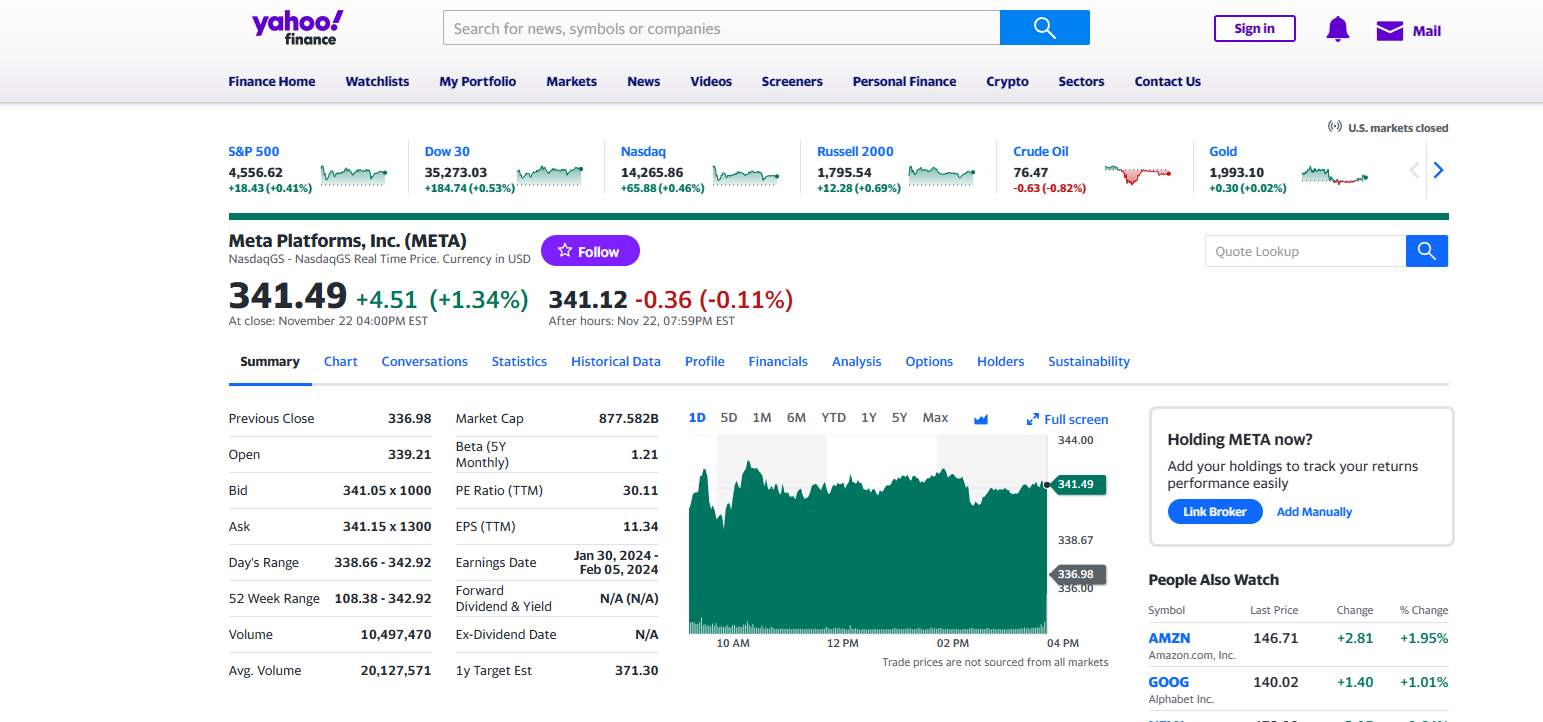
**Challenge 2: Time Series Prediction**

**Overview:**

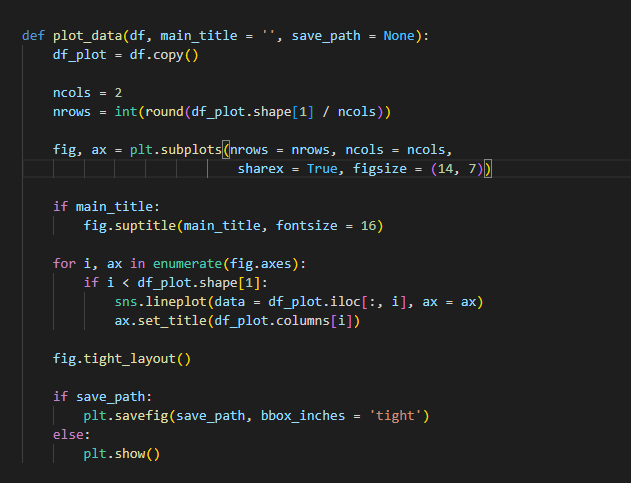
The objective of this project is to develop an advanced Time Series model for monitoring and predicting the “Closed” stock prices of ALphabet Inc. and Meta Inc. This predictive model aims to empower investors with valuable insights, aiding them in making informed decisions. The input data of the model consists of historical records of the stock “Closed” prices, and the output is the prediction of the stock price in 30 days.

