**Time Series Analysis of S&P 500**

A. Pawan DIXIT1, Shubham CHITKARA2, Rahul SHARMA3

1. **Introduction**

Stock price forecasting is a popular and important topic in financial and academic studies. Time series analysis is the most common and fundamental method used to perform this task.

A proper analysis of stock data will give the investor better gains. This study will show how machine learning can be used to guide an investor’s decision-making process. The existing methodologies focus on linear models viz. AR (Autoregressive), MA (Moving Average), ARMA (Autoregressive moving average), ARIMA (Autoregressive integrated moving average) and other non- linear models which we are not discussing because it is out of scope of our work. These models are not capable of identifying the complex interactions and latent dynamics existing within the data. Applying Deep learning methods to these types of data will give more accurate results than the existing methods. Deep learning architectures can identify the hidden patterns in the data and is also capable of exploiting the interactions existing within the data, which is, at least not possible by the existing financial models. The deep learning architectures that we are demonstrating in our project report is RNN (Recurrent Neural Networks) which can be used to predict stock price. The performance of the models can be quantified using error percentage.

This project aims to combine the conventional time series analysis technique with information from the S&P 500 stock data gathered from Wall Street Journal website to predict daily changes in stock price. The Standard & Poor's 500, often abbreviated as the S&P 500, or just the S&P, is an American stock market index based on the market capitalizations of 500 large companies having common stock listed on the NYSE or NASDAQ.

1. **Methodology**

**2.1 Key Concepts**

* **Stationary Series -** A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time. The mean of the series should not be a function of time rather should be a constant. The variance of the series should not a be a function of time. This property is known as homoscedasticity. The covariance of the ith term and the (i + m)th term should not be a function of time. Until and unless a time series is stationary, we cannot build a time series model. In cases where the stationary criterion is violated, the first requisite becomes to stationarize the time series and then try stochastic models to predict this time series. There are multiple ways of bringing this stationarity. Some of them are Detrending, Differencing etc.
* **Non-Stationary Series -** Non-stationary data, are unpredictable and cannot be modeled or forecasted. The results obtained by using non-stationary time series may be spurious in that they may indicate a relationship between two variables where one does not exist. In contrast to the non-stationary process that has a variable variance and a mean that does not remain near, or returns to a long-run mean over time, the stationary process reverts around a constant long-term mean and has a constant variance independent of time. Examples of non-stationary processes are random walk with or without a drift (a slow steady change) and deterministic trends (trends that are constant, positive or negative, independent of time for the whole life of the series).
* **Random walk -** A random walk is a mathematical object, known as a stochastic or random process, that describes a path that consists of a succession of random steps on some mathematical space such as the integers that follow no discernible pattern or trend. Although, a random walk process has a constant mean over time, it is not a stationary process because it has a time dependent variance and covariance. A Random walk with a drift and a trend is a stochastic trend series.
* **Autocorrelation -** is the correlation of a signal with a delayed copy of itself as a function of delay. Informally, it is the similarity between observations as a function of the time lag between them.
* **Dickey-Fuller Test -** In statistics and econometrics, an augmented Dickey–Fuller test (ADF) tests the null hypothesis that a unit root is present in a time series sample. A unit root test tests whether a time series variable is non-stationary and possesses a unit root. The null hypothesis is generally defined as the presence of a unit root and the alternative hypothesis is either stationarity or trend stationarity or explosive root depending on the test used.
* **ARIMA -** Stands for Autoregressive Integrated Moving Average. It is a generalization of an autoregressive moving average(ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

The autoregressive model specifies that the output variable depends linearly on its own previous values

AR(p): Yt = β0 + β1Yt-1 + β2Yt-2 + ... + βpYt-p + et

Moving average model is a common approach for modeling univariate time series. It specifies that the output variable depends linearly on the current and various past values.

