**STOCK PRICE PREDICTION – LSTM**

**Table of contents:**

* **Introduction – Stock price Prediction**
* **What Is LSTM?**
* **Why do we used LSTM for stock price prediction**
* **Practical Implementation of Stock Price Prediction by using LSTM**

Predicting how the stock market will perform is one of the most difficult things to do and an uncertain task. There are a lot of methods and tools used for stock market prediction. There are so many factors involved in the prediction – physical factors vs. psychological, rational and irrational behavior, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy. Over the years, various machine learning techniques have been used in stock market prediction, but with the increased amount of data and expectation of more accurate prediction, the deep learning models are being used nowadays which have proven their advantage over traditional machine learning methods in terms of accuracy and speed of prediction.

In this article, we will discuss the **Long-Short-Term Memory (LSTM)** one of the popular deep learning models, used in stock market prediction.



Before moving directly to explaining the Stock Price Prediction by using LSTM first we will discuss What is LSTM? Why do we use LSTM for stock price prediction?

**What is LSTM?**

LSTM is a special kind of RNN, capable of learning **long-term dependencies.** Remembering information for long periods is practically their default behavior, not something they struggle to learn. It has an advantage over traditional neural networks due to its capability to process the entire **sequence of data.**

LSTM is its analogy with **conveyor belts!**

Industries use them to move products around for different processes. LSTMs use this mechanism to move information around. We may have some addition, modification, or removal of information as it flows through the different layers, just like a product may be molded, painted, or packed while it is on a conveyor belt.

**Why do we use LSTM for Stock Price Prediction?**

We have two algorithms for predicting the stock price i.e Time Series Forecasting by using ARIMA, SARIMAX, and deep learning model i.e LSTM but here the question is why we use LSTM?

Because ARIMA has predicted the stock price in the short term period if you want to predict the stock price based on the long term then go with LSTM. Because LSTM stores the data for the long term. LSTM gives you good accuracy and prediction rather than ARIMA for predicting the price of the stock.

LSTMs are very powerful in sequence prediction problems because they’re able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price.

**Stock Prediction**

In thisarticle, we will use the IBM historical data to predict the future price stocks using the LSTM Recurrent Neural Network. This data set contains 3270 observations with 6 attributes.

* The columns *Open* and *Close* represent the starting and final price at which the stock is traded on a particular day.
* *High*, *Low* represent the maximum and minimum price of the share for the day.
* *Volume is* the number of shares bought or sold in the day

As stock market analysis has been divided into two parts:

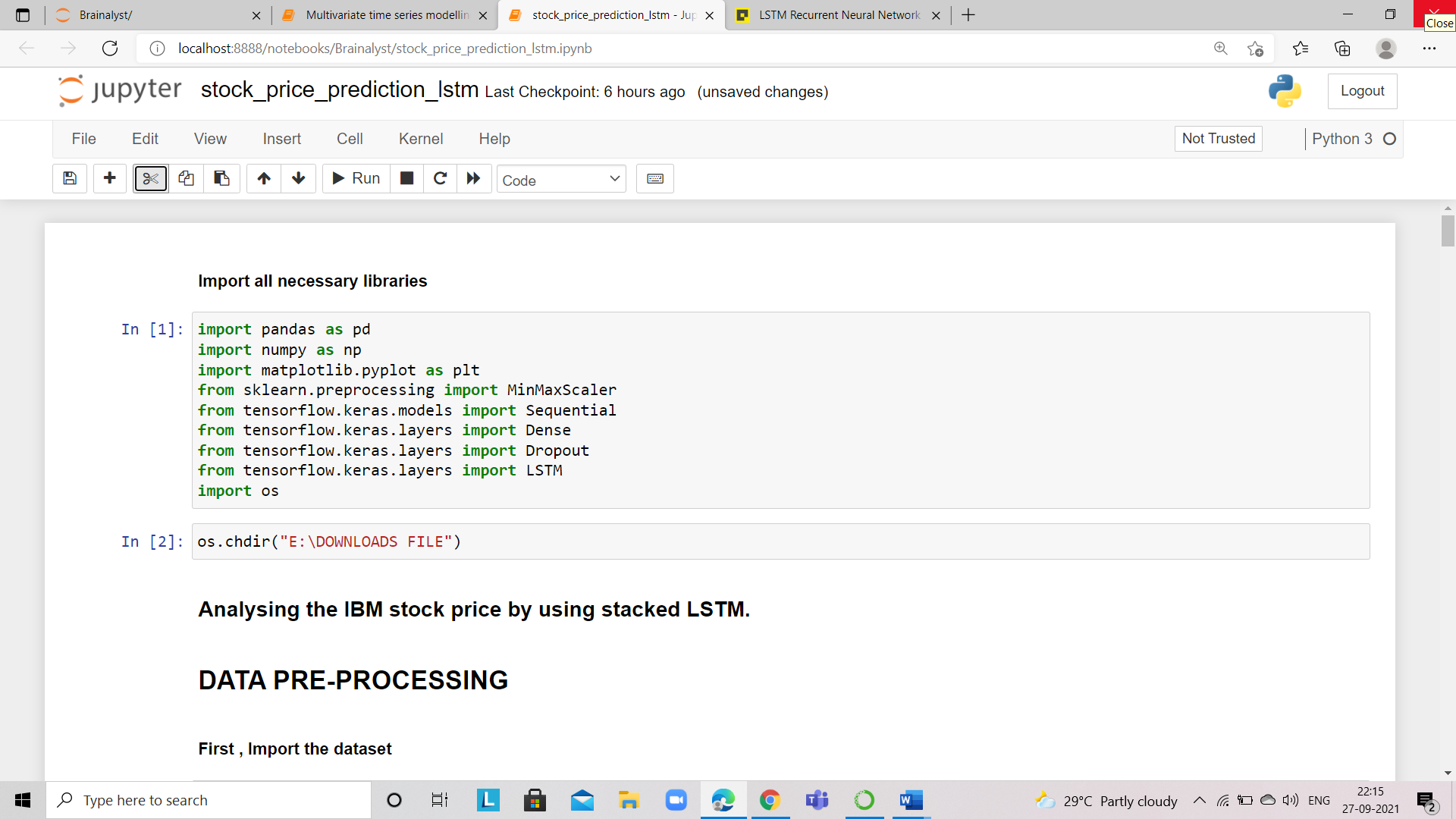
* Fundamental Analysis involves analyzing the company's future profitability based on its current business environment and financial performance.
* Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market.

And here we discuss the technical analysis part.

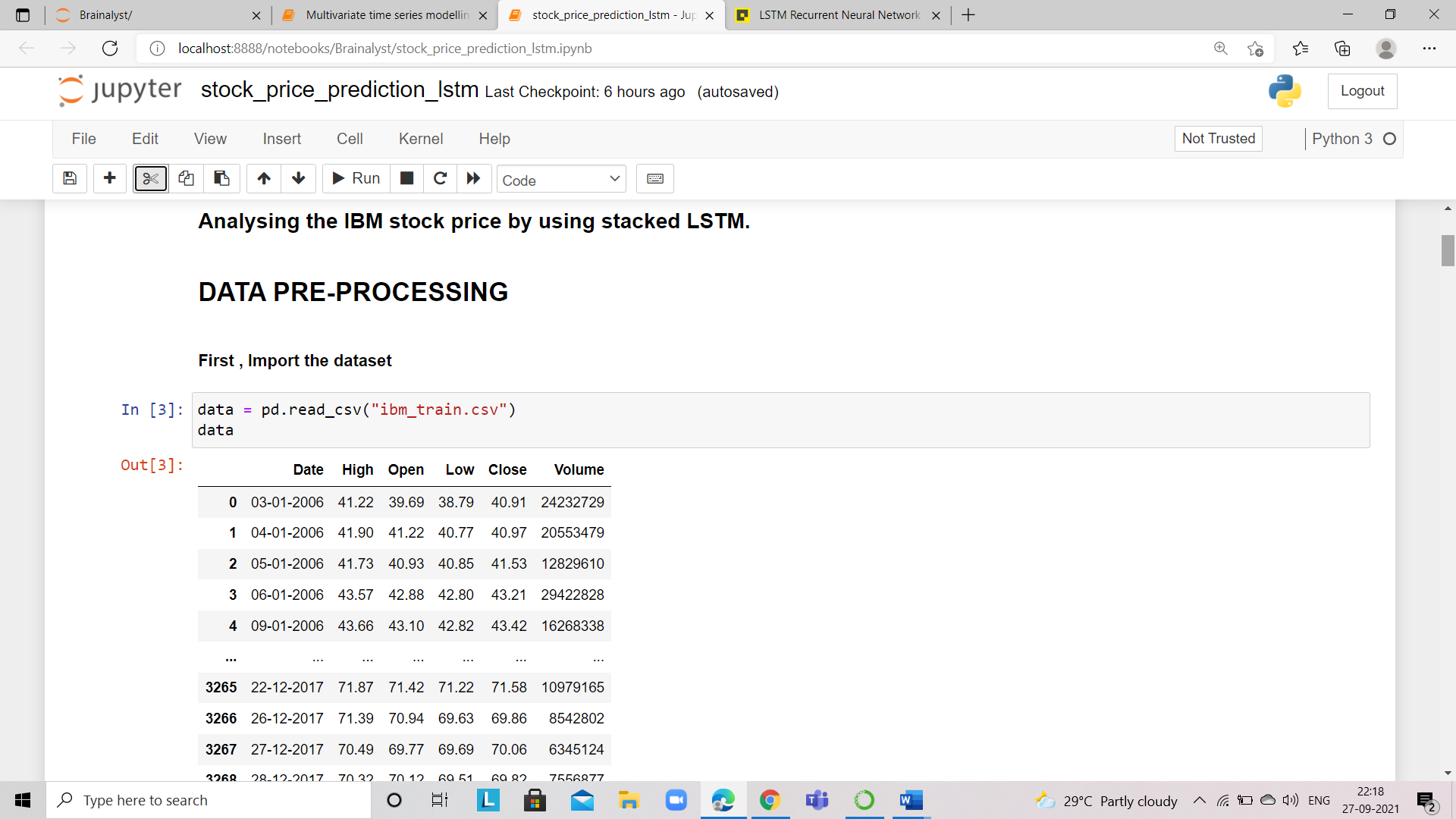
Let’s dip dive into implementation!