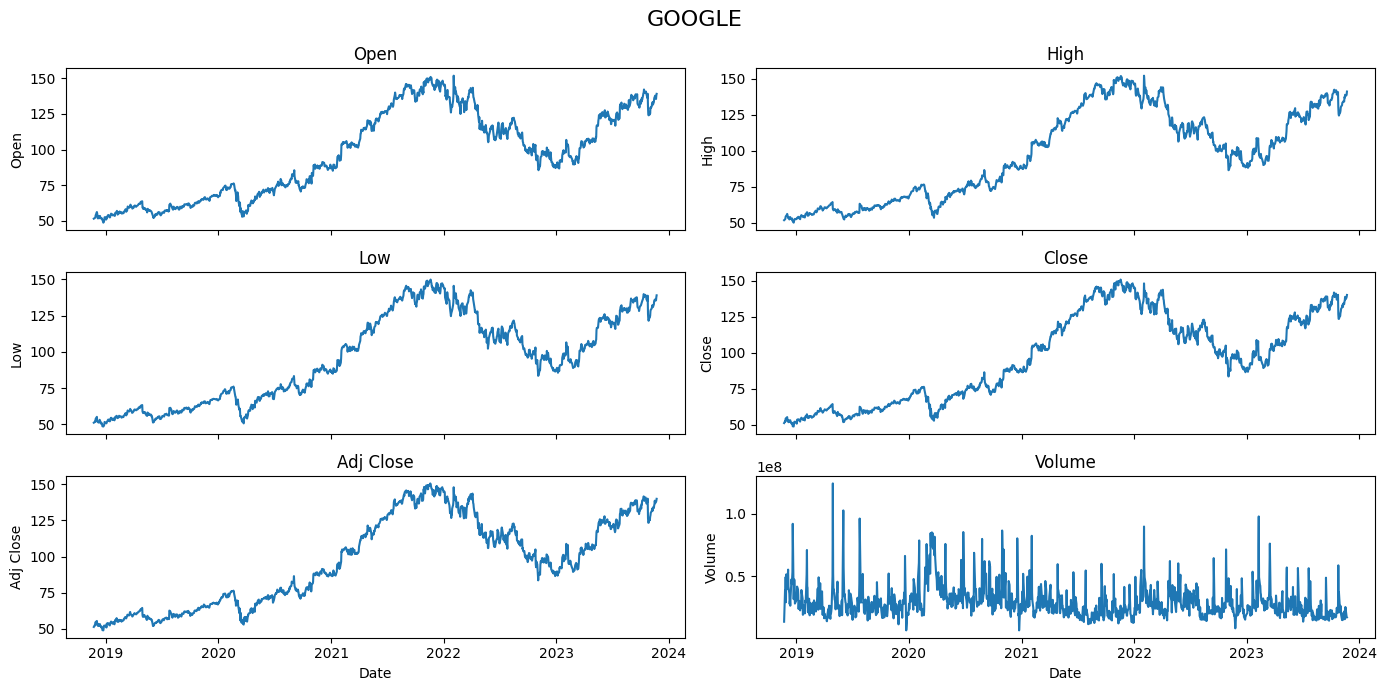
**Data**

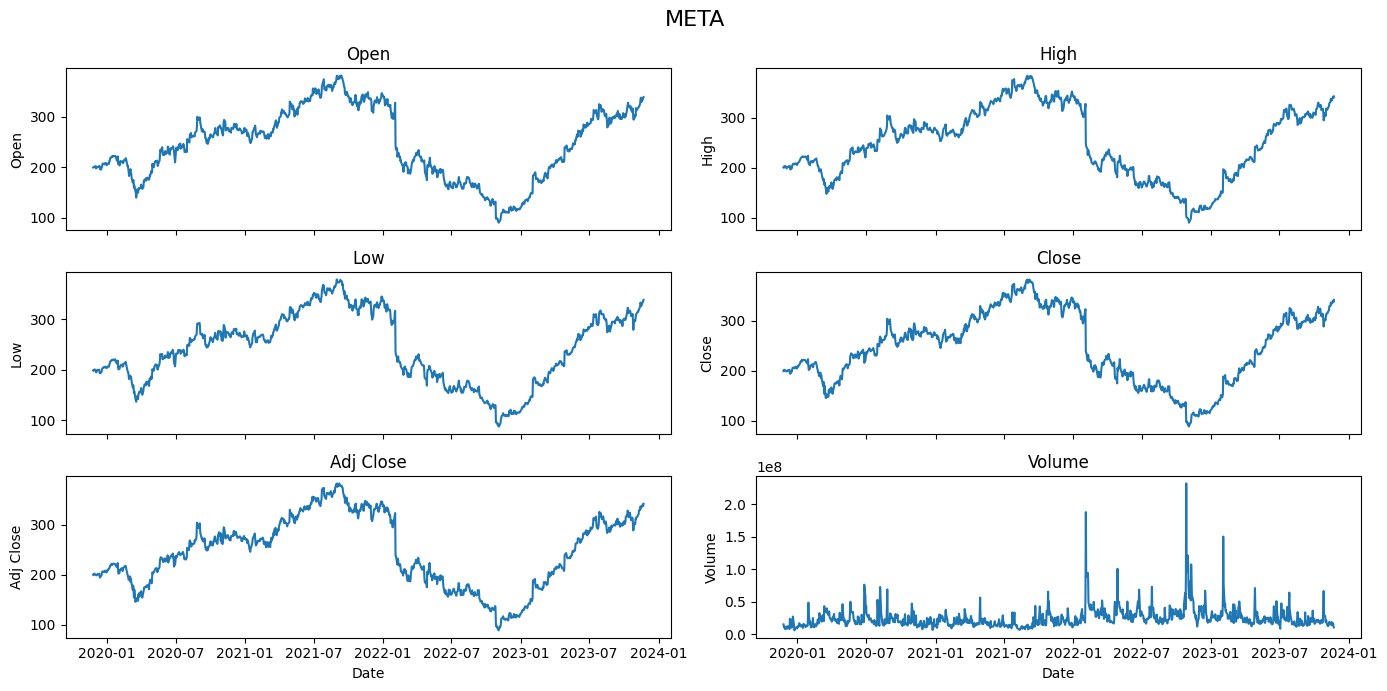
The dataset used for this project comprises of historical records of the “Closed” prices of Alphabet Inc. and Meta Inc. stocks for 5 years, ranging from 2019 to 2023. The dataset is gathered from Yahoo Finance website and is preprocessed in the chronological order.

 **Exploratory Data Analysis (EDA)**

EDA involves a thorough analysis of the dataset to extract insights that inform subsequent steps in the modeling process. Statistical summaries, visualizations (such as line plots, histograms, and correlation matrices), and trend analyses help understand the underlying patterns and relationships in the data. EDA also guides feature selection and engineering.

This function plots several diagrams for each attribute in the dataset. The original stock dataset has 6 columns: Open, High, Low, Close, Adj Close, Volume. These plots are line graphs describing the tendency of the stock price over the years.



