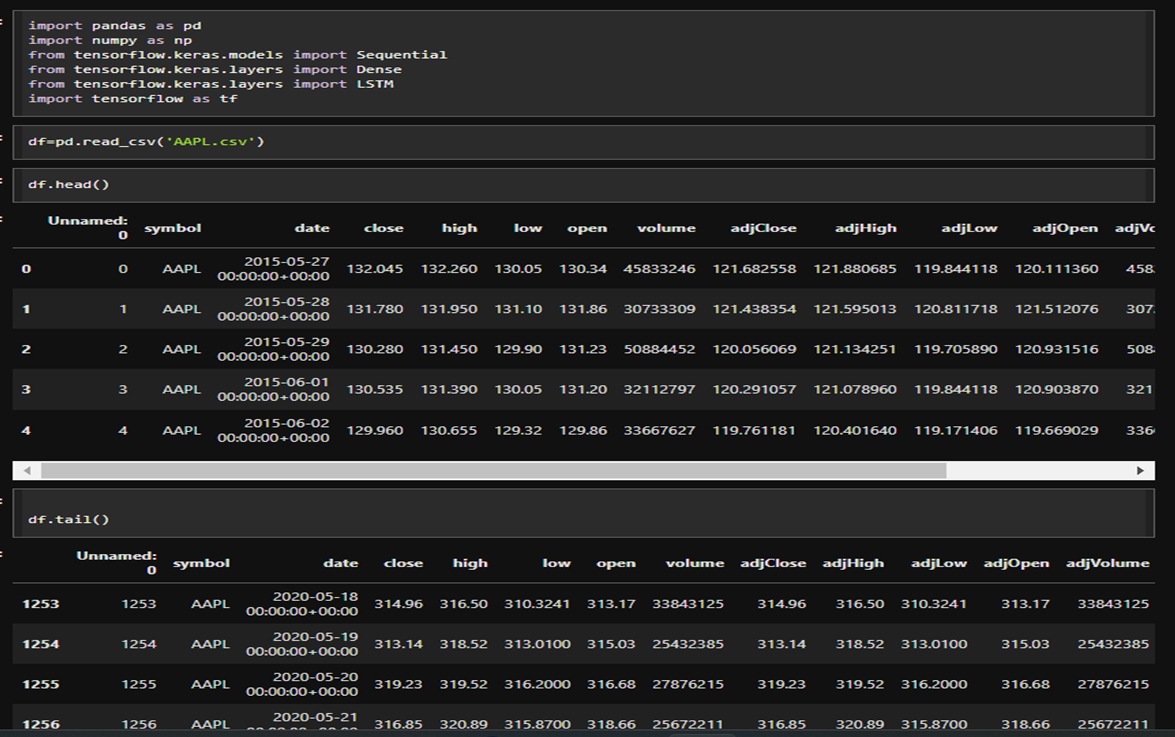
1. **Price Plots:**
   * Plot the actual stock prices alongside the predicted prices for both the training and testing sets. This allows for a visual assessment of the model's ability to capture trends and predict future prices.