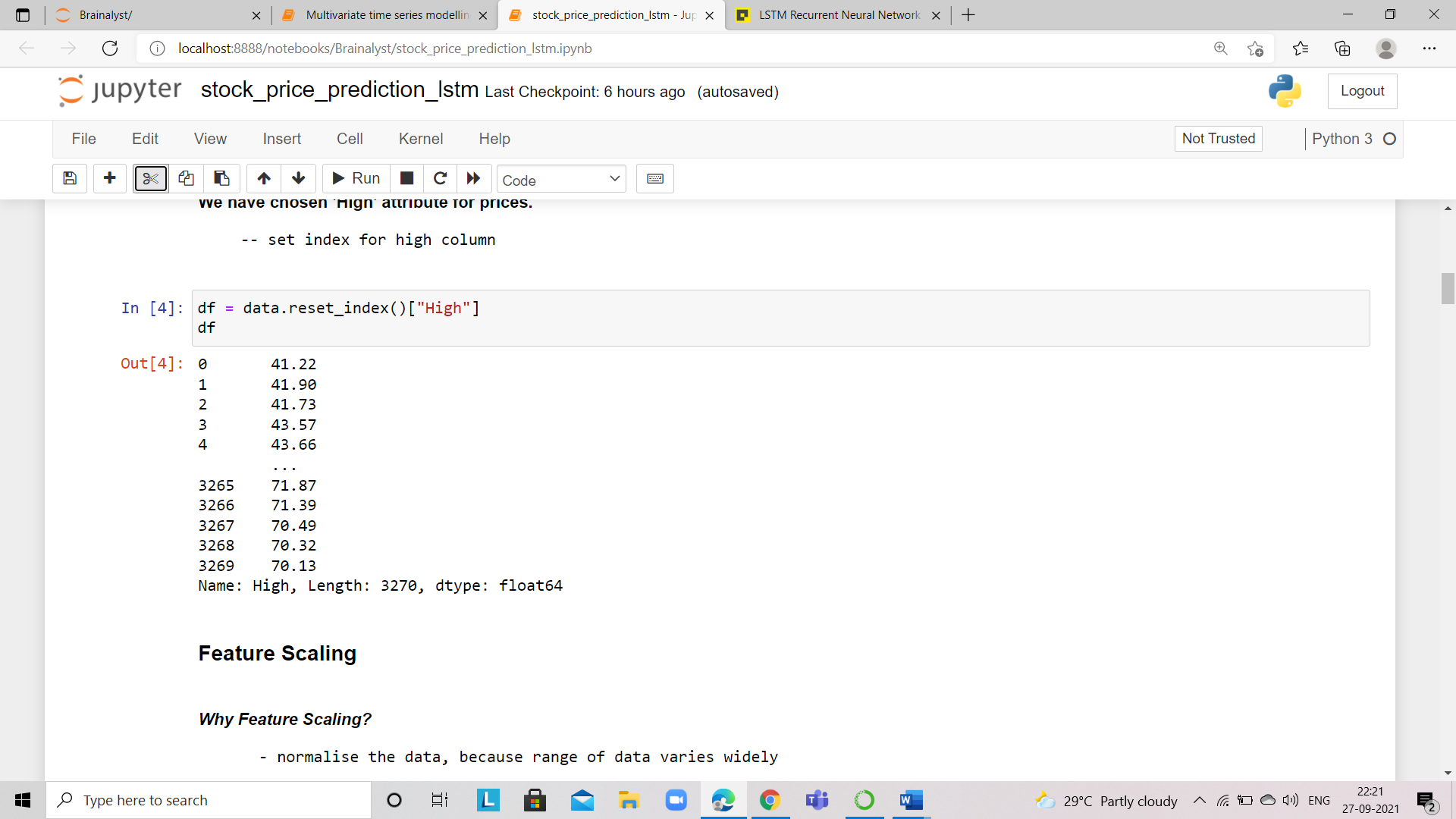
**IMPLEMENTATION**

1. Import all the required libraries



1. Now read the CSV file



1. After importing the data, now we have to choose which column we want to predict, so here we will use the column "High", so set the index as a high column to make the series or array.
2. Now we get our time series, now we need to do feature scaling because if you see the range of high column as it varies widely so we normalize our data by using MinMax Scaler.

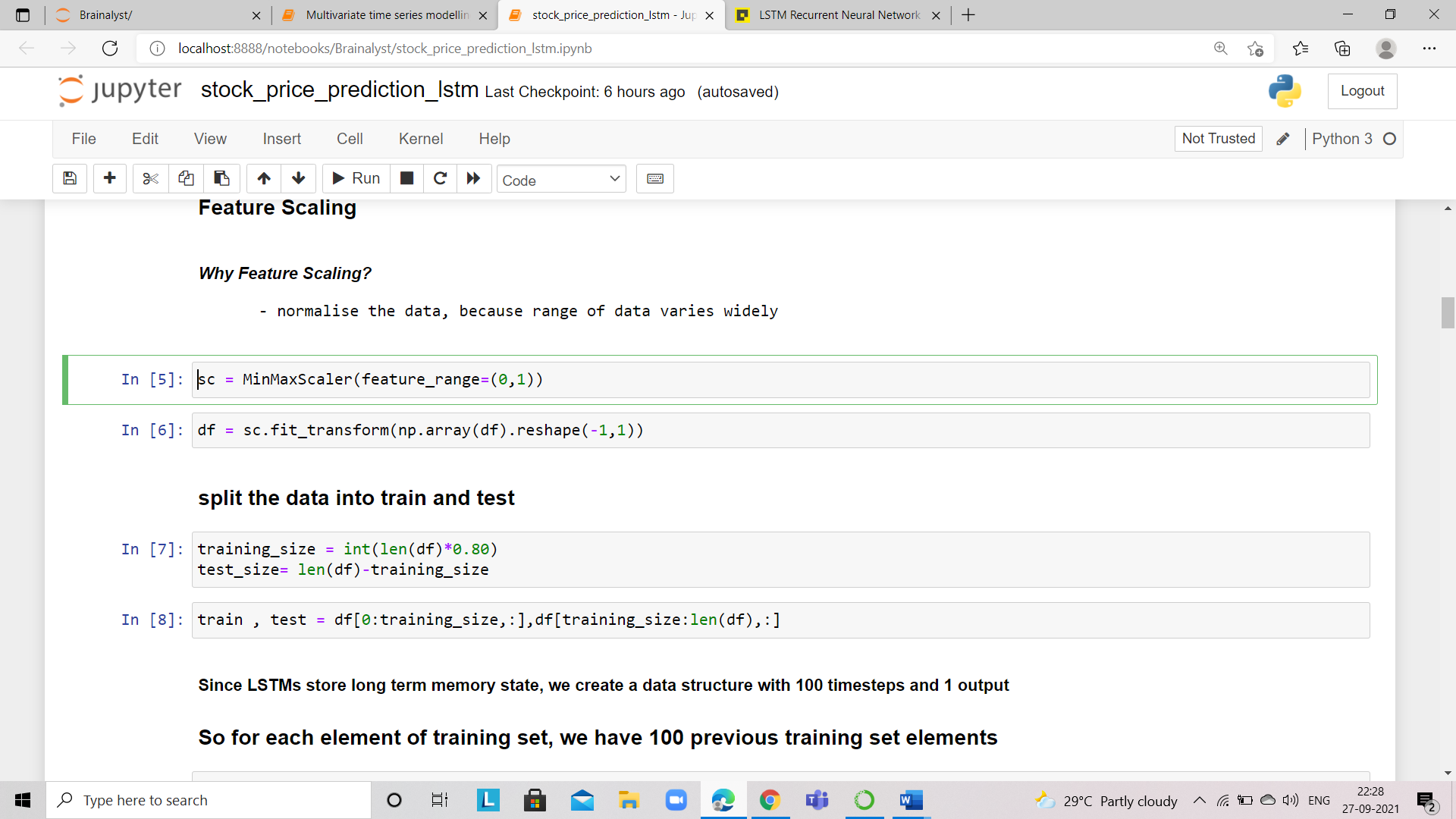
What is MinMax Scaler?