MA (q): Yt = µ + et + θ1et-1 + θ2et-2 + ... + θq et-q

ARIMA models are generally denoted ARIMA (*p*, *d*, *q*) where parameters *p*, *d*, and *q* are non-negative integers, *p* is the order (number of time lags) of the autoregressive model, *d* is the degree of differencing (the number of times the data have had past values subtracted), and *q* is the order of the moving average model.

* **RNN** - A recursive neural network (RNN) is a kind of deep neural network created by applying the same set of weights recursively over a structured input, to produce a structured prediction over variable-size input structures, or a scalar prediction on it, by traversing a given structure in topological order. An RNN deals with sequence problems because its connections form a directed cycle. In other words, it can retain state from one iteration to the next by using its own output as input for the next step. In programming terms this is like running a fixed program with certain inputs and some internal variables. The simplest recurrent neural network can be viewed as a fully connected neural network if we unroll the time axes.

1. **Result**

**3.1 Exploring Data**

The S&P 500 daily stock data was downloaded from Wall Street Journal dating from January 1995 to March 2018. The below figure shows the daily closing stock prices for the given data.

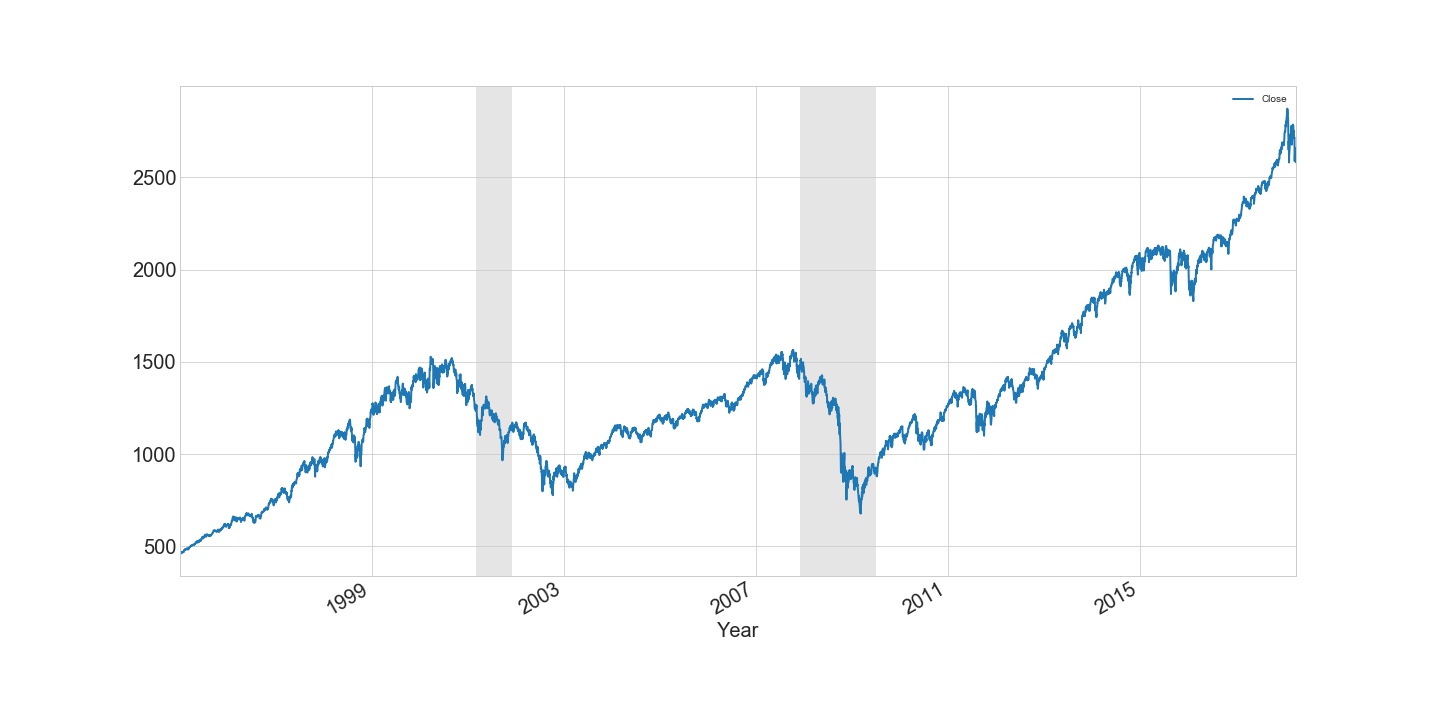


Figure 1-S&P 500 daily index

As we can see that stock prices went down significantly during the periods that are marked grey in the figure above. These represent periods of recession in United States during 2001 & 2008.

We took the moving average of the data over 1 year period to make the series smooth and to see the overall trend over last 23 years.

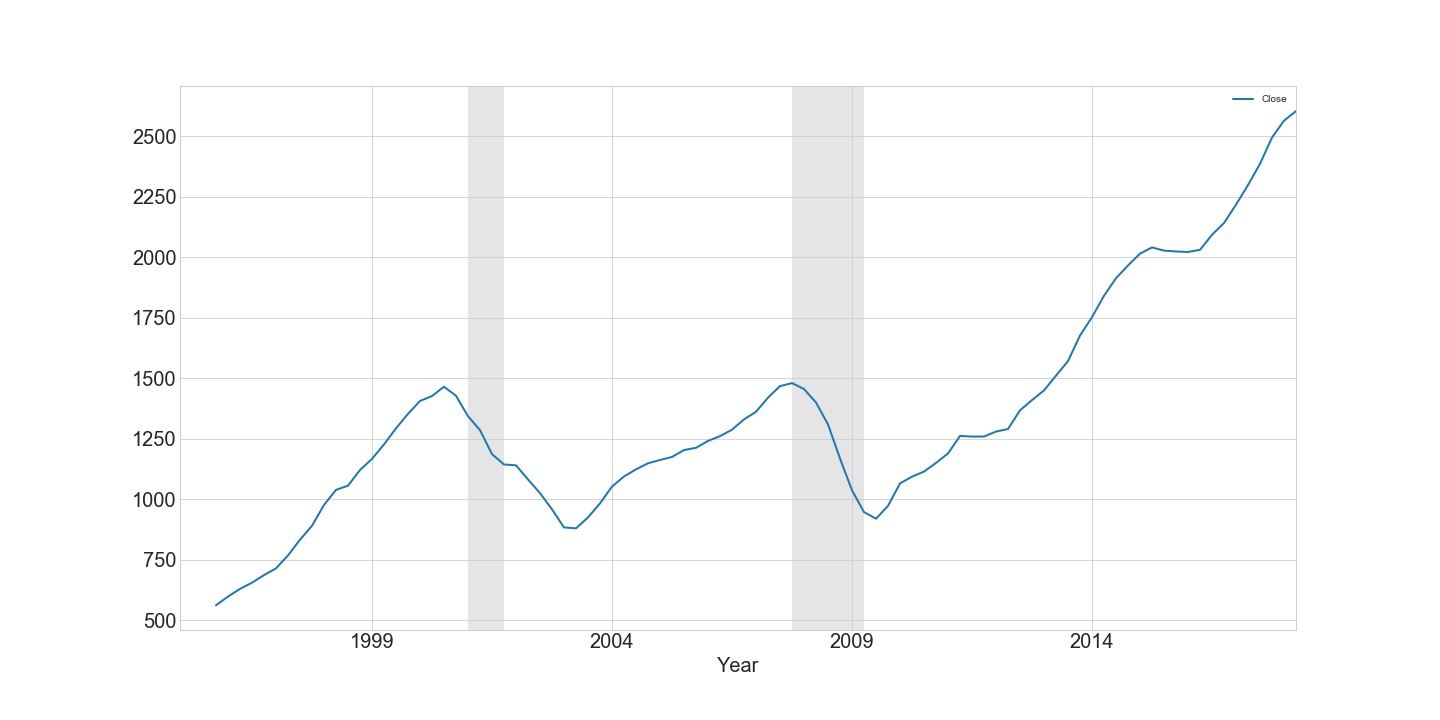


Figure 2 - S&P 500 Moving average (1-year)

From the above figure, we can say that there is an overall upward trend in the stock prices except for during the recession period which kind of makes sense as the economy went down as well.