**Future Prediction**

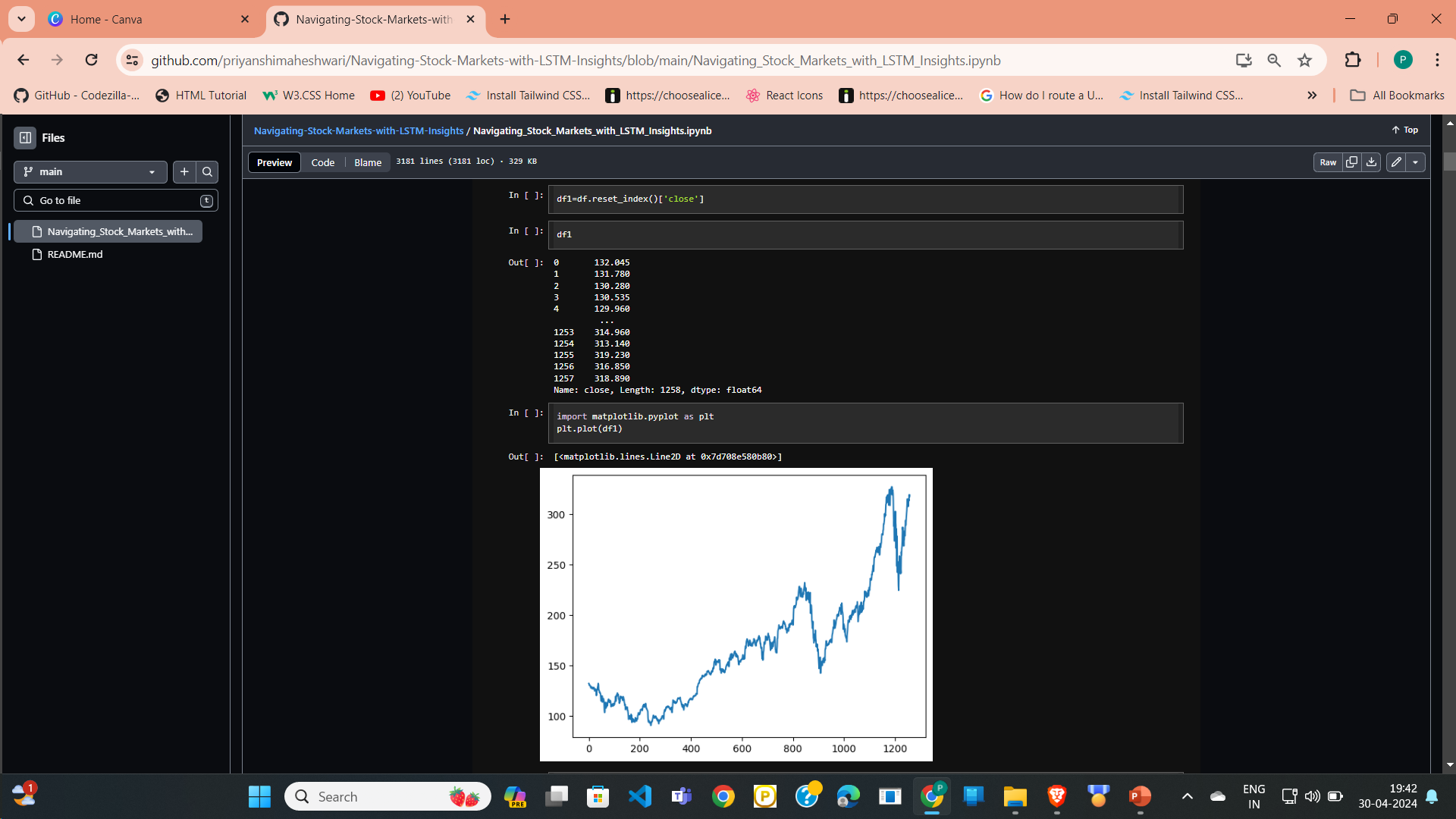
1. **Rolling Window Approach:**
   * Utilize a rolling window approach to predict future prices. This involves feeding the model with the most