The min-max scalar form of normalization uses **the mean and standard deviation to box all the data into a range lying between a certain min and max value**. For most purposes, the range is set between 0 and 1.

URL - <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

the referred link for min-max scaler if you want to study minimax scaler.

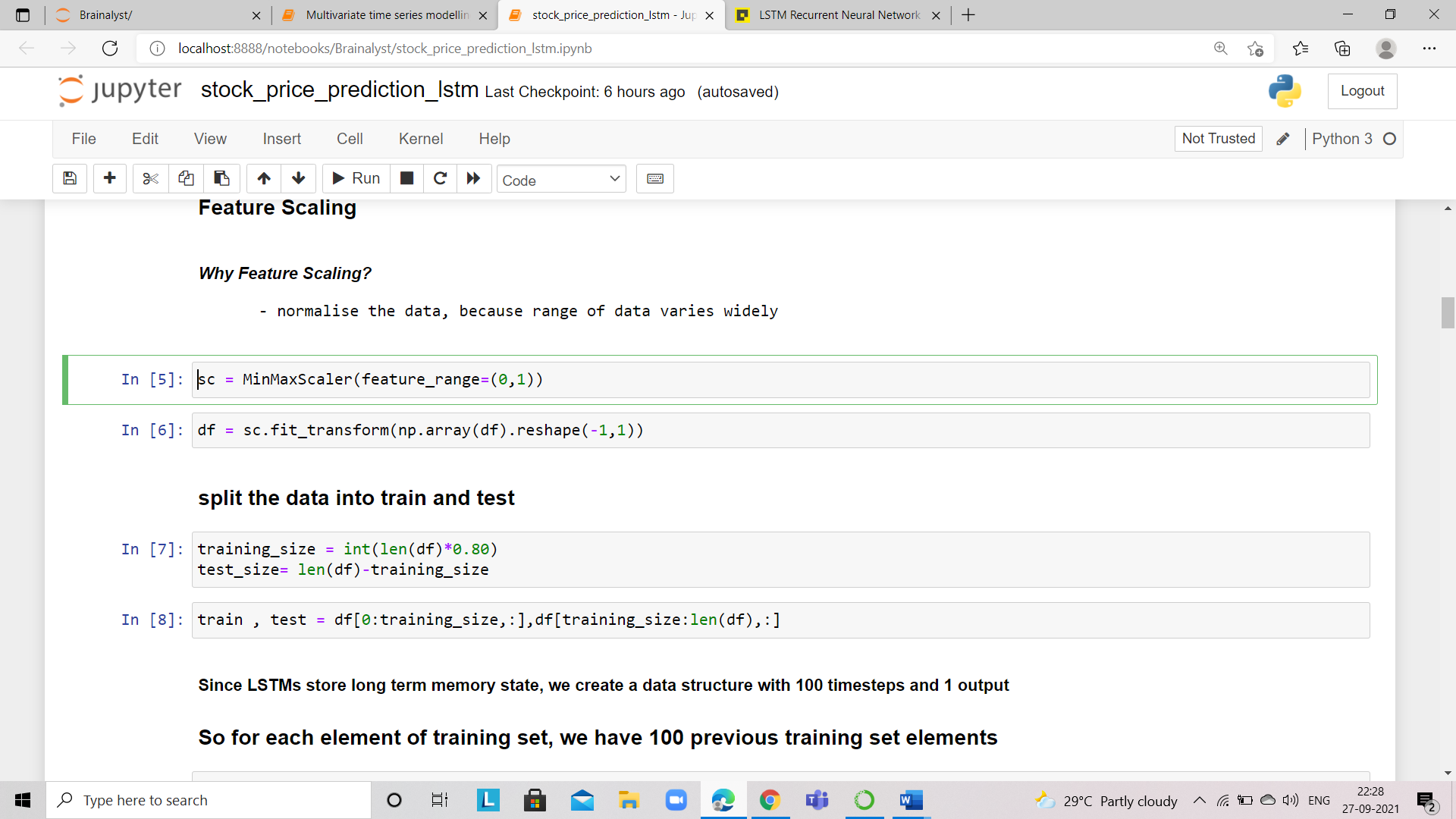
After scaling, we have to Fit data, then transform it.



Finally, our Data Pre-Processing is done, now data is prepared for our deep learning model.

1. Now our next task is to split the prepared data into train and test for trained the model and for validating the model.

We split the data into an 80:20 ratio as 80% of data is used for training the model and 20% of data is used for validating the model.



1. After splitting the data, now we preprocess our data again for our LSTM.

Since the LSTM model is trained on long-term memory state so our model needs previous data for predicting today's price? Right?

We need to create the timesteps….i.e on how many days you want to predict today's price.

Let's clear this with a basic example:

In our data, we have data for 6 days…..based on 6 days we want to predict the future stock price for the 7th day.

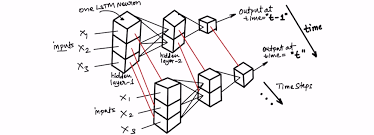
120,130,125,137,140,142 our data of 6 days now we want to predict the stock price of the 7th day and we get the output to suppose 140 based on 6 days.

For the 8th day now we predict based on 130,125,137,140,142,140 on the again previous 6 days and suppose we get the output 145.

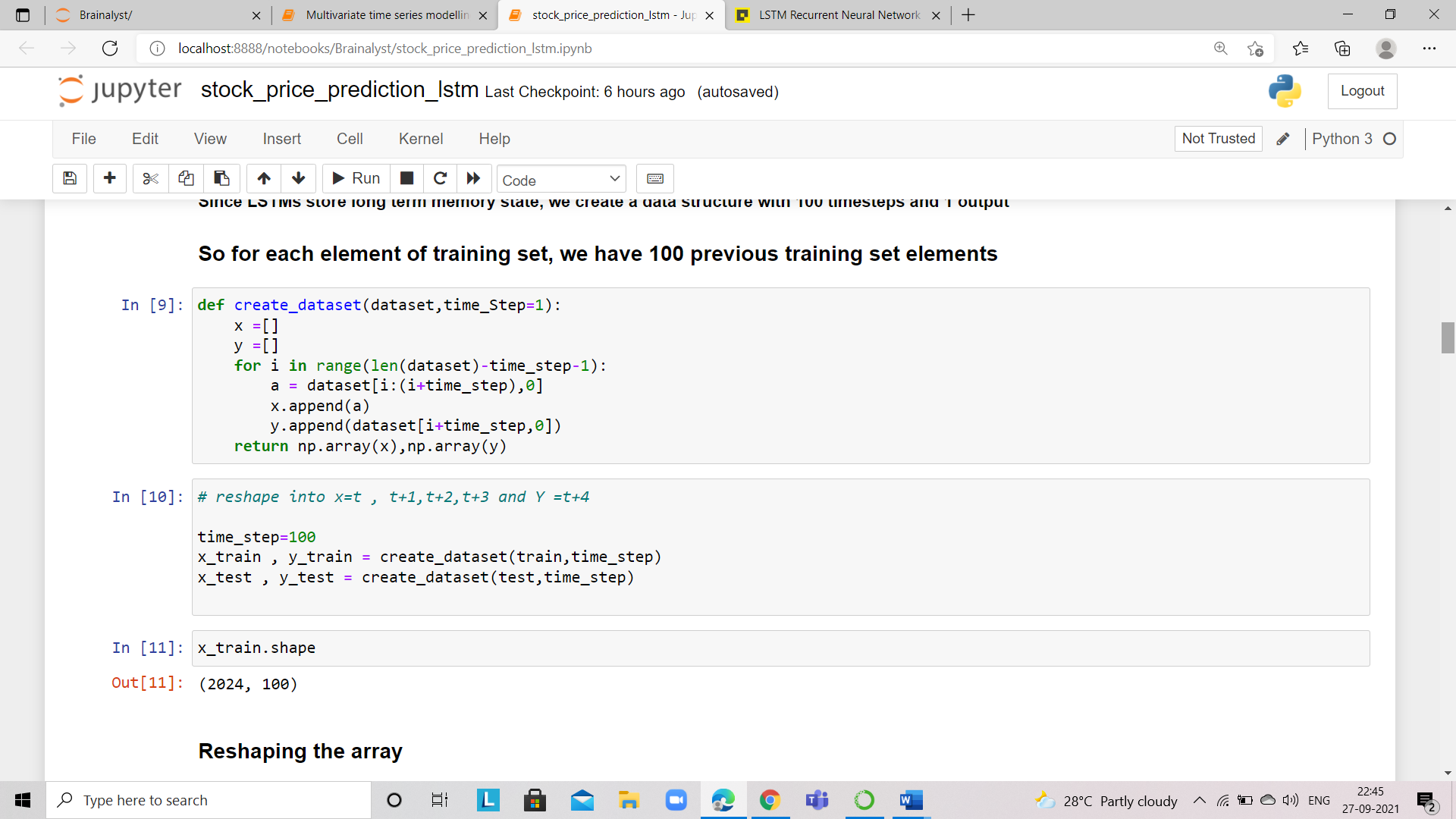
For the 9th day, we predict based on 125,137,140,142,140,145 on the again previous 6 days and suppose we get the output of 143.

