Linear Regression

Throughout the process of evaluating prospective model, we will use Google stock data for a period of two years obtained through yfinance library in python. Given below is the plot of this data.

Figure 1: Adjusted Close Price of Google Stock over two years

Linear Regression using feature derived from time index (time dummy)

This technique uses the time index to generate a dummy feature. In this case, it is just an incrementing sequence. For example, if the dummy feature for date 12-12-2022 is 1, then for 13-12-2022 it will be 2 or more. It is illustrated in the figure below:

	Adj Close	time
Date		
2020-10-14 00:00:00-04:00	78.403999	0
2020-10-15 00:00:00-04:00	77.956497	1
2020-10-16 00:00:00-04:00	78.650497	2
2020-10-19 00:00:00-04:00	76.730499	3
2020-10-20 00:00:00-04:00	77.796501	4

We will fit a linear regression model using the generated feature as our independent variable. The model learns parameters which yields results plotted as given :

Figure 2: Plot showing Linear Regression using Dummy feature

However, if we only consider its performance on test data, the results obtained are unsatisfactory. It is illustrated below.

Mean Squared Error = 2236.070748193419

The plot for just the test data is given below:

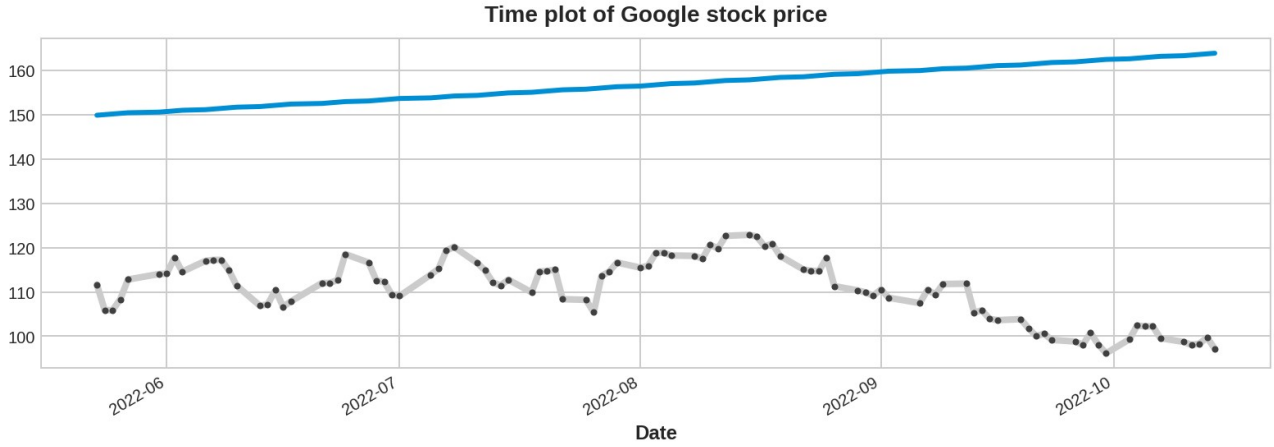

Figure 3: Plot showing linear regression using dummy feature for the test set

Linear Regression using Lag features

Sometimes the target variables (Variables we're trying to predict, Adjusted Close price in our case) are closely associated with their past values. Thus it is beneficial to use past values of the target variable as an independent variable to predict future values. This can be achieved by Lag features. It is simple the target variable shifted down one cell. In other words, given yesterday's value, we have to figure out a way how it may be used to predict today's value. An example of Lag feature with a lag of 1 (target variable shifted by one time step) is given below.

	Adj Close	time	Lag_1
Date			
2020-10-15 00:00:00-04:00	77.956497	1	78.403999
2020-10-16 00:00:00-04:00	78.650497	2	77.956497
2020-10-19 00:00:00-04:00	76.730499	3	78.650497
2020-10-20 00:00:00-04:00	77.796501	4	76.730499
2020-10-21 00:00:00-04:00	79.665497	5	77.796501