recent sequence of data points of last 100 days) and predicting the next 30 day's price.

**CODING AND SCREENSHOT**

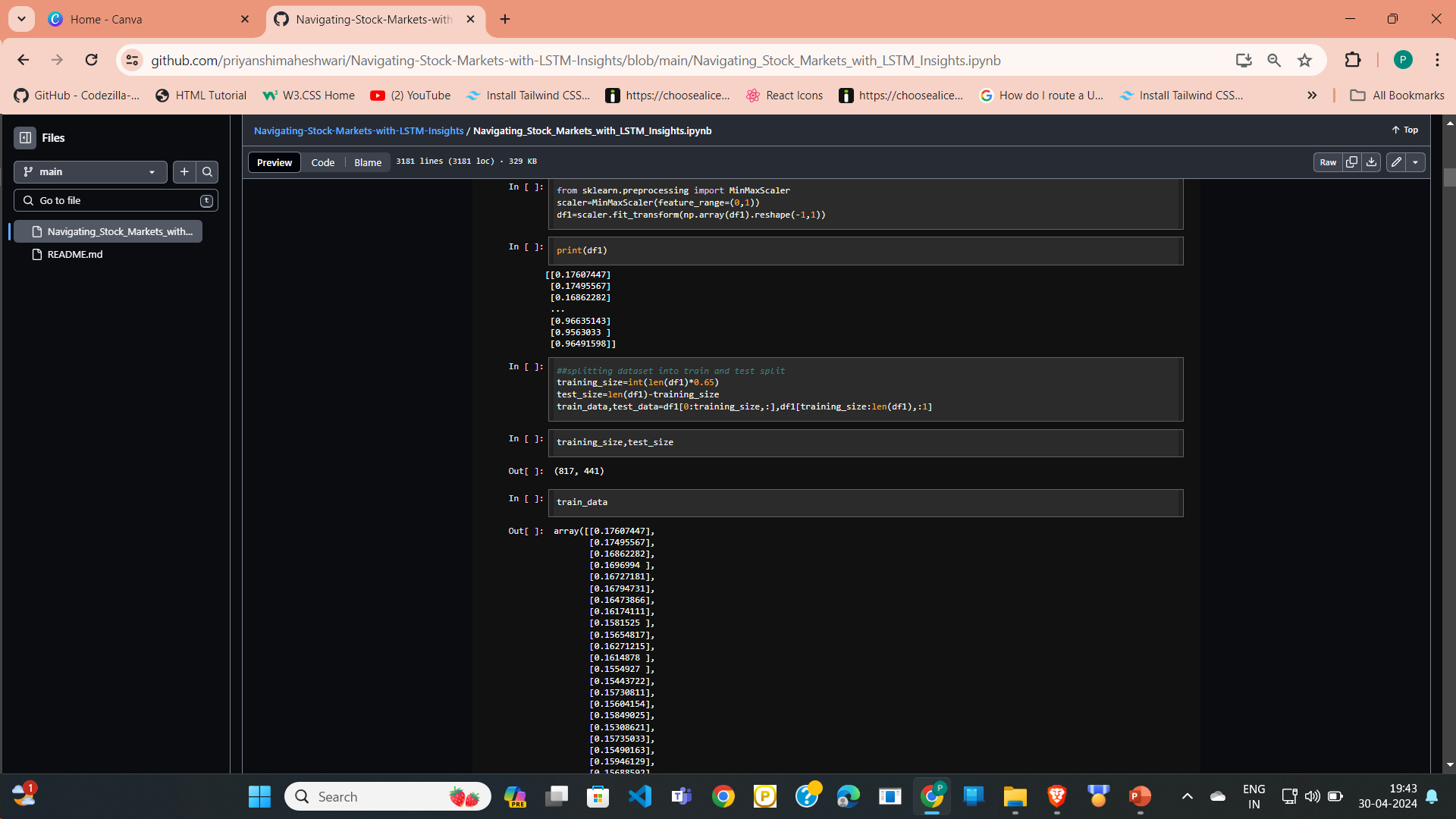
1. **Import all the libraries and data file**