And this process begins to continue until the day you want to predict.

So here timestep =6 and output =1

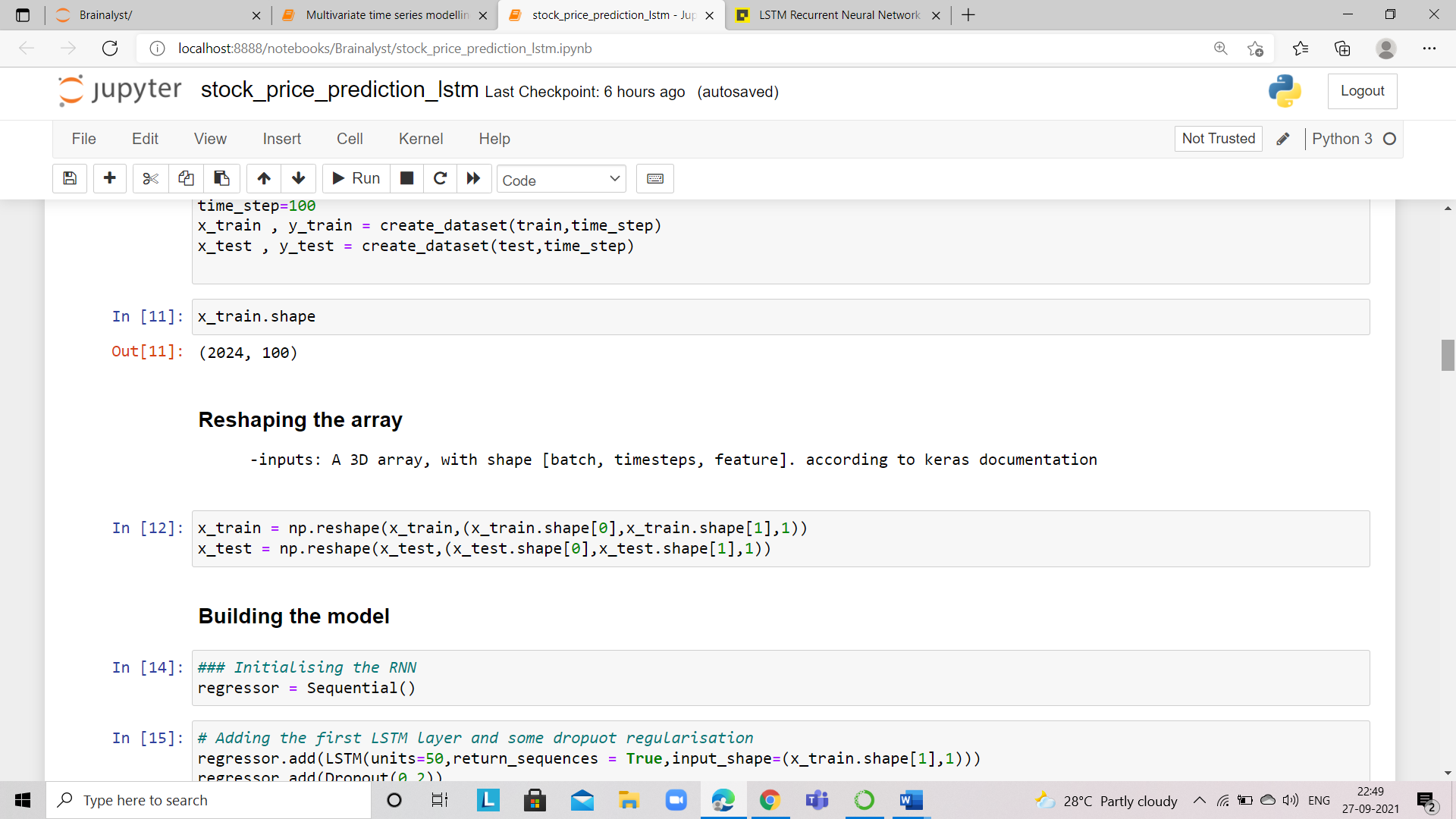


But in our data, we decide the **timesteps = 100 you can change if you want to improve the accuracy of the model.**



Based on 100 timesteps, we split data into X and Y, because the model needs dependent and independent variables…so split data accordingly.

Before building the LSTM model always ensure that your x\_train and x\_test should be an array and the shape of that array is 3D, if x\_train and x\_test is the shape of 2D so reshape your data into 3D. Because LSTM always takes a 3D shape.



I am sure after seeing the code you get doubt in mind how to reshape what we did here.

So our array should be like this A 3D tensor, with shape [batch, timesteps, feature].

URL - <https://keras.io/api/layers/recurrent_layers/simple_rnn/>

Batch – observations of training data

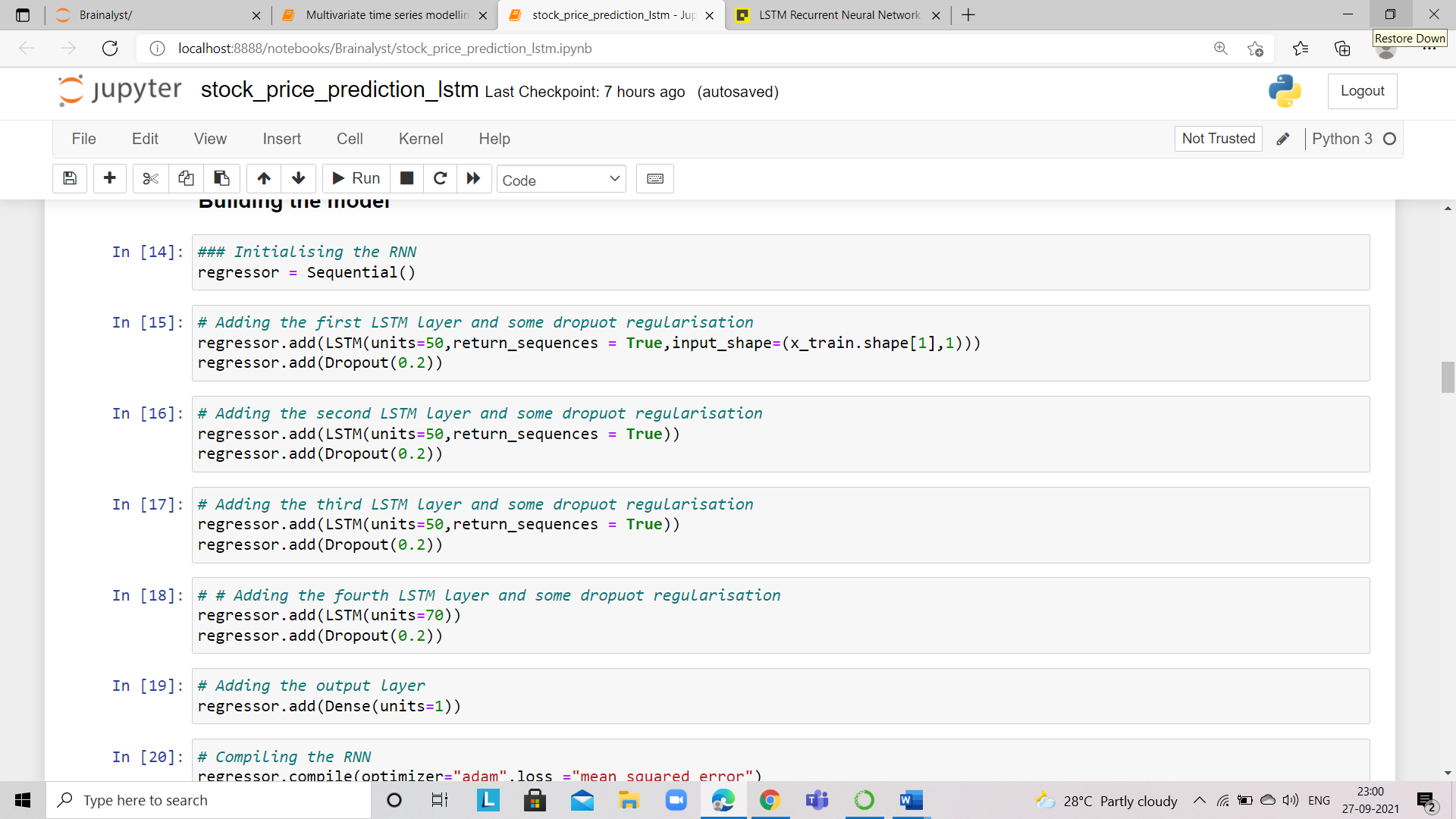
Timesteps – 100(defined earlier)

