Name: Moaz Nabeel

Capstone-2: Anonymous Product Forecasting

Problem Statement:

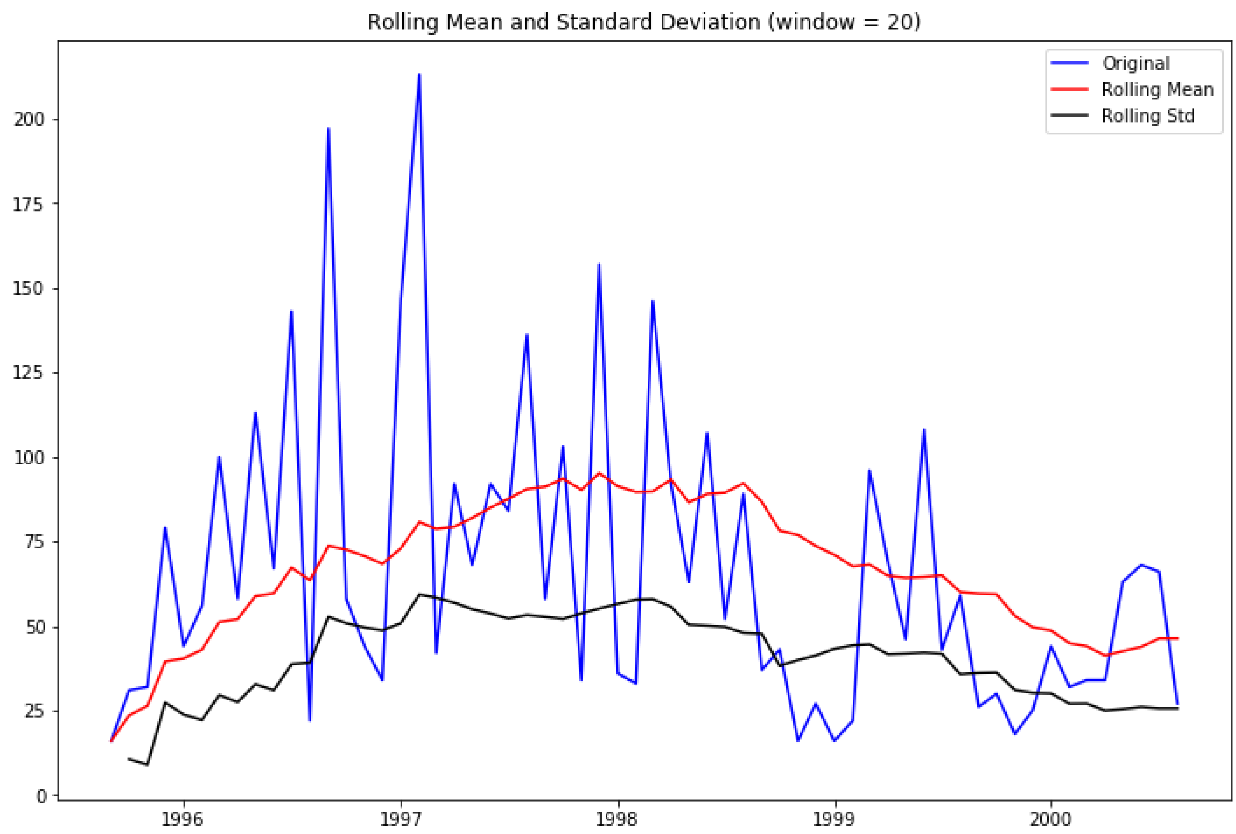
The client for the data is a production planning facility and forecasting for the anonymous product is necessary for production, planning and control. The problem will be solved using traditional time forecasting techniques such as Holt's Linear Method, Holt's Winter Method, ARIMA and SARIMA as well as deep learning technqies such as LSTM network and ConvLSTM. First four years of data will be used to train the methods and neural networks mentioned above. Forecasts will be made on the last year of the dataset and the error metric will be RMSE (root mean squared error).