So, after seeing these plots we went ahead to see if we can find any relationship between US’s economy and S&P 500. We normalized quarterly GDP of United States and quarterly prices of S&P 500 to bring both of them to the same scale.

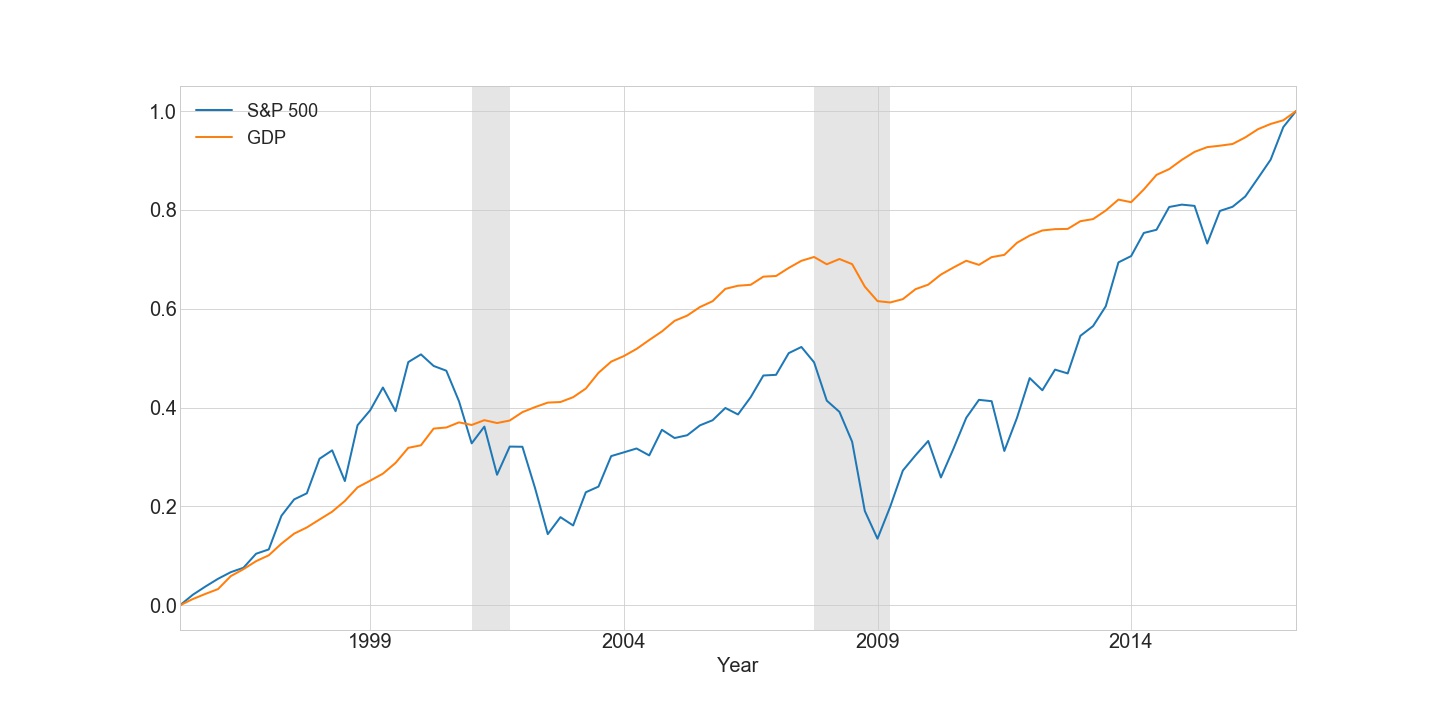
****

Figure 3 - Normalized S&P 500 quarterly vs Normalized GDP quarterly

Although at first it might look as if there is no significant relationship between these two but on observing carefully, we can see that even small changes in GDP inflicts significant changes in S&P 500. During the 2001 recession the GDP goes flat from an increasing state and simultaneously S&P 500 stock index saw a significant decline. Similarly, during the 2008 recession, the economy went down and S&P Index also declined to a great extent. Since then GDP has always been increasing and so are S&P 500 Indices. This is probably because the stock prices are more volatile than a country’s GDP.

Exploring the data further, we looked for autocorrelation in the series using autocorrelation plot.