Feature – feature in our train data as we have only 1 feature as discussed earlier

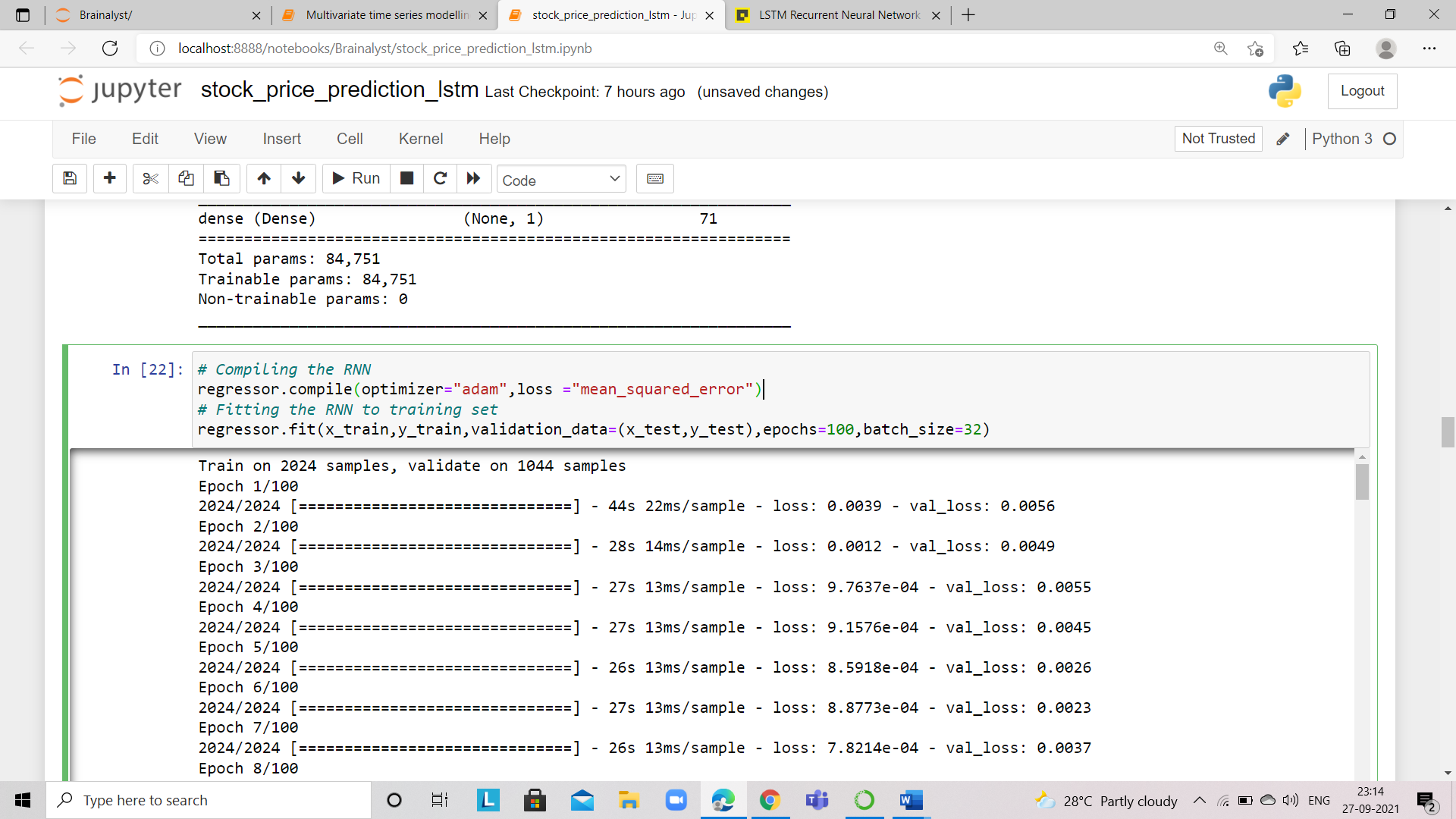
Now finally our data is prepared for building the model.

1. **Building the model**

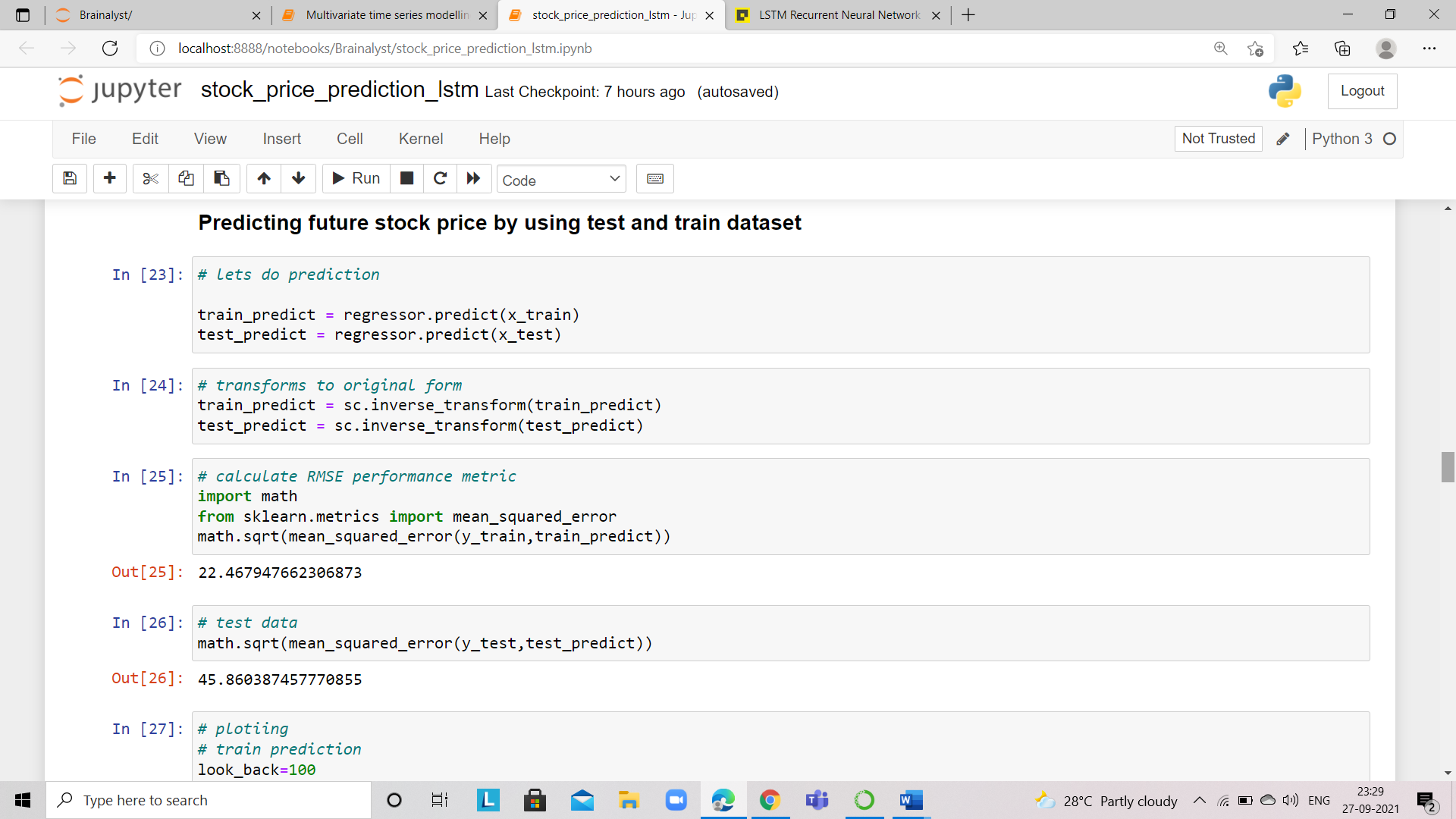
Define the LSTM Recurrent Neural Network. Here, you can add more LSTM layers and adjust the dropout to improve the accuracy of the model.