****

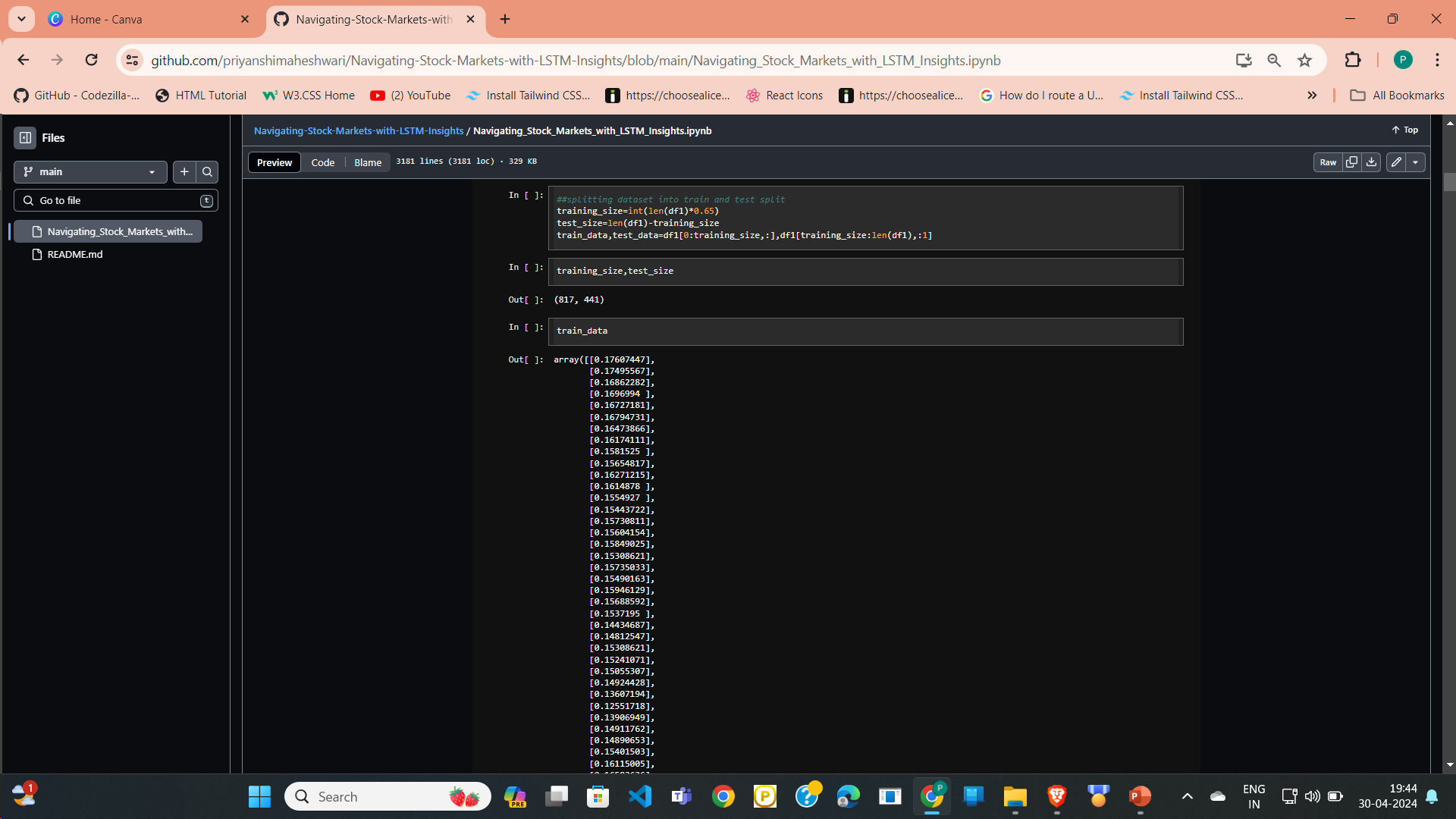
**2. Using closing values for our experimentation of time series with LSTM and plotting the graph for it**