Mean Squared Error (lag of 1) = 6.966226422688199

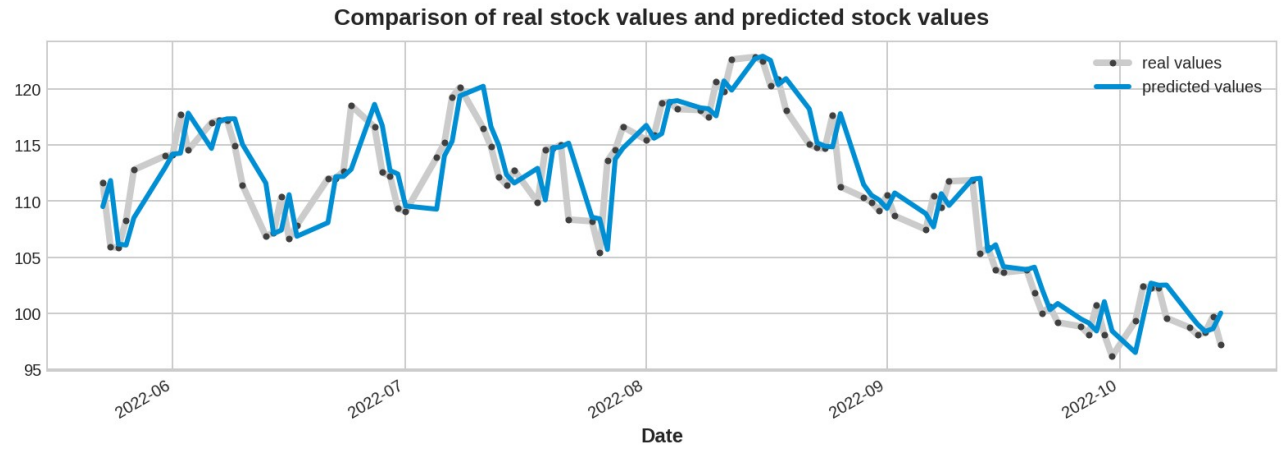

Figure 4: Linear regression using lag feature (lag of 1) (plot for test set data)

Mean Squared Error (lag of 20) = 90.02484941126214

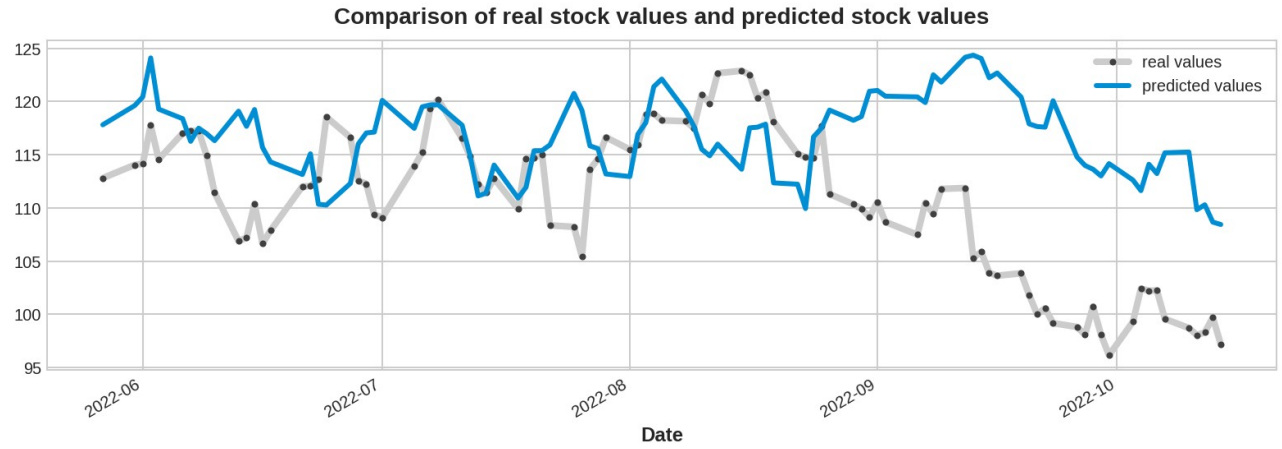

Figure 5: Linear regression using Lag feature (lag of 20) (plot for test set)

While the results obtained using lag features are better than using dummy feature, as we try to predict distant values, the predictions tend to become inaccurate or rather, they fail to account for recent changes.
In a nutshell, we still have no way to predict values further into future. For our application to be of any use, it must be able to predict values to a certain point in future with satisfactory accuracy.

Trends
Trends are long term changes obtained in the mean of time series data. In order to find evidence of any trend in our time series data, we will have to increase the time series data window from 2 years to 4 years. Given below is the plot for the same.

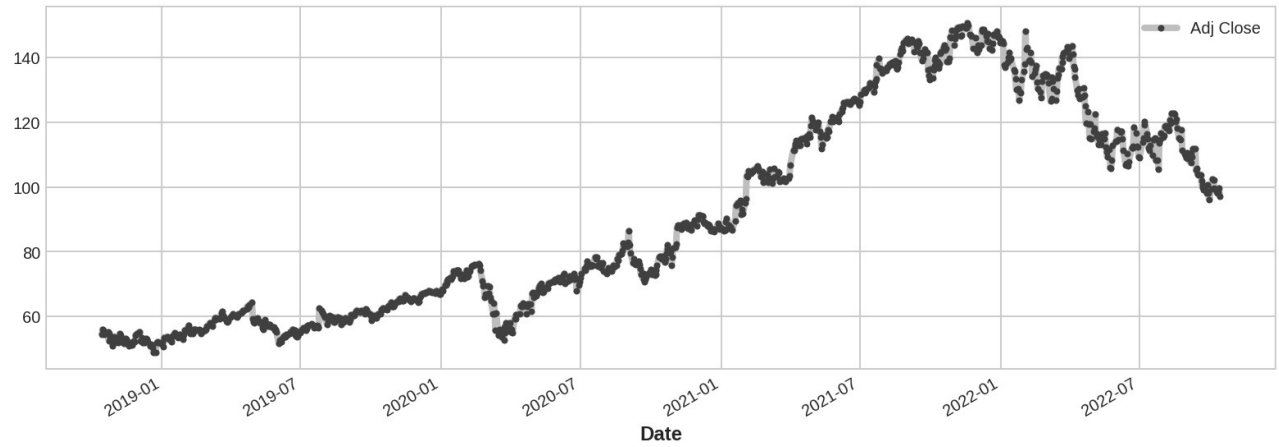

Figure 6: Plot for Google stock prices (over 4 years)