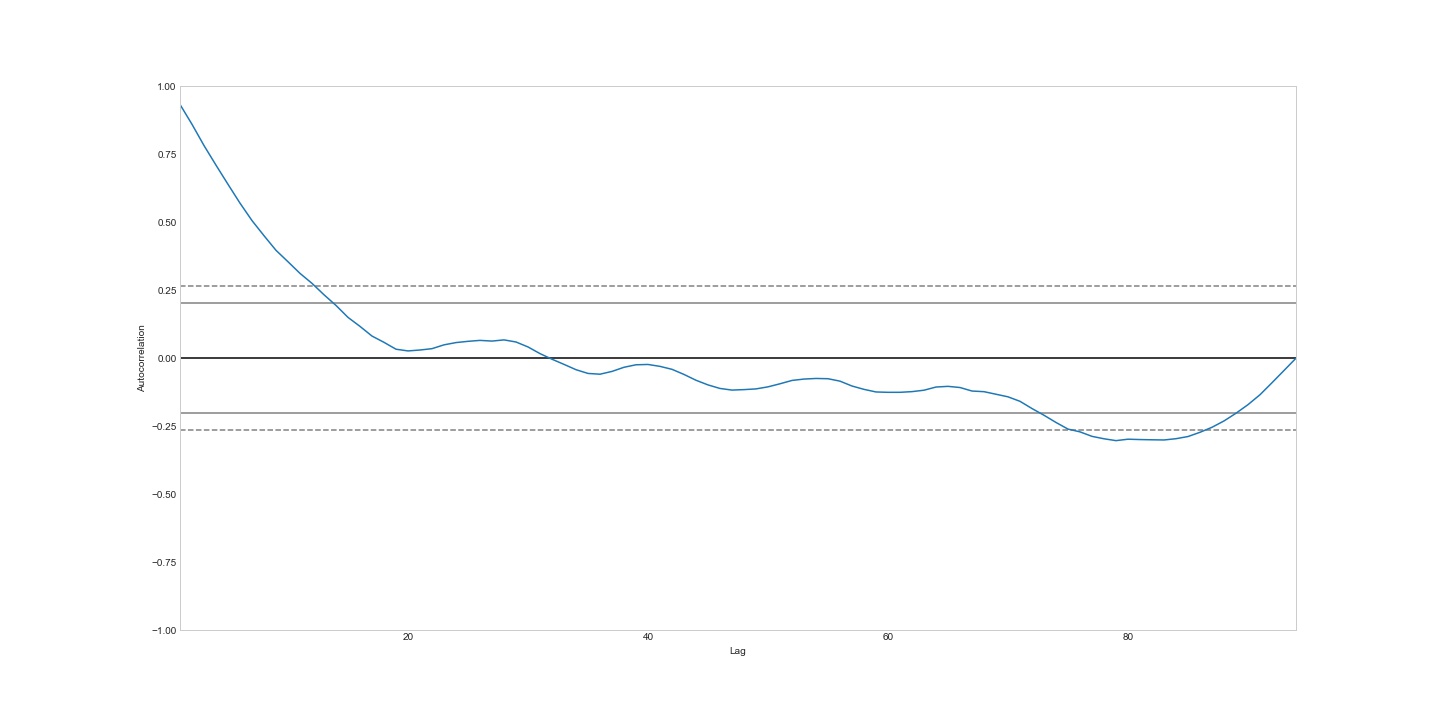


Figure 4 - Autocorrelation of quarterly S&P 500 index

The first few lags in the series are highly correlated which gives the indication of a stochastic trend. Further we used the Dickey-Fuller test to check for stationarity of the series which indicated that the series is non-stationary and confirmed the stochastic trend. To remove this stochastic trend, we took the first order difference of the series which turned out to be stationary. The below graph represents the first order differenced series. It is clear that none of the lags are highly correlated.