****

* 1. **Normalizing the Data**



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

****

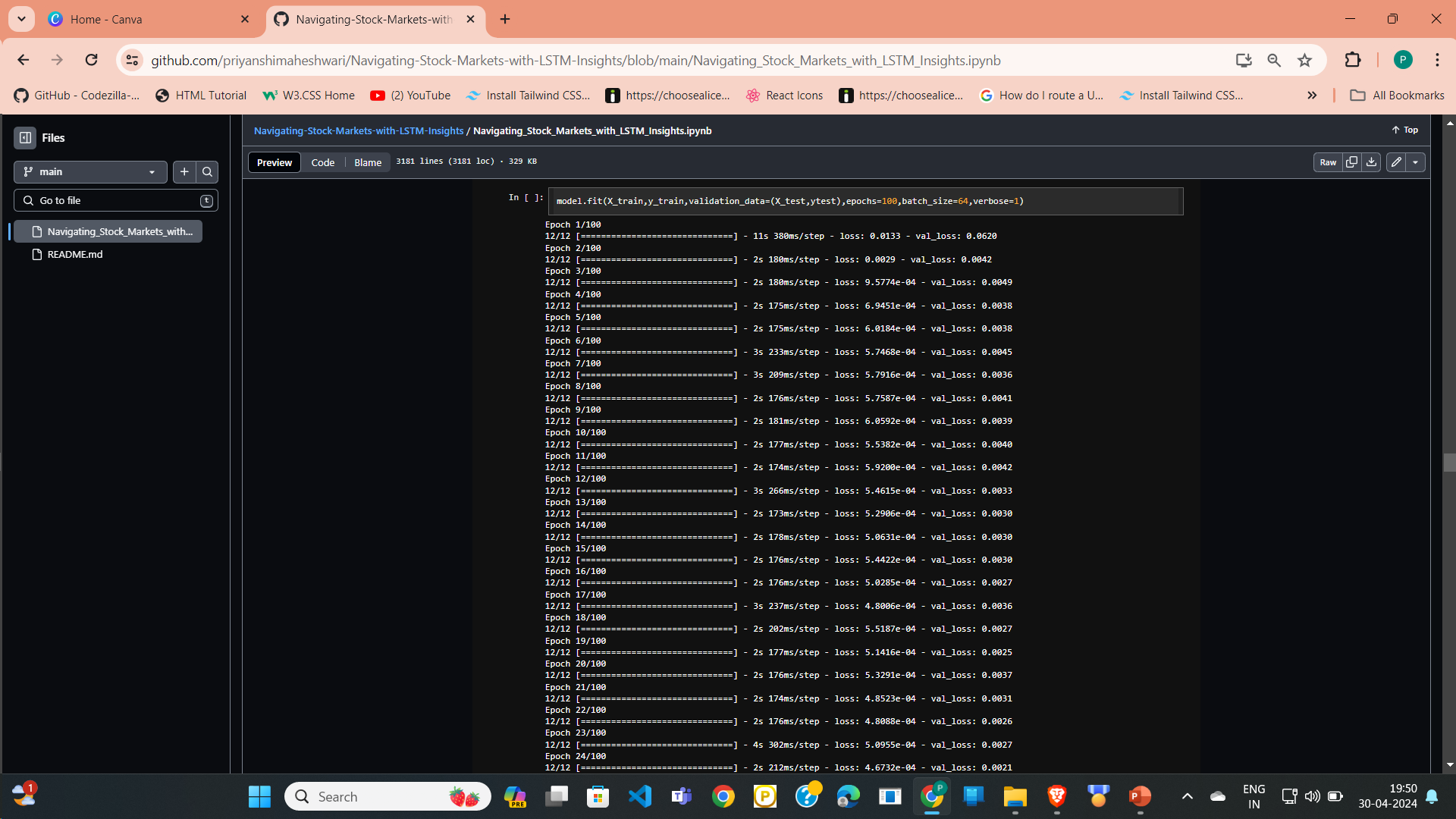
## **5. Convert an array of values into a dataset matrix and reshaping it**