It is evident from the graph above, that the trend, ignoring any small term effects, is increasing. But how do we determine the nature of the trend? Is it linear, quadratic, exponential and so on? To answer these questions, we must first get an idea from the plot of the moving average of our time series data.

Having applied linear, quadratic, cubic and bi-quadratic trend models, we found that bi-quadratic proves to be a promising choice. While the test-error using linear trend is the least among all, it will not account for the dip in the prices in future.

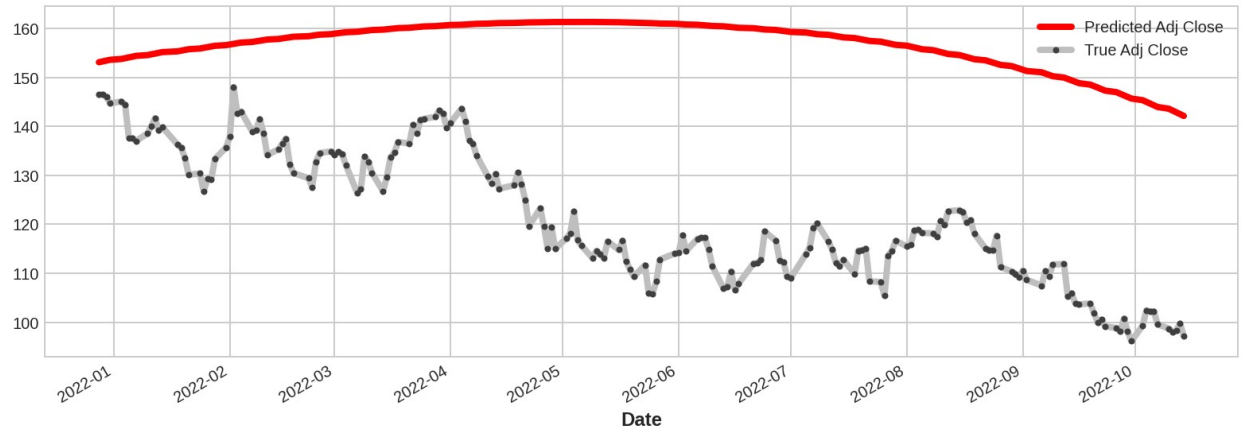

Figure 7: Plot for Predicted vs True values using bi-quadratic trend

Mean squared error = 1401.636686924792

This model is still not complete, since this only predicts the trend of our time series (one of the 3 components of a typical time series).

Given below is a trend forecast for 30 days using above trend model.

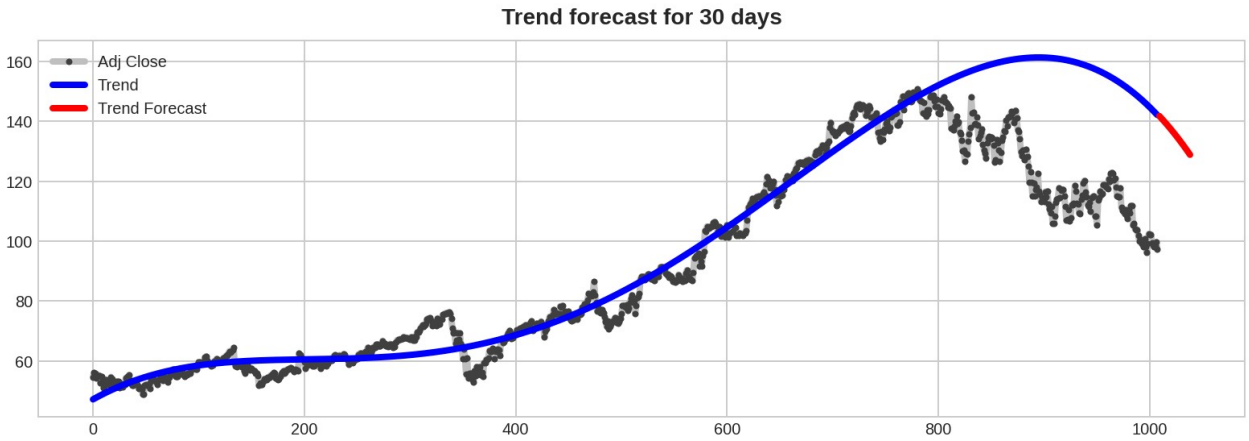

Figure 8: Trend forecast plot for 30 days