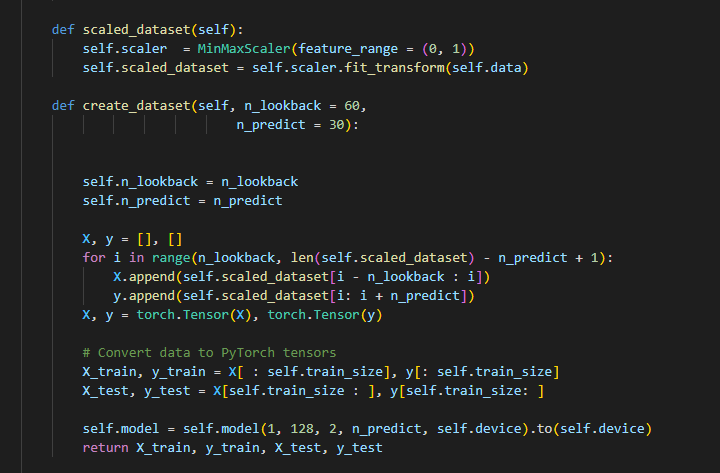


**Feature Engineering**

Determining the window size for the Time Series data is indeed a crucial aspect of feature engineering. The window size defines how many past observations of the model is considered when making decisions. I denote the window size for past observation is *n\_lookback*. A larget *n\_lookback* allows the model to capture longer trends but may increase computational complexity and poorly performs on the data with too much randomness. Based on the trend and the seasonality of the plot, the window size is set to 60 days.

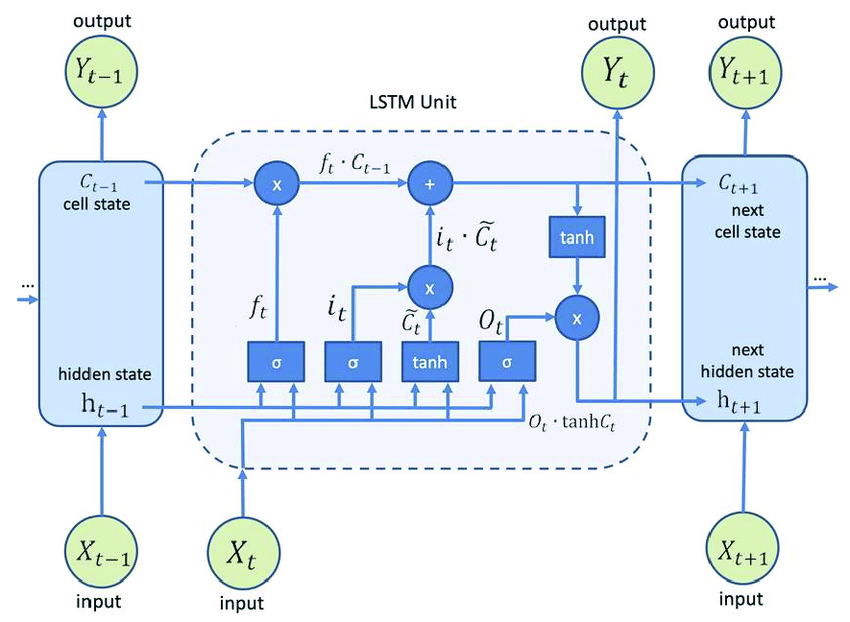
The forecast horizon, denoted as *n\_forecast* determines how many time steps into the future the model aims to predict. It is the time period that you want to forecast the stock price. Most Time Series models predict only 1 future day and use the sliding window technique to predict the others days into the future. That approach is suitable for forecasting in the short-terms changes. In this project, I set n\_forecast to 30 days.

After that, the data is split into training and testing sets with the proportion 80% and 20% respectively.