## 

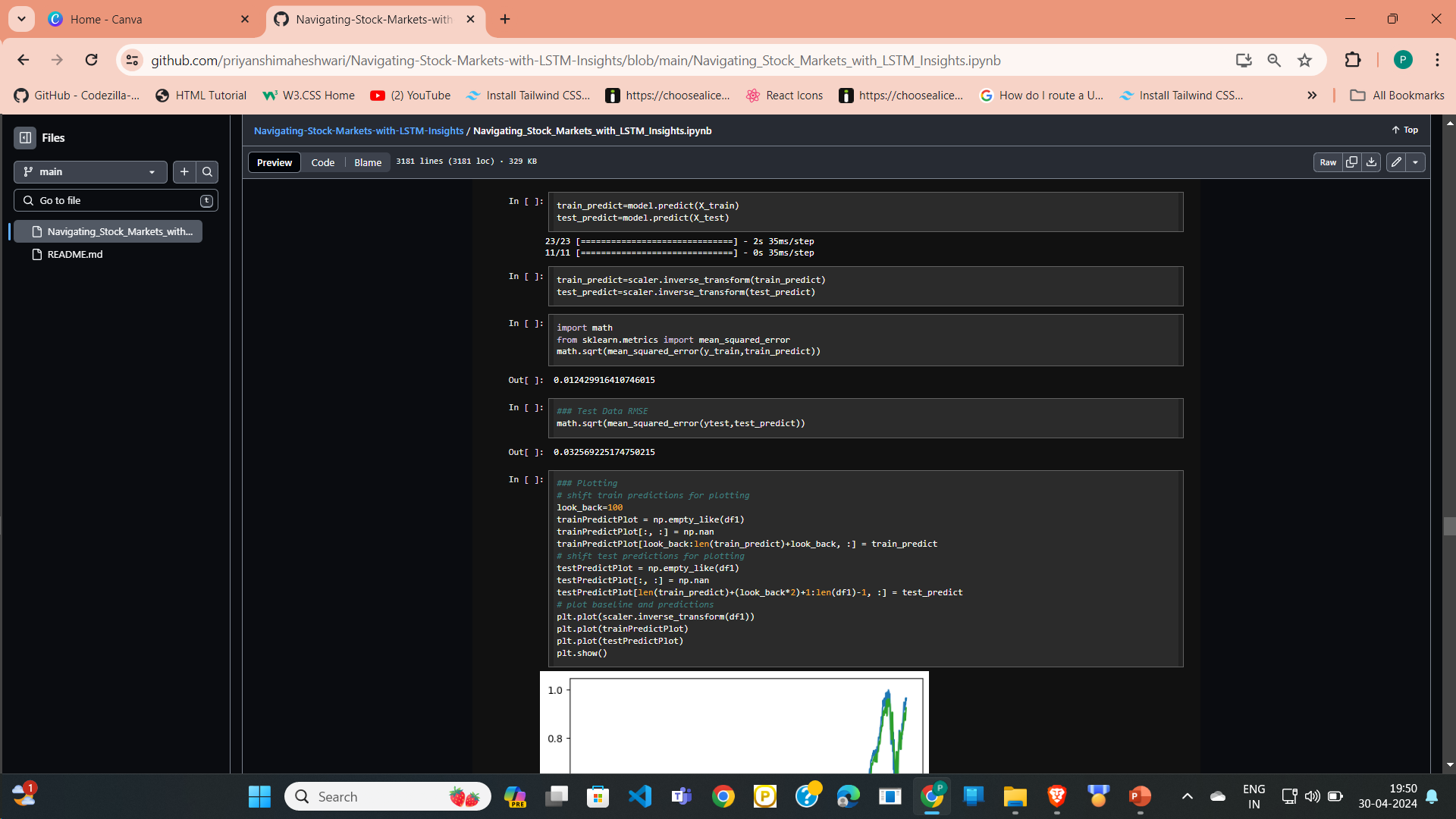
## **6. Building the Lstm models with Lstm and dense layer**