1. Compile and train the model defined in the above step. Iteratively, you can increase or decrease the epochs and batch size to get more accuracy.

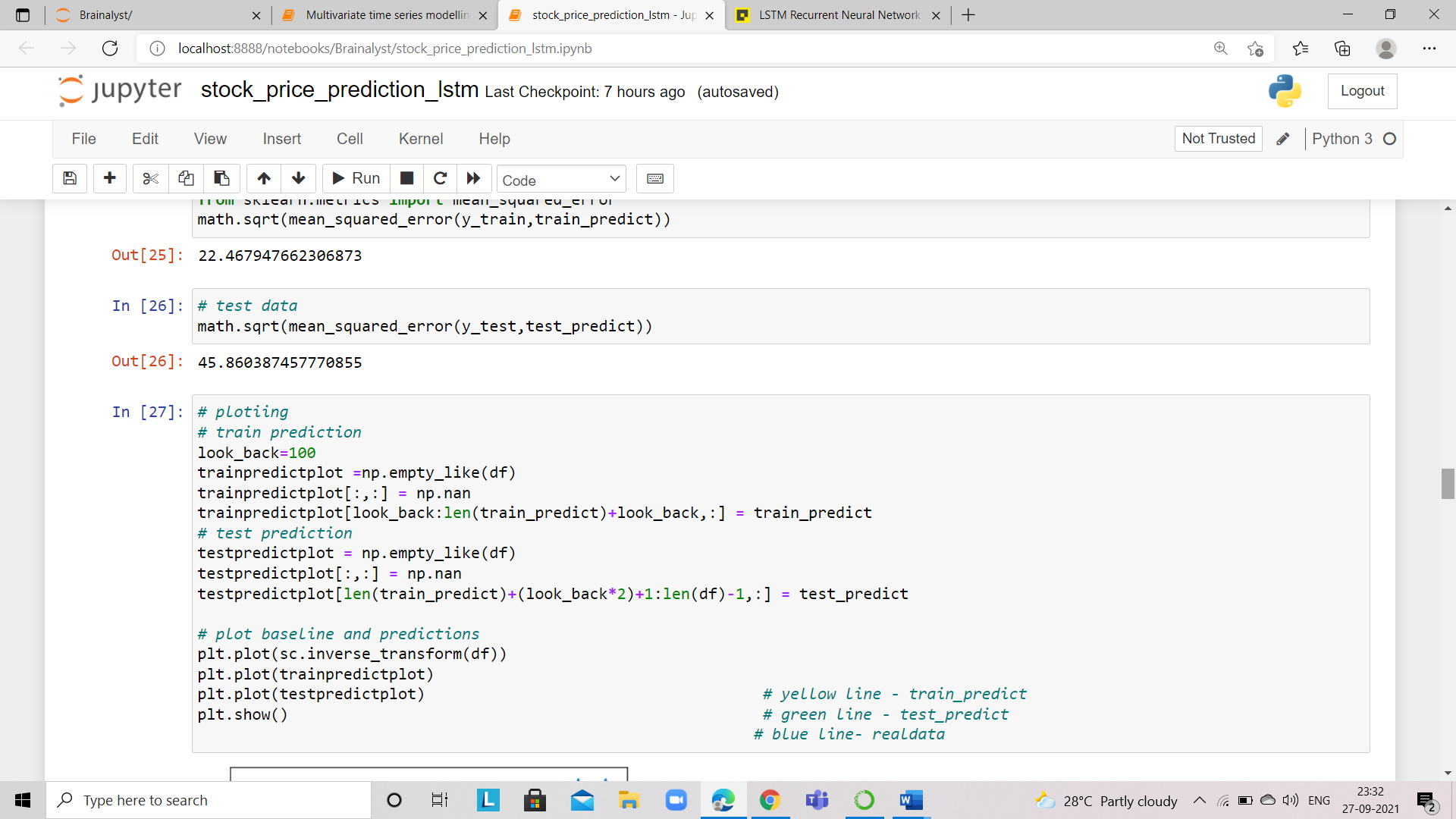


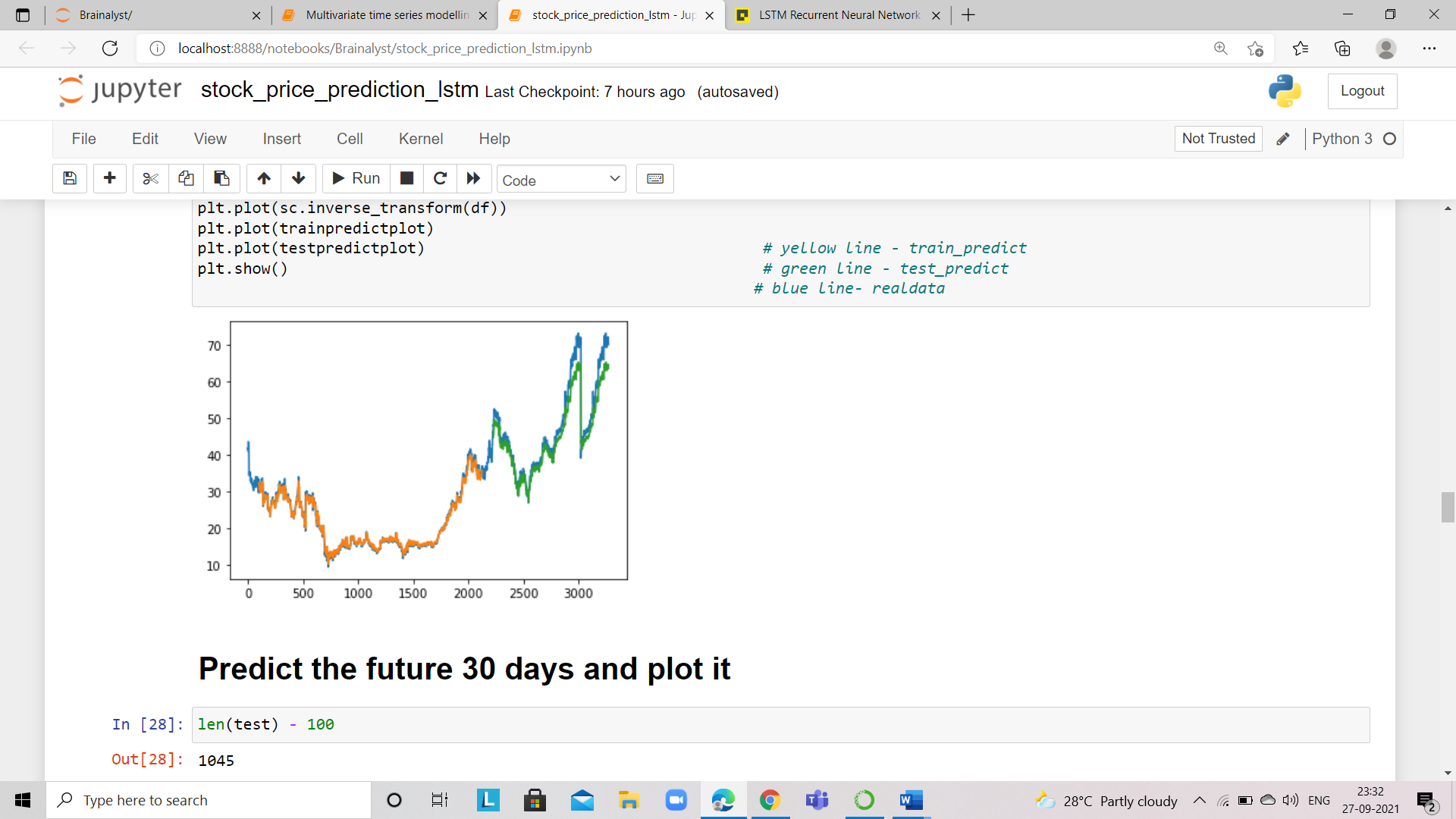
Now the model is trained we want to predict the future stock price for test data and calculate the RMSE (Root Mean Squared Error) on the train and test data.