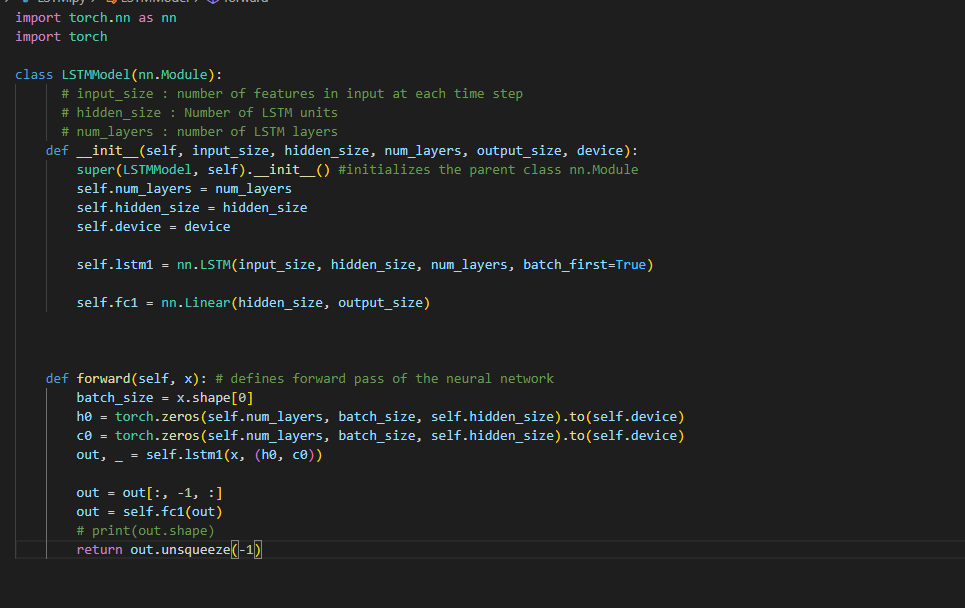


**Model selection**

Choosing the right Time Series model is crucial for accurate predictions. In this project, Long Short-Term Memory (LSTM) is used to capture the non-linearity relationships between each lags value of the dataset.



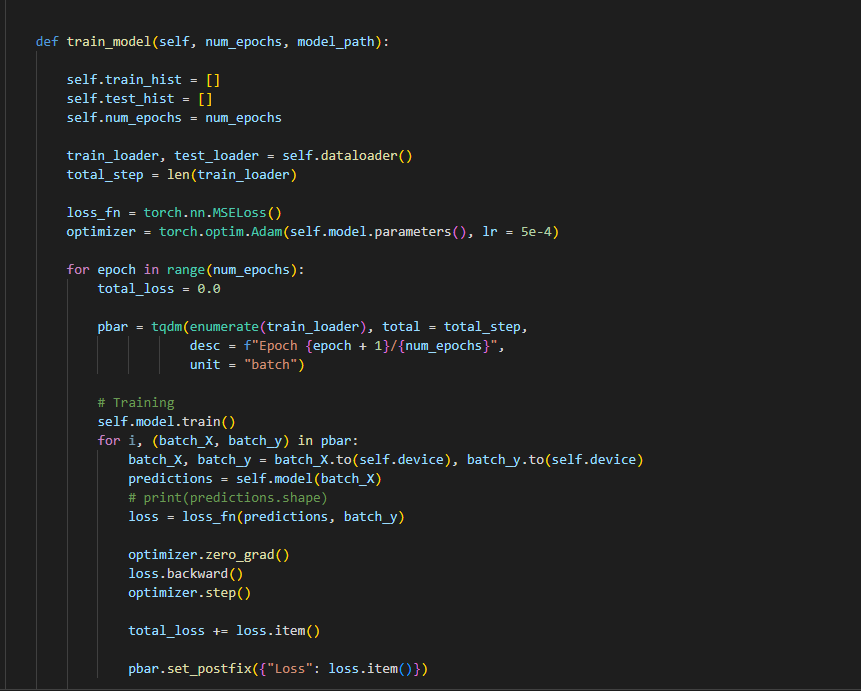
LSTM is a type of recurrent neural network (RNN) designed to capture long-term dependencies in the sequential data. Unlike traditional neural networks, LSTM networks include the memory cells that can store and retrieve information for extended periods. Since the n\_lookback is 60 days, LSTM is suitable for this job.

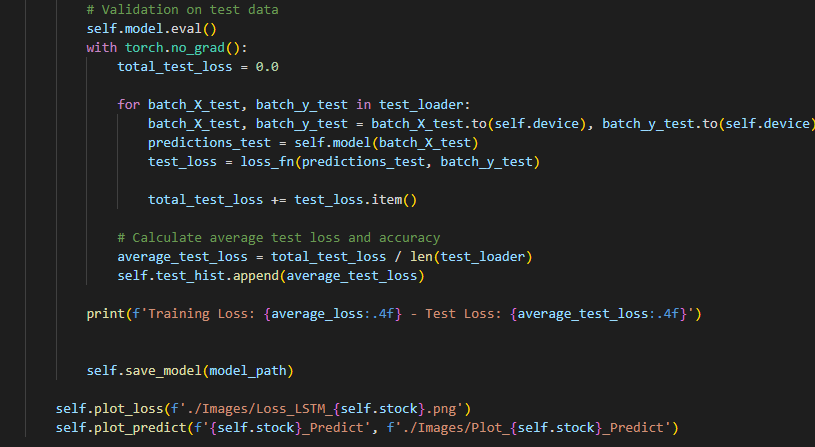


After traversing the LSTM cells, it takes the last element of the sequence and goes through a Fully Connected Layer to forecast 30 days ahead.