Dataset Description:

The dataset used in this project is provided by Southern Illinois University Edwardsville and shows the product demand for five years. The data is a univariate time series in nature. The dataset did not need a lot of cleaning or wrangling. However, reading in the dates had to be parsed using the pandas.read_csv command.The dataset did not have any missing values or dates.

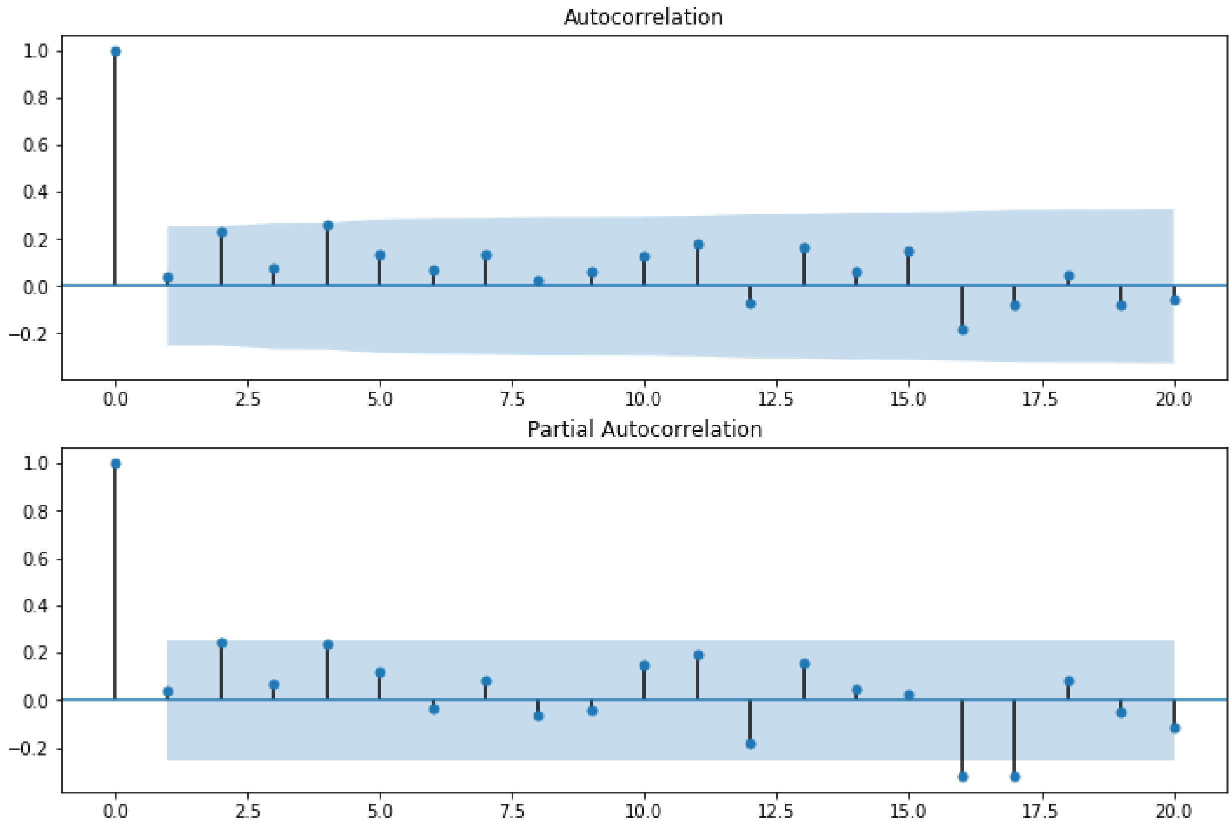
Data Story:

Since the data is a univariate time series data, not much story can be made. However the trend of demand for the product can be shown in the figure below along with the rolling means and standard deviations.



Exploratory Data Analysis:

For time series data it is highly recommended to see the ACF and PACF plots to check if there is autocorrelation in the data. The plots are shown below.



Also, in time series data it is recommended to check if the data is stationary or not. To check for stationarity, Dickey-Fuller test was done and the output of the p-value was below 0.05 and that proved that the data is stationary. The output of the test is shown below.