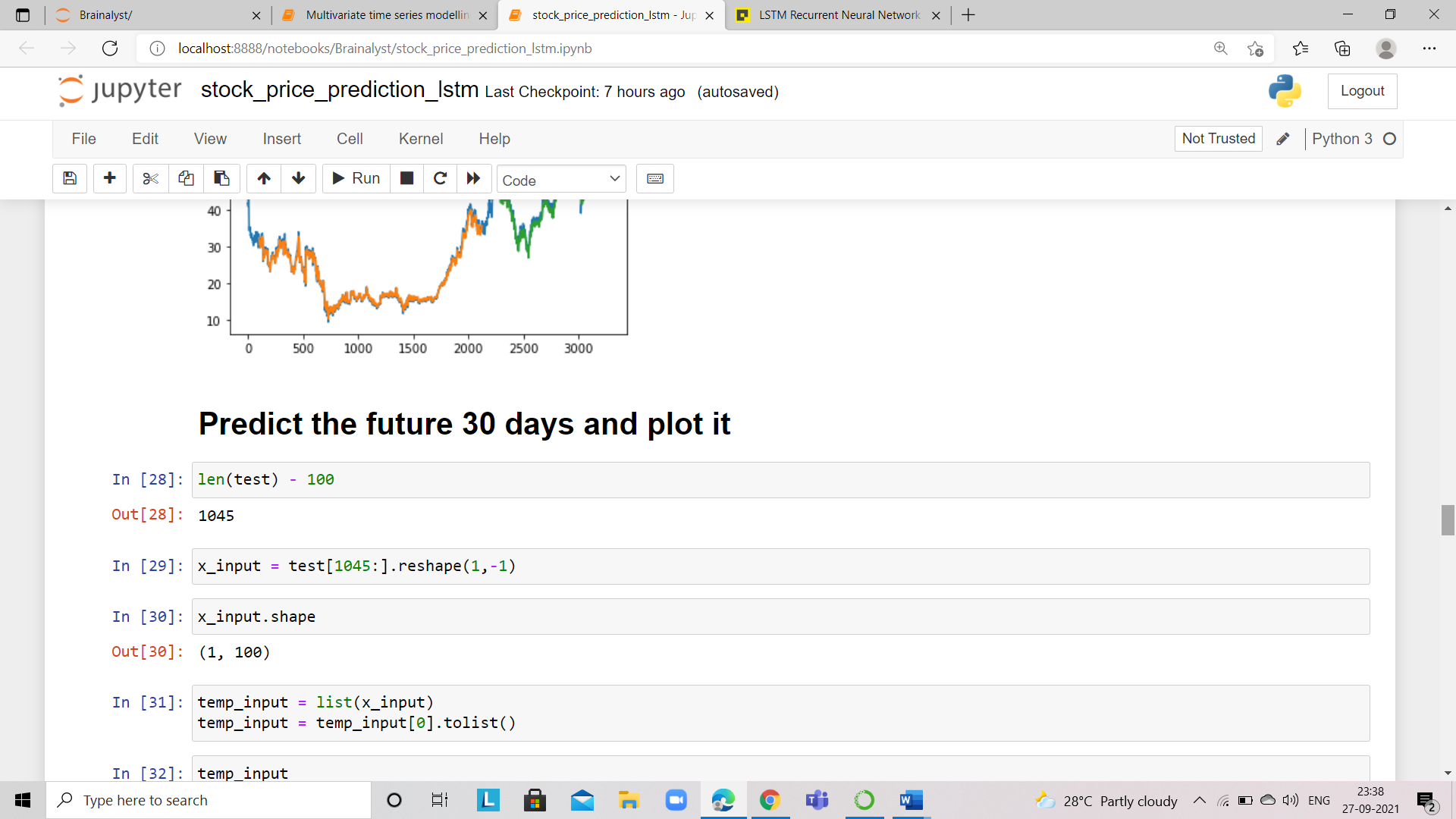
After getting predictions, inverse the prediction to the original form to compare with the real stock price.

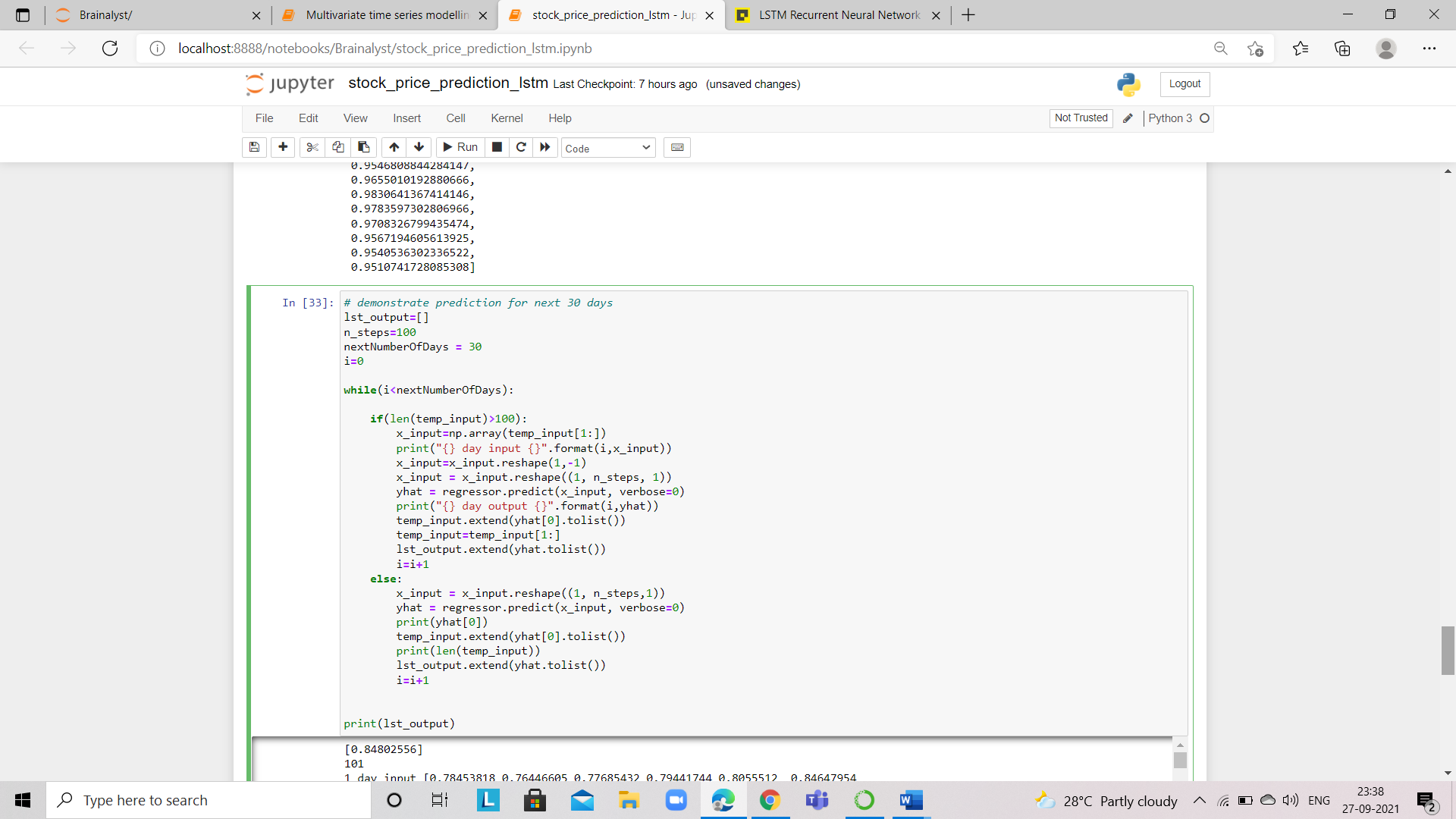
1. Visualize the predicted stock prices with original stock prices.