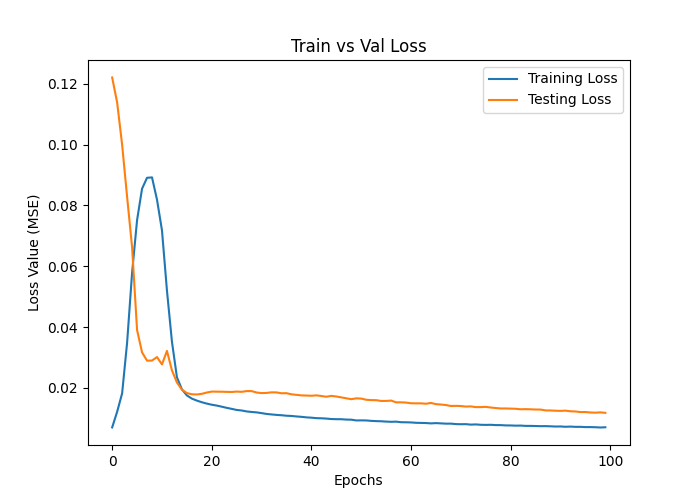
**Model Training**

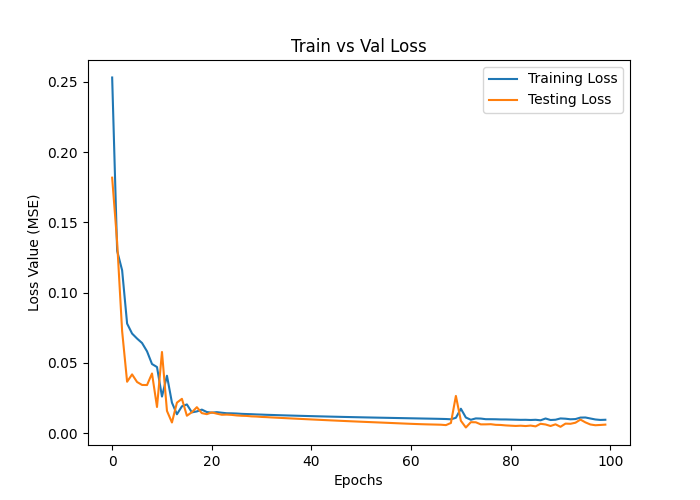
The chosen model, whether it's LSTM or another Time Series model, undergoes a training process. During training, the model learns to map input sequences to corresponding output sequences. Hyperparameters, such as the learning rate and the number of hidden layers in the case of neural networks, are adjusted to optimize performance.