ADF Statistic: -4.146829

p-value: 0.000810

Critical Values:

1%: -3.548

5%: -2.913

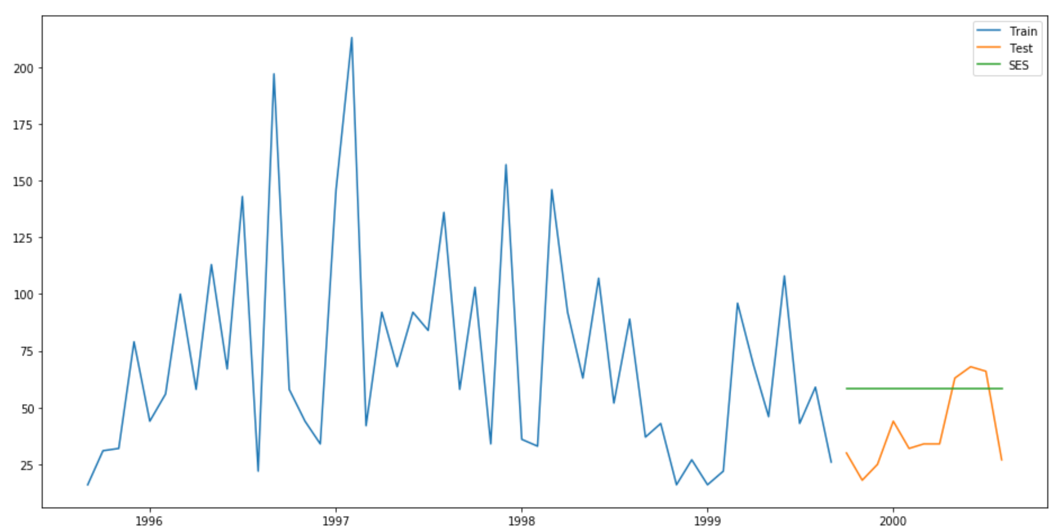
10%: -2.594

Traditional Time Series Forecasting:

In this section, we will use traditional techniques used for time series forecasting in the industry and these models will serve as benchmark or baseline model for the deep learning algorithms in the next section.

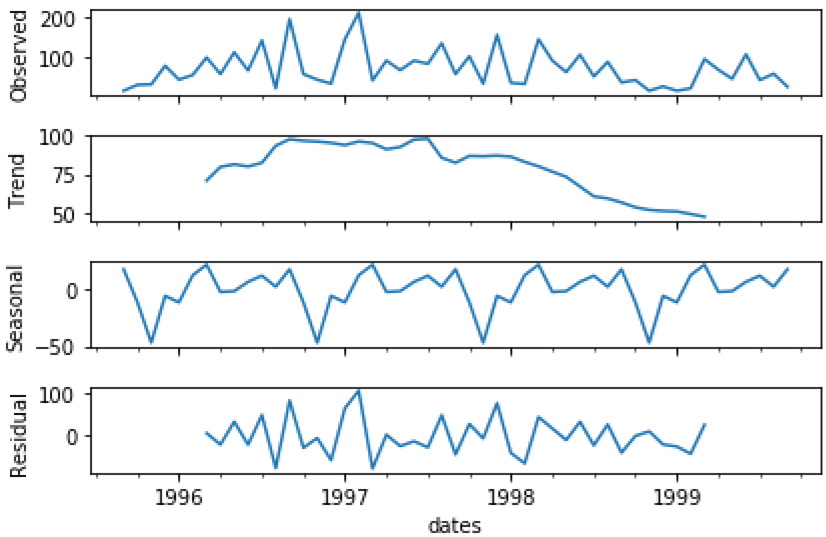
Simple Exponential Smoothing:

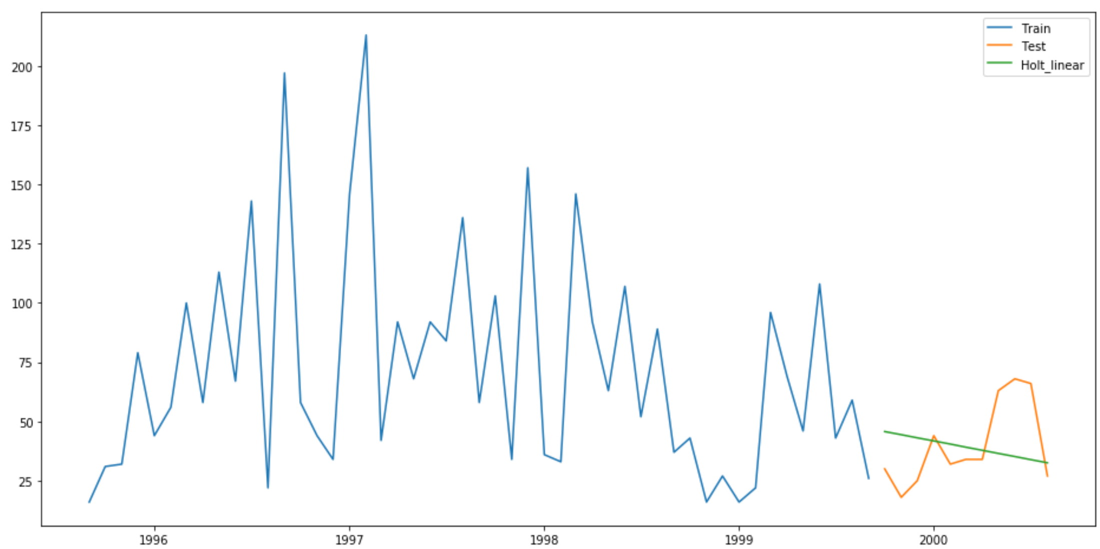
The simplest of the exponential smoothing methods is naturally called **simple exponential smoothing** (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern.The plot for our data with predictions and actual data for the last year is shown below. The root mean squared error was 24.8.



Holt’s Linear Trend:

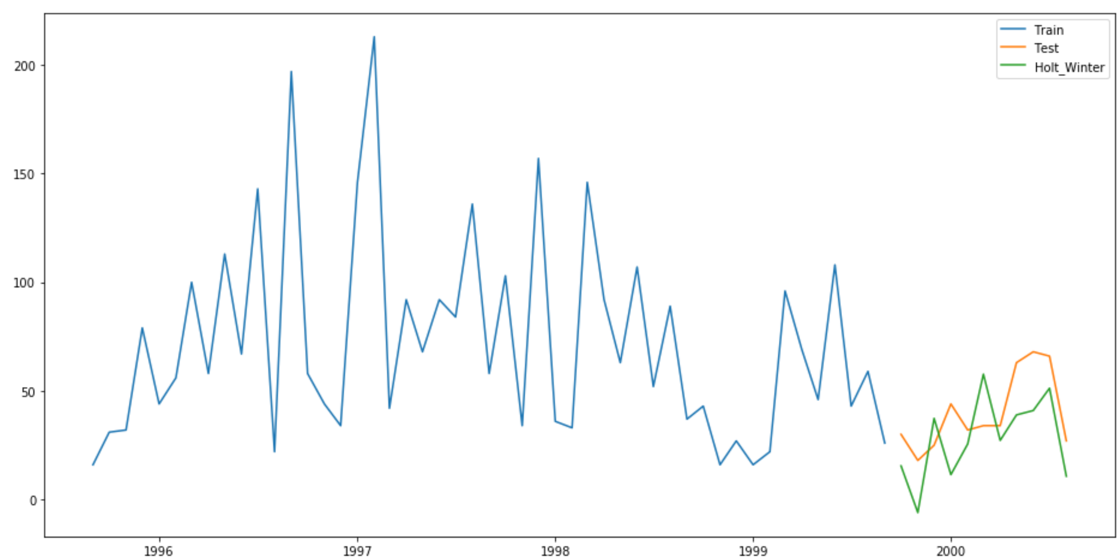
Holt extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend). The forecast and the original data is shown in the plot below along with common time series data factors such as trend , seasonality and residuals. The root mean squared error was 19.6.