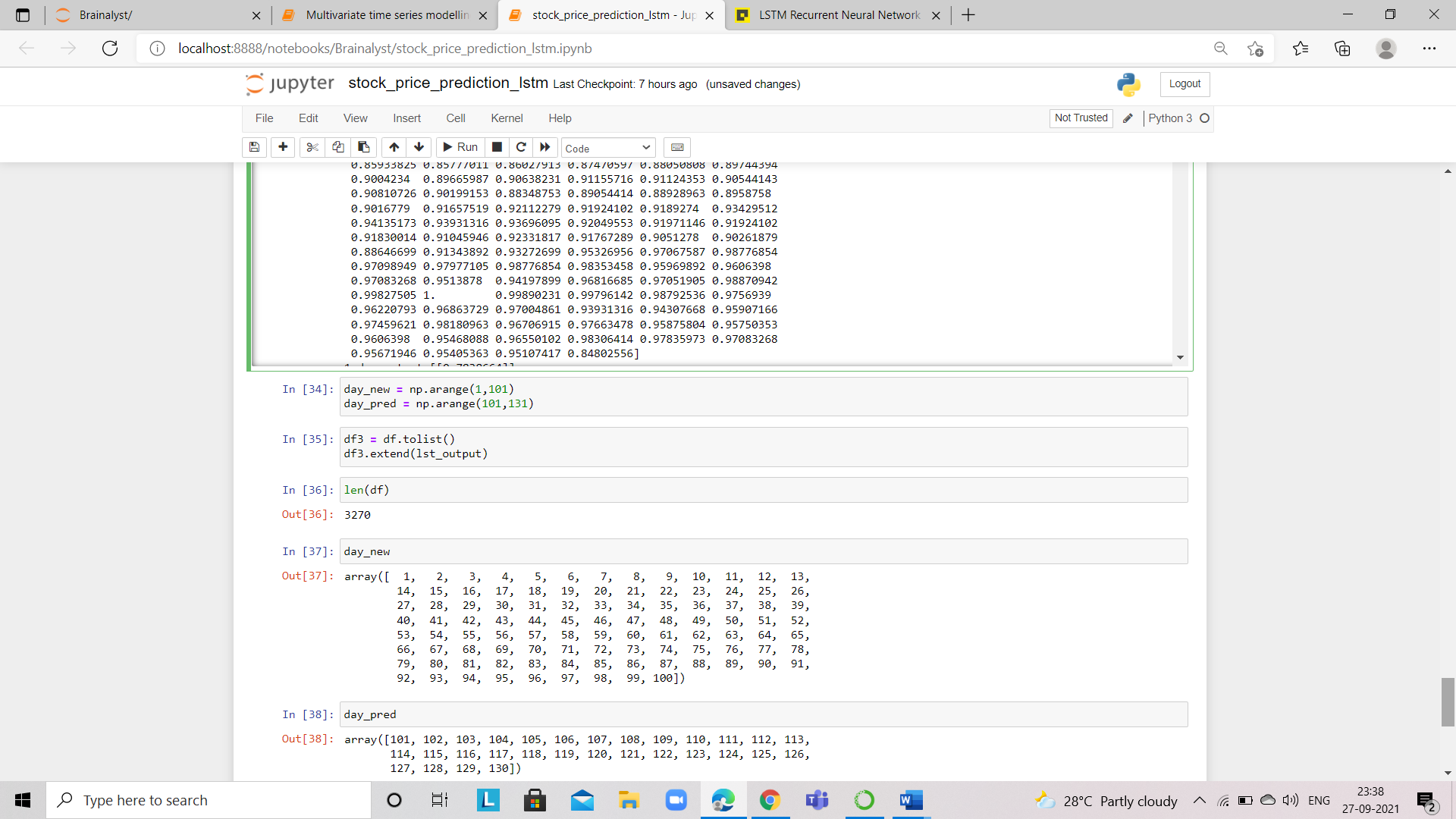
Now done with the prediction for the test data. If someone wants to ask to predict the future stock price for the next 30 days.

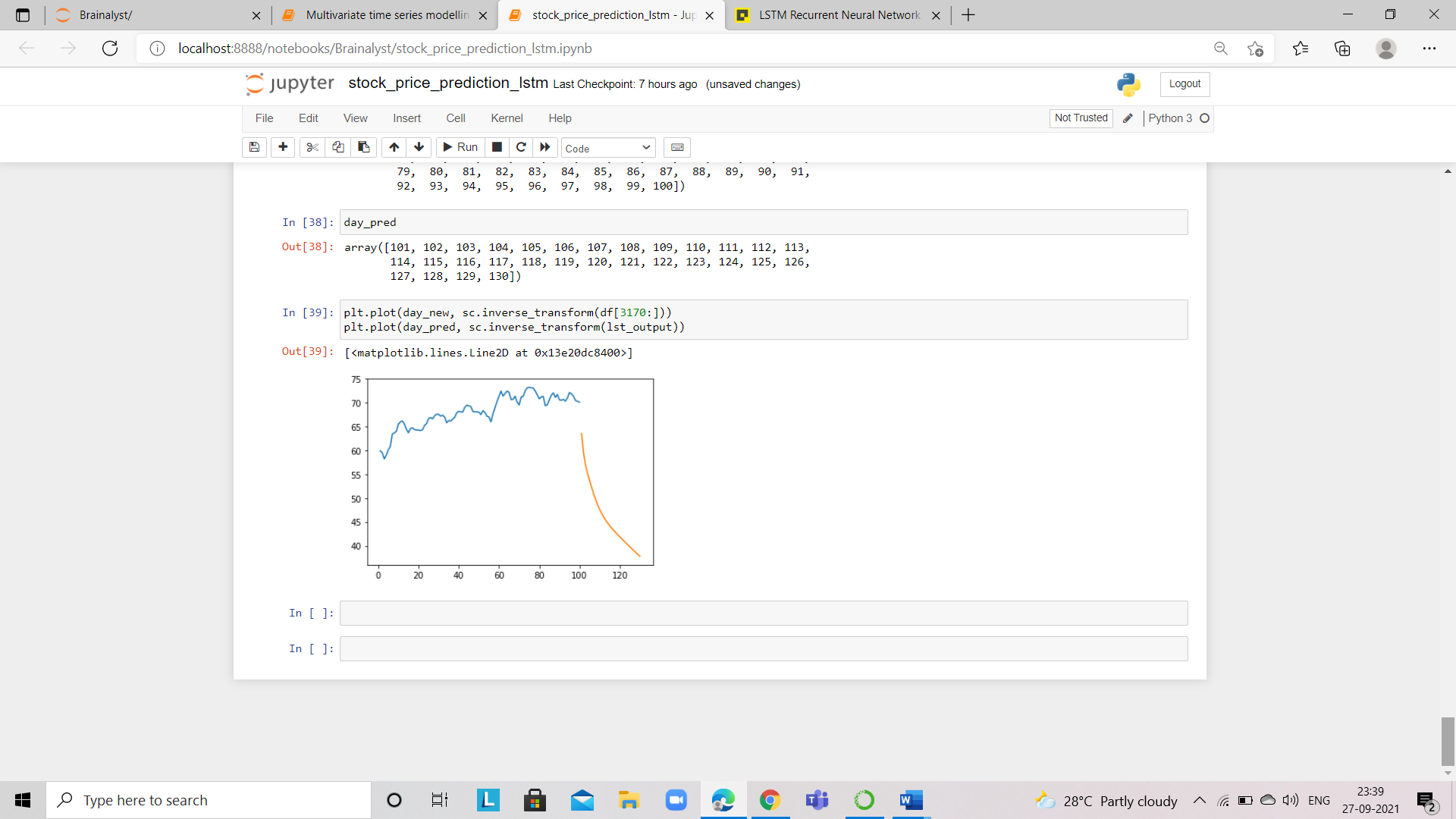
* Earlier we define the 100 timesteps so we subtract from test data and from that index we create new datapoints for predictions
* And create a list for those points



Define the timesteps here for our new data points and run a while loop based on the given condition.

* And finally, visualize the predicted stock price for the next 30 days.





Our LSTM model, predict the future stock price for the next 30 days is going to be down as shown in the picture - orange line for predictions.

However, The LSTM model can be tuned for various parameters such as changing the number of LSTM layers, adding dropout value, or increasing the number of epochs. But are the predictions from LSTM enough to identify whether the stock price will increase or decrease? Certainly not!. The stock price is affected by the news about the company and other factors like demonetization or merger/demerger of the companies. There are certain intangible factors as well which can often be impossible to predict.

I hope you understand now how the stock price prediction is to be done by using the LSTM model.

*Go ahead, Happy Learning!*