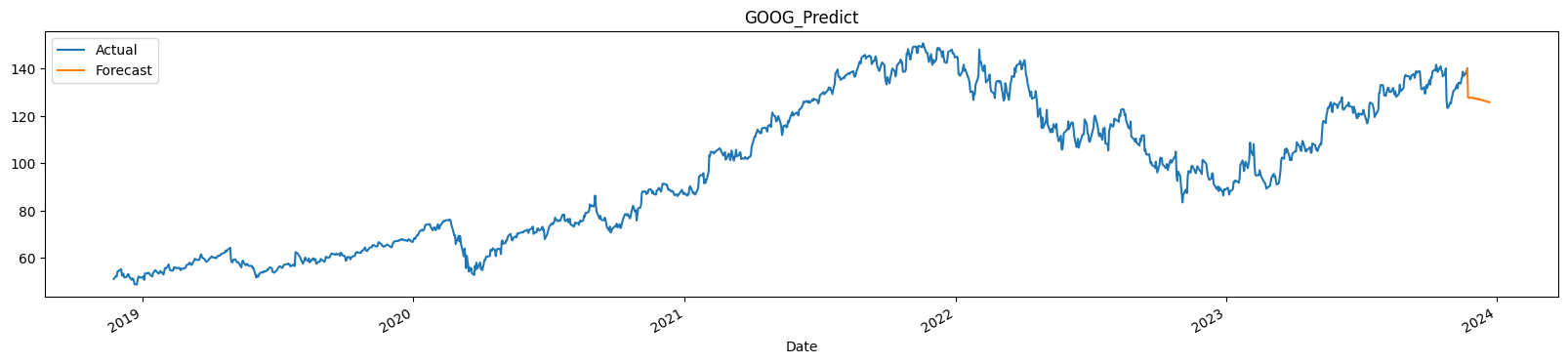
The loss function being used in this project is Mean Square Error, which is commonly used in many Time Series Model. The Adam optimizer is chosen for several reasons. This algorithm can adjusts the learning rate that can accelerate convergence and improves the efficiency. It also combines the concepts of momentum optimization and RMSprop which performs better than traditional stochastic gradient descent (SGD). The training is set to 100 epochs total. The result of the training is presented as loss function curves below.

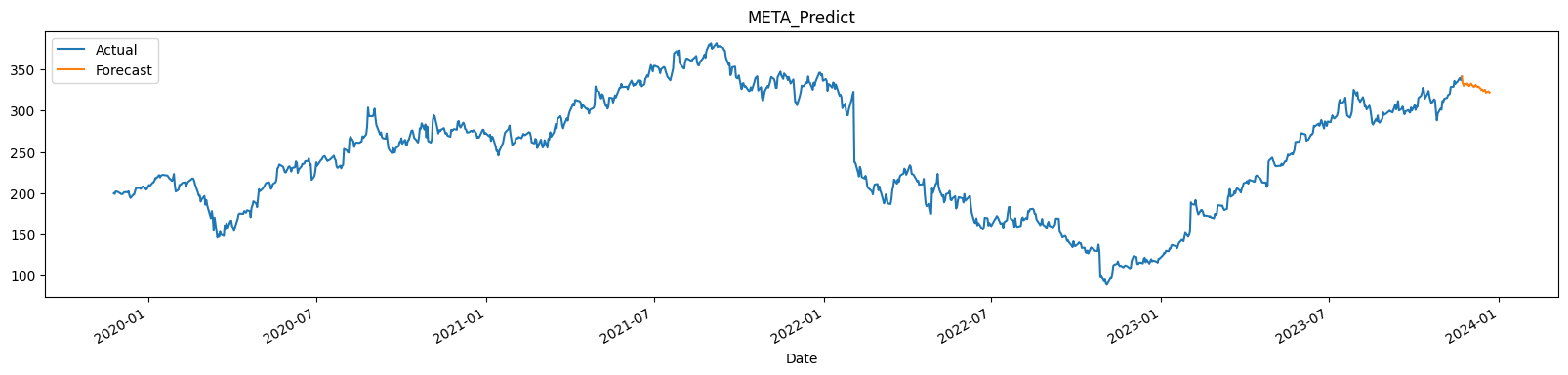
 (Loss function plot of Alphabet Inc Stock)

Loss function plots of META Inc Stock

**Model Forecasting**

After model training, I extract the most recent 60 days' data from the dataset and attempt to predict the stock prices for the upcoming 30 days. Here are the forecasted results based on this subset of the dataset.





The result forecast shows that that there will be a downward trend in the GOOG and META stocks. This result is not very accurate since we only use 1 variable to predict and discard all other factors (geopolitical events, economic indicators, …). Although, 1-day ahead forecasting (n\_forecast = 1) yields better results than 30-day forecasts (n\_forecast = 30) for one day, it does poorly on trying to predict the trend in the long-term period.

**Deployment**

Due to the short-time limitation, I use an open-source Python library called Streamlit for displaying Time Series data and Forecasting model. This library is designed to streamline the process of turning data scripts into shareable web applications. With Streamlit, it is simple to create interactive and customizable dashboards with just a few lines of code.

The current stock price and the percentage of changes are obtained and extracted throught the Selenium library. This functionality enables my web application to dynamically refresh stock values in real-time, allowing investors to monitor them instantly