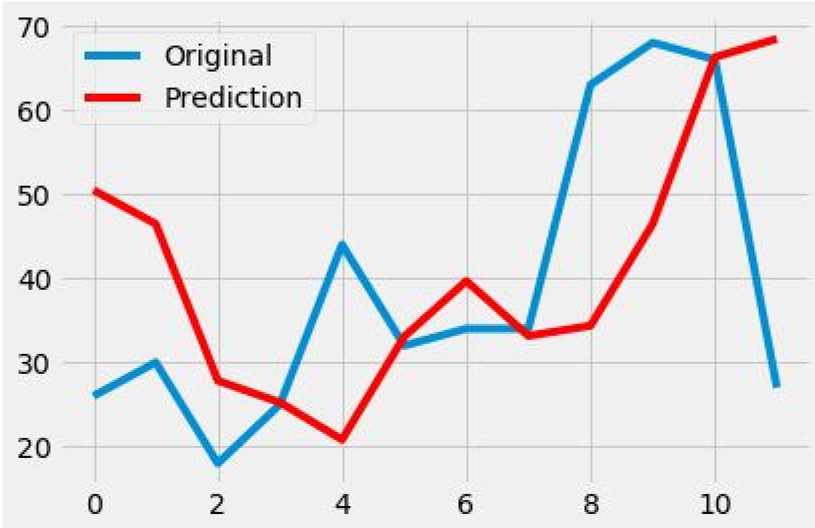
Holt's Winter Method:

Holt and Winters extended Holt's method to capture seasonality. The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level ℓ_t, one for the trend b_t, and one for the seasonal component s_t, with corresponding smoothing parameters α, β and γ. We use m to denote the frequency of the seasonality, i.e., the number of seasons in a year. For example, for quarterly data m=4, and for monthly data m=12. The plot of the forecasted and original data is shown below and the root mean squared error was 20.09.



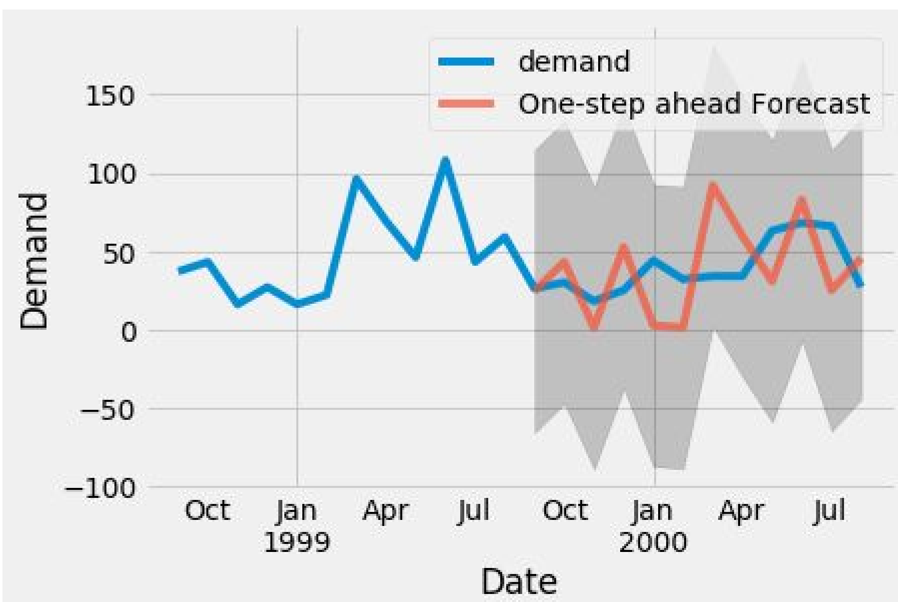
ARIMA:

ARIMA (autoregressive integrated moving average) method developed by Box and Jenkins (1976). In this method, a mathematical model is fitted that is optimal with respect to the historical time series data. The for simple ARIMA is shown below and the root mean squared error was 19.45.