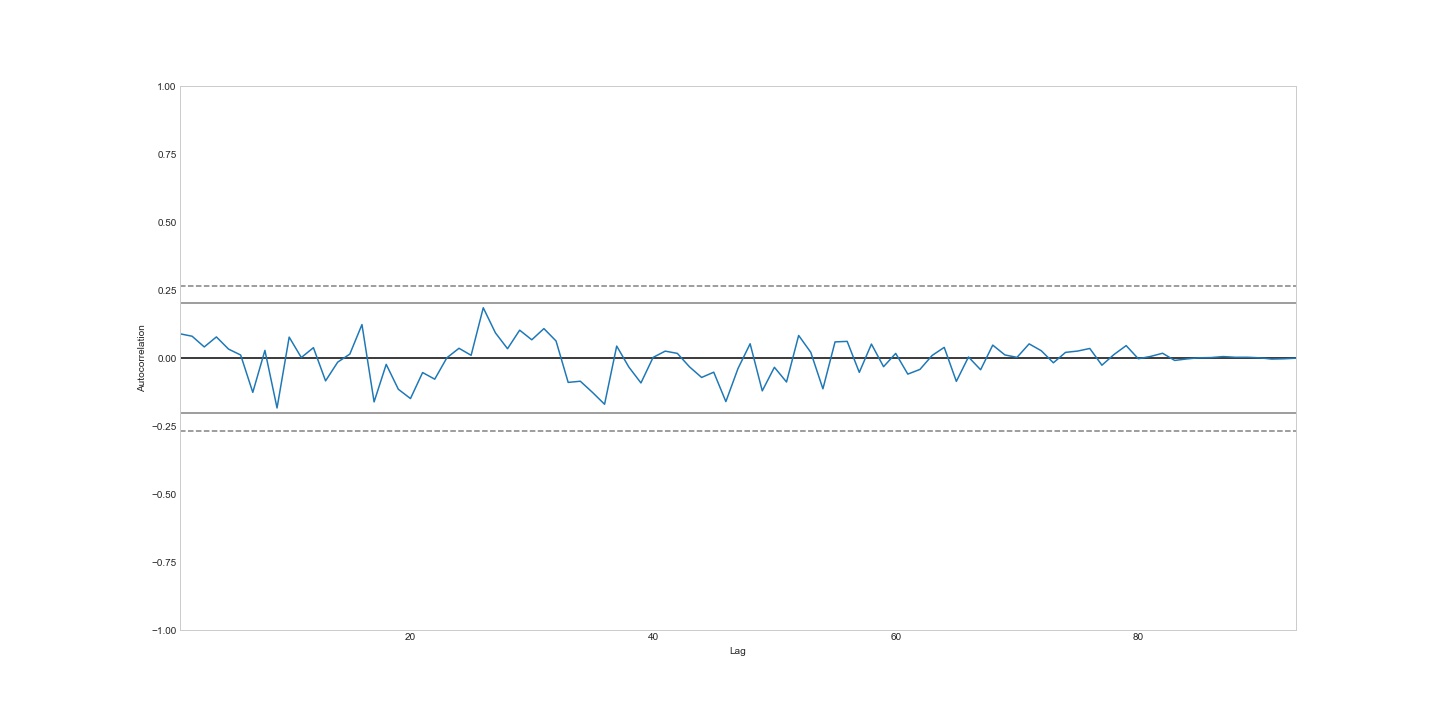


Figure 5 - Auto-correlation plot of first order difference of S&P 500 index

Now that we know that the first order differenced series is stationary, we can go ahead and model the series using ARIMA. We used the *auto.arima* function in the *forecast* package of R to find the best ARIMA model for this time-series on the basis of BIC (Bayesian Information Criterion). After creating the model, we used one-step ahead recursive forecast method for prediction. The figure below shows the original data vs the forecasted values for the series.