## 

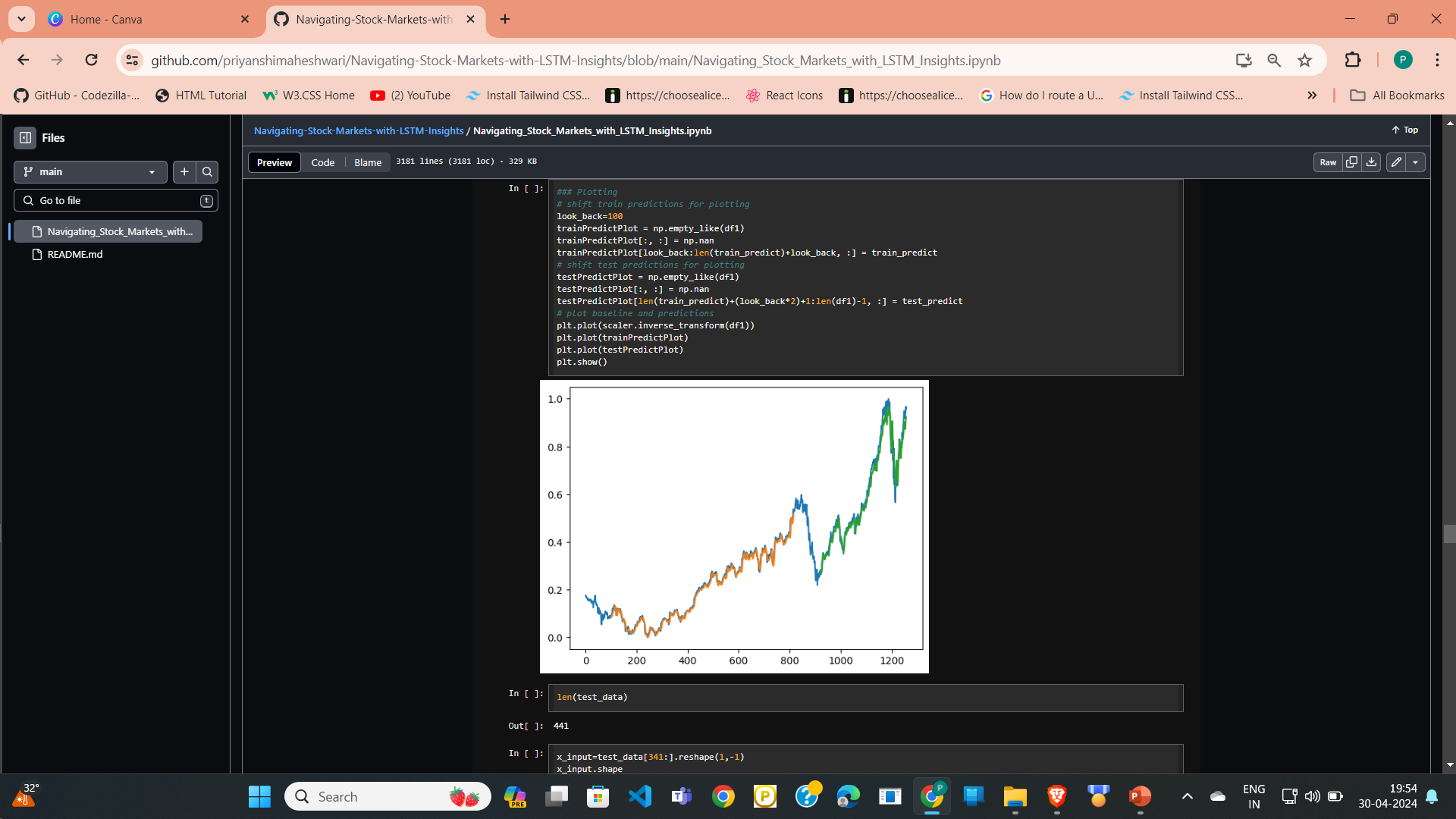
## **7. Model fitting**



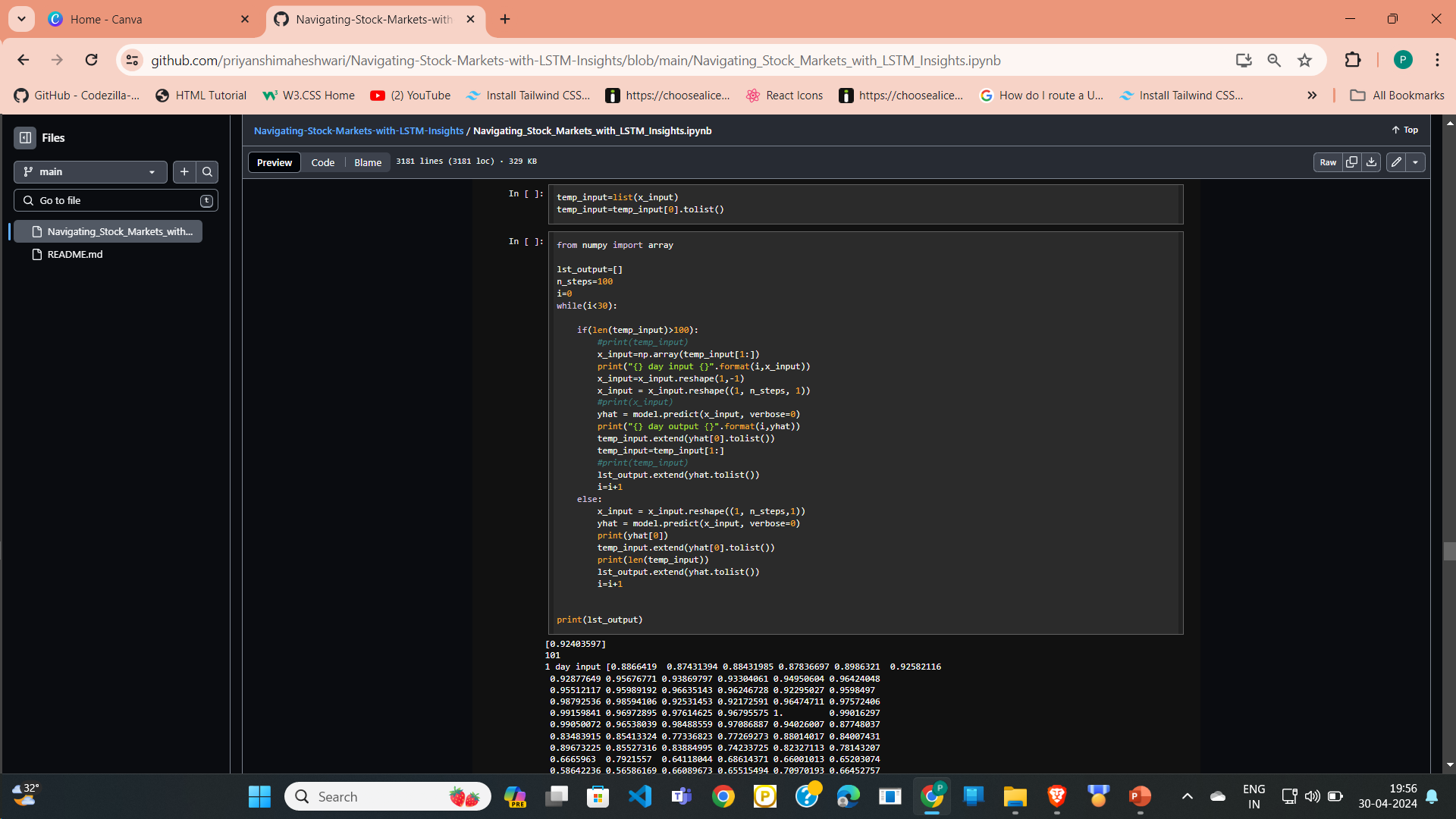
**8. Making prediction on both train and test data and Invert the scaling and calculate mse for both train and test data**