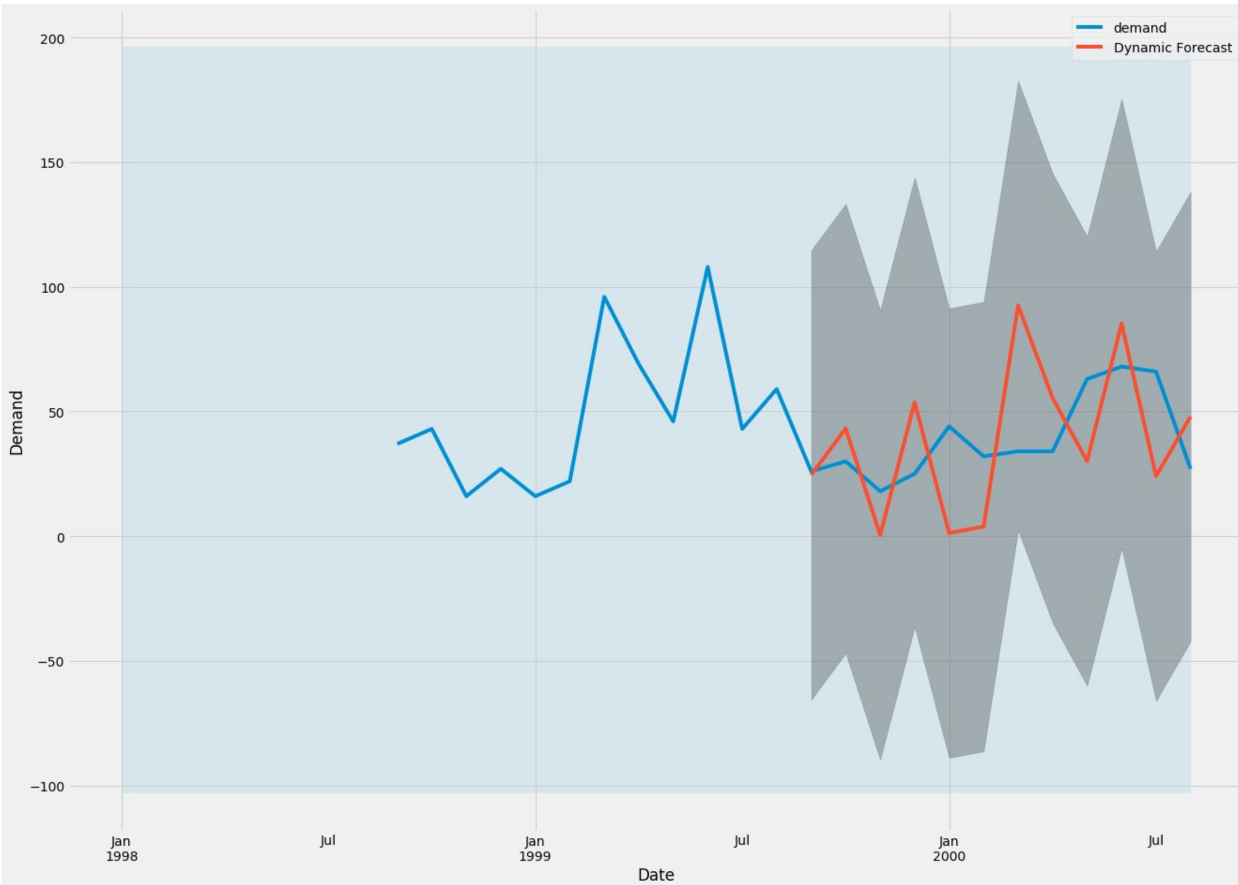


SARIMA:

SARIMA is just like ARIMA except that it adds the seasonality along with the trend. With SARIMA the test set error was 30.56. The forecasted plot for one step ahead forecast have been shown below.