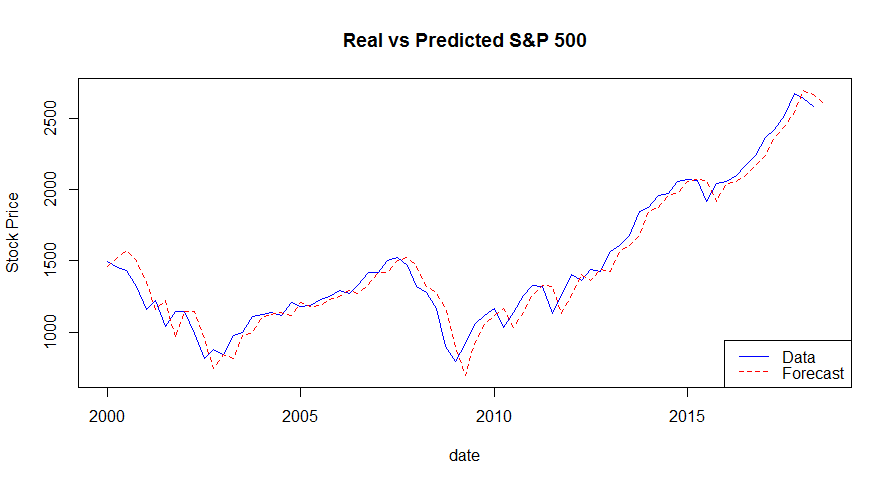


Figure 6 - ARIMA Model and its Forecast

The Mean Squared Forecast Error value for this ARIMA model was 10835.62

We then created a Recursive Neural Network (RNN) model to forecast the time-series using *tensorflow* package in Python. The RNN had 1 hidden layer with 500 neurons and a learning rate of 0.001. This model used a basic RNN cell and it ran for 200 epochs. The figure below shows the original data vs the forecasted values for the series.