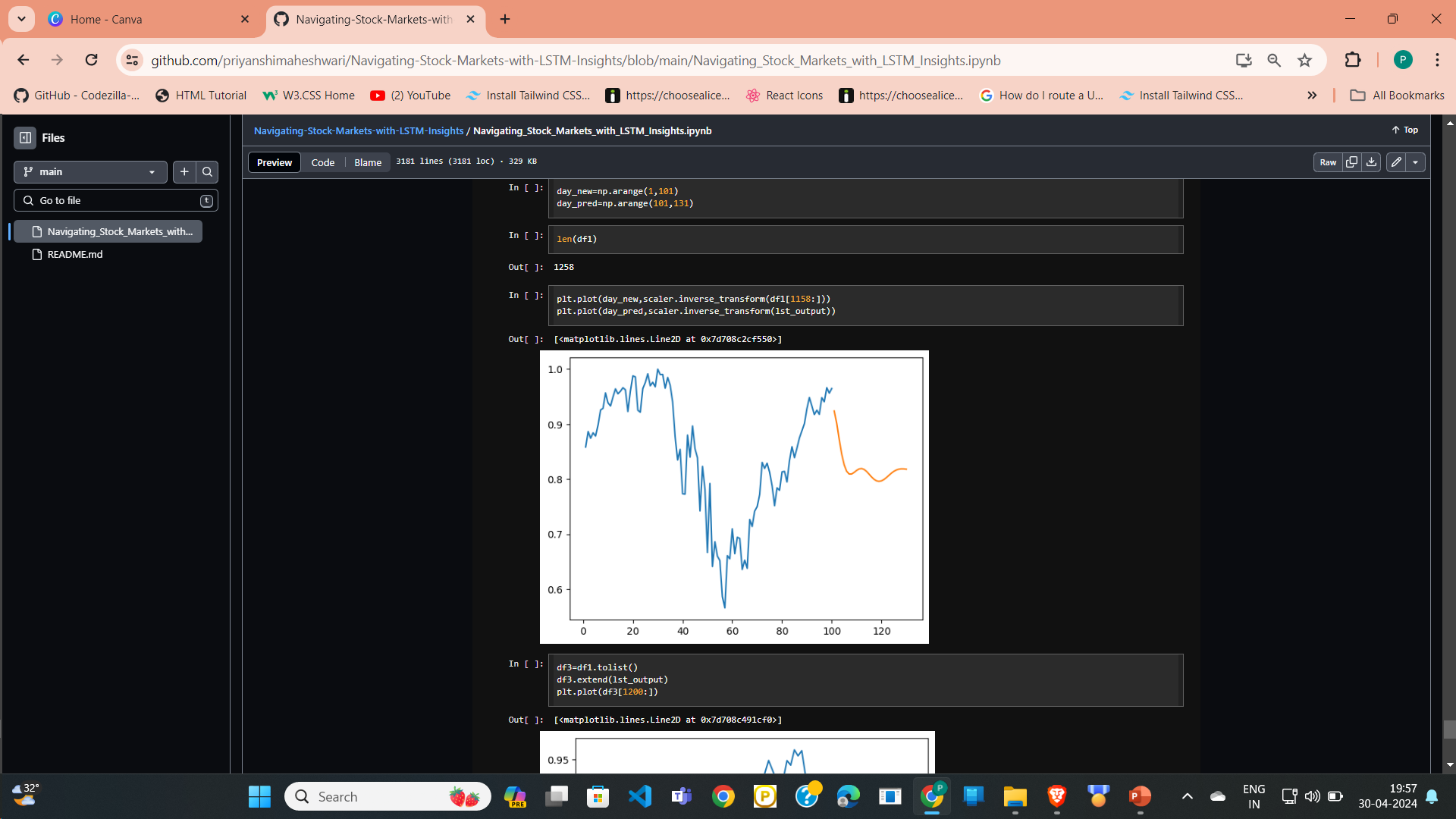
**9. Plotting the graph**

****

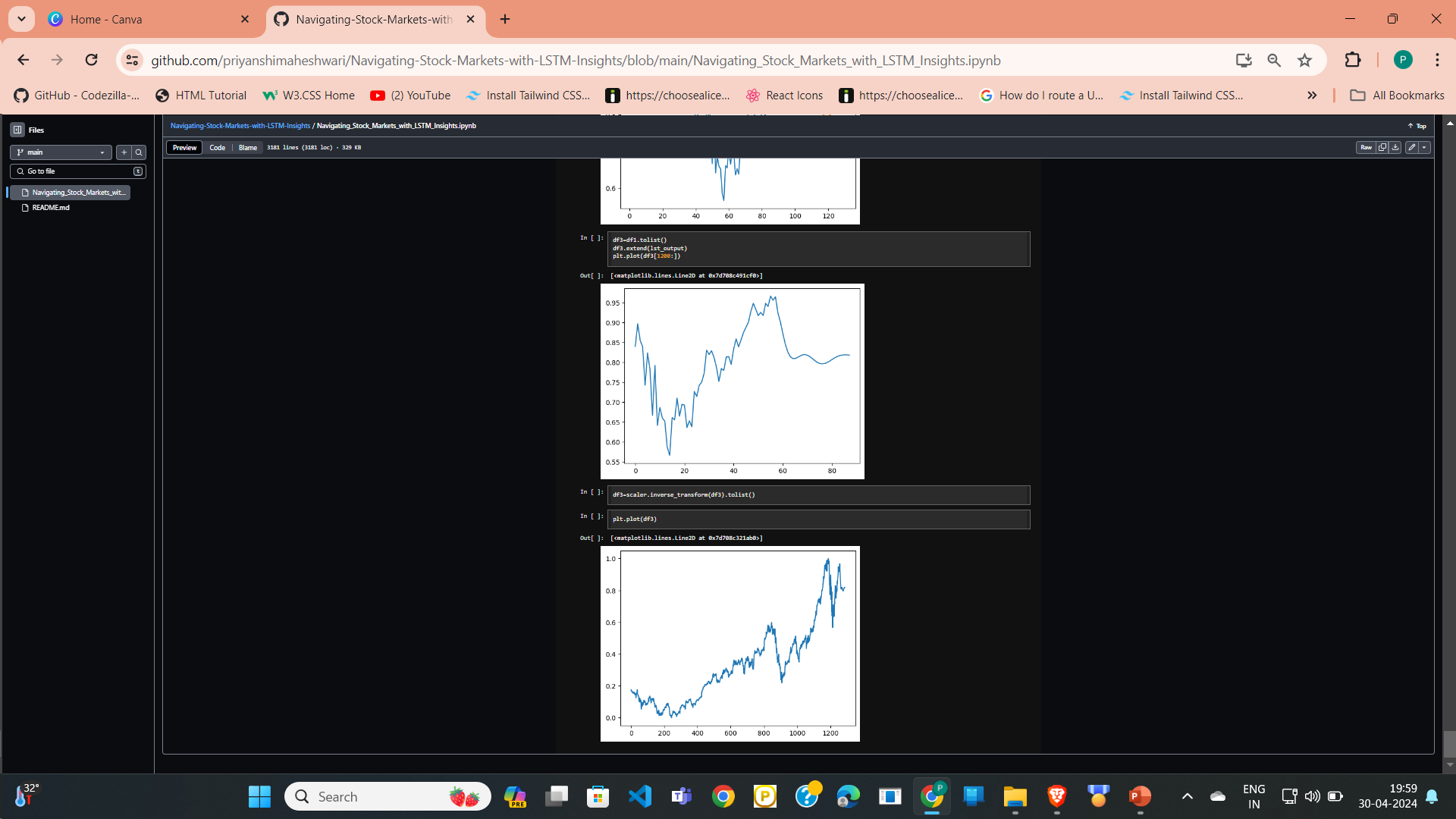
**10. Prepare input and generate prediction for next 30 days**

****

**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/apple
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/apple](http://www.nasdaq.com/market-activity/stocks/apple)
6. TensorFlow and Keras Documentation: For detailed information on model building, training, and evaluation.