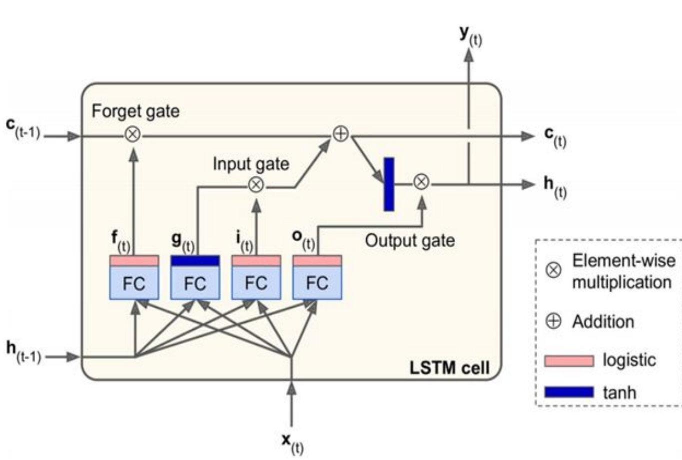
The forecasted plot for dynamic forecast with SARIMA is shown below and its RMSE was slightly higher than the one step ahead SARIMA forecast of about 30.82.



Deep Learning Forecasting:

LSTM:

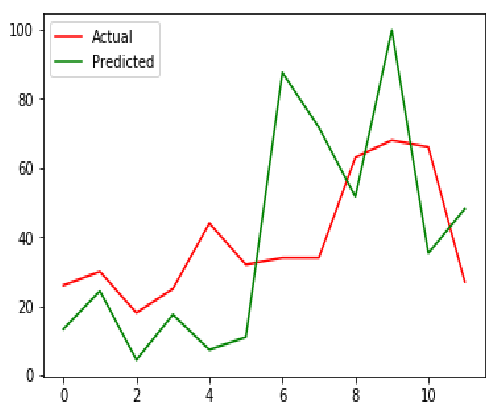
The architecture of LSTM is shown below. LSTM cell looks exactly like a regular cell, except that its state is split in two vectors: h(t) and c(t) ("c" stands for "cell"). You can think of h(t) as the short-term state and c(t) as the

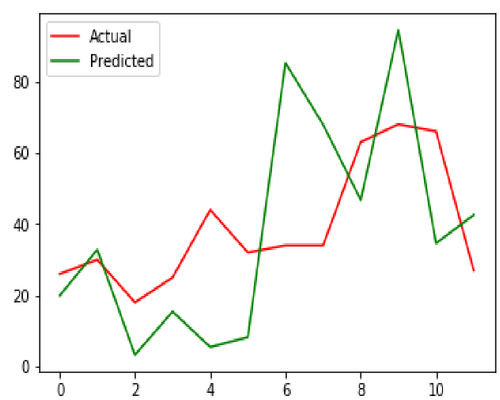


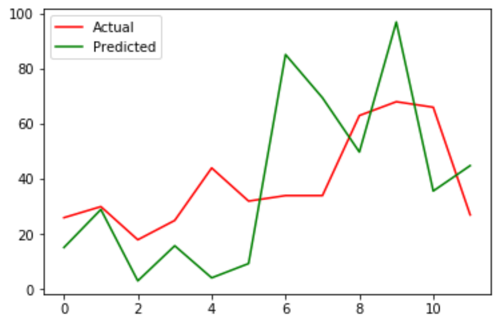
long-term state. Now let's open the box! The key idea is that the network can learn what to store in the long-term state, what to throw away, and what to read from it. As the long-term state c(t–1) traverses the network from left to right, you can see that it first goes through a forget gate, dropping some memories, and then it adds some new memories via the addition operation (which adds the memories that were selected by an input gate). The result c(t) is sent straight out, without any further transformation. So, at each time step, some memories are dropped and some memories are added.