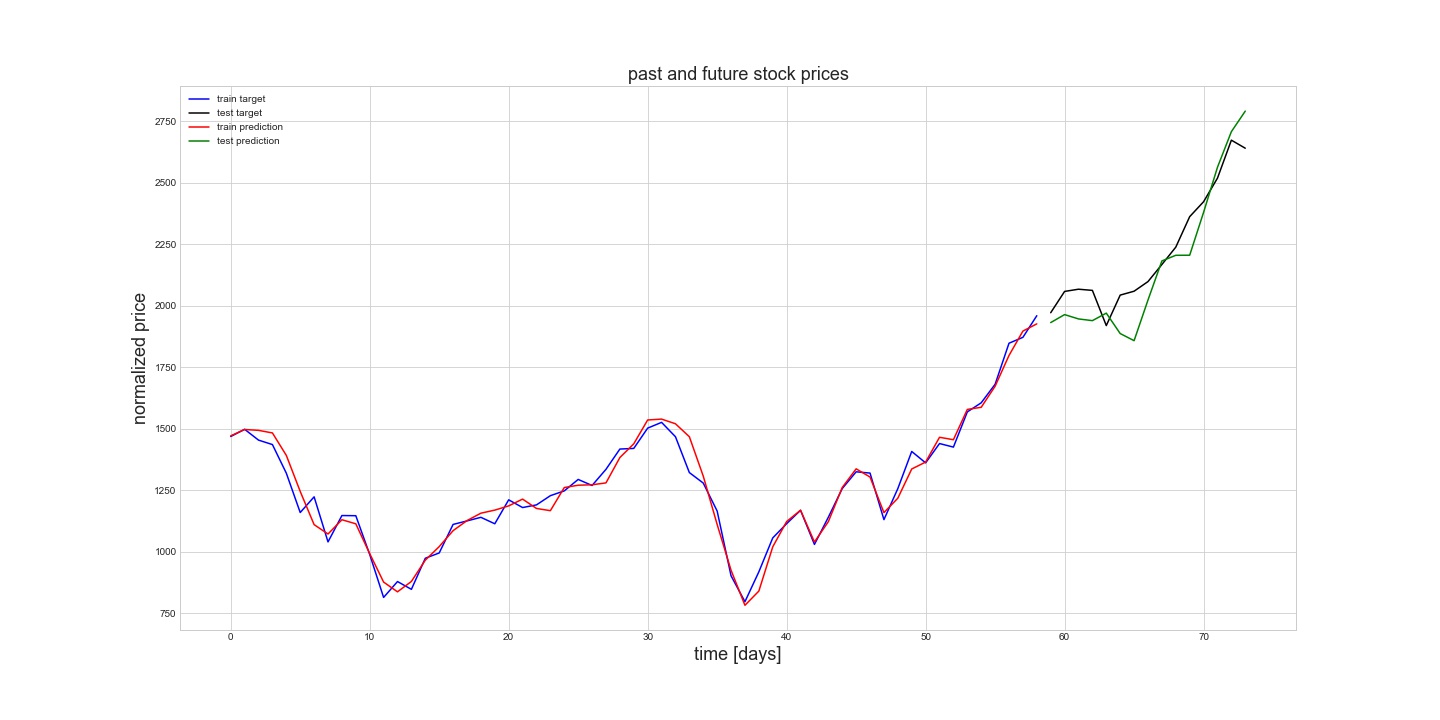


Figure 7 - RNN Model and its forecast

The Mean Squared Forecast Error value for this RNN model was 1860.64

**Conclusion**

To sum up things we found that the S&P 500 index increased from 459.11 to 2581.88 which clearly shows there is an increasing trend. On further investigation, we found that it was a stochastic trend so we used first order differencing to make it stationary.

We created ARIMA and RNN models to forecast S&P 500 series and found that RNN outperforms ARIMA on the basis of Mean Squared Forecast Error (MSFE). So if we have to forecast the stock prices just on the basis of previous data we would recommend Investors to use our RNN model over ARIMA.

On an end note we would like to mention that it is never easy to forecast stock indices in real world because it is not just the trend but also various other factors that affect the stock prices which are even more difficult to predict. Had it been easy to predict stock prices, everyone would have been a billionaire. So, it is always better to have insights of an economist along with the results of statistical model to get a good forecast.

**References**

1. Time Series Analysis and Its Applications, Robert H. Shumway, David S Stuffer, 2006
2. Introduction to Econometrics, Stock, J.H. & M.W. Watson, 2014
3. <https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>
4. [https://en.wikipedia.org/](https://en.wikipedia.org/https://www.pyimagesearch.com/2016/09/26/a-simple-neural-network-with-python-and-keras// )
5. [https://www.pyimagesearch.com/2016/09/26/a-simple-neural-network-with-python-and-keras//](https://en.wikipedia.org/https://www.pyimagesearch.com/2016/09/26/a-simple-neural-network-with-python-and-keras// )
6. <https://iamtrask.github.io/2015/11/15/anyone-can-code-lstm//>
7. <https://blog.statsbot.co/time-series-prediction-using-recurrent-neural-networks-lstms-807fa6ca7f>
8. <https://www.investopedia.com/articles/trading/07/stationary.asp>