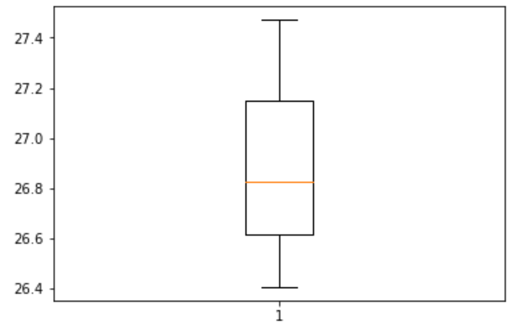
Results from LSTM:

The model developed for the data set used had 8 layers, first was the input layer, and from second layer to seventh layer, hidden layers were added with 'relu' activation function. The optimizer used in the output layer was 'adam', which is an industry standard. Validation of the forecast with respect to the actual data was done three times. And the plots are shown below.







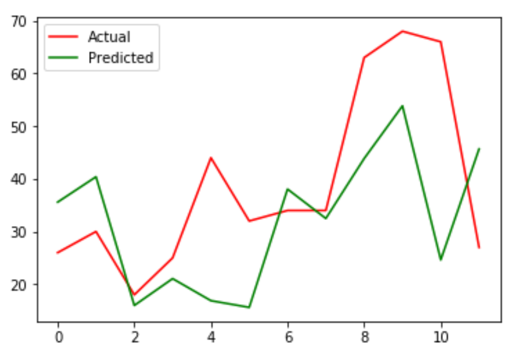


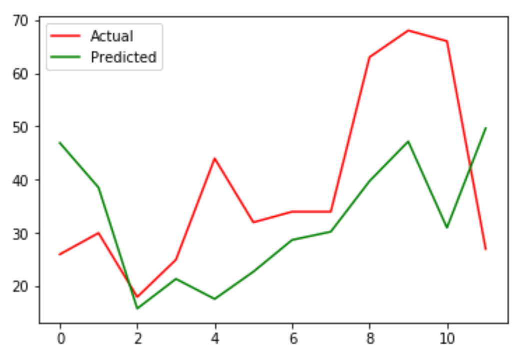
LSTM: 26.900 RMSE (+/- 0.438)

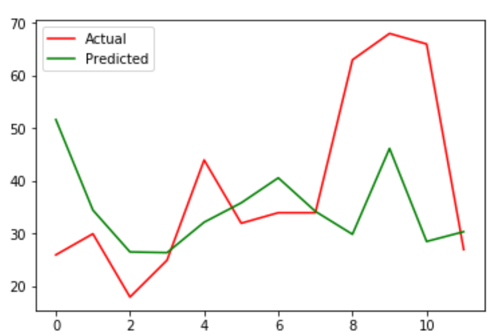
<u>ConvLSTM/CNN-LSTM:</u>

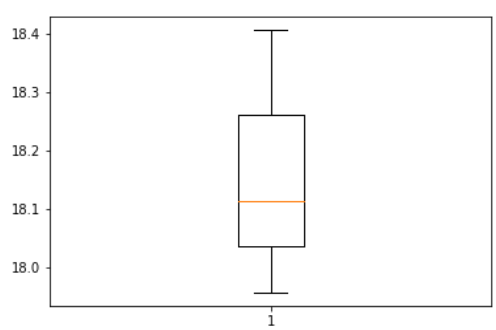
ConvLSTM is combination of CNNs and LSTMs where the LSTM units read input data using the convolutional process of a CNN.

<u>Results of ConvLSTM:</u>









CNN-LSTM: 18.159 RMSE (+/- 0.186)

Comparison Table:

MODEL/ ALGORITHM	**RMSE**
Simple Exponential Smoothing (SES)	24.81
Holt's Linear Trend (HLT)	19.61
Holt’s Winter Method	20.09
ARIMA	19.45
SARIMA	30.56
LSTM	26.9
CNN-LSTM	18.16

Conclusion:

Overall, with univariate time series data traditional techniques performed quite well. However, convolution long-short term neural networks were able to perform the best out of the models and algorithms used. But if we look at forecasting for univariate time series data ARIMA did quite well and for a lot less computational power. In conclusion, deep learning technique such as Conv-LSTM is quite useful for time series forecasting and can handle multivariate time series better than traditional techniques.