Google Stock Prediction and Time Series Analysis

Aswathnarayan Muthukrishnan Kirubakaran

Meenakshi Muthiah

Stock Prediction

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit.

Project Description

In this project, we use a neural network to predict the google stocks. It is a regression problem to predict the google stock. Since it is a time series data, we are using recurrent neural network. Also, we performed the Time series forecast.

Files included

- Google_Stock_Price_Test: Test dataset of google stock price.
- Google_Stock_Price_Train: Train dataset of google stock price.
- Recurrent Neural Network.ipnb: Complete script used to train and test the model.
- TimeSeriesAnalyis.ipnb: Complete script used for time series analysis.

Model Architecture Design

It is a recurrent neural network which works well for sequence or time series dataset. It is a five-layer network which uses LSTM to avoid long term memory issues.

- Scaled the features MinMaxScaler with a range of 0 to 1.
- Created data structure with 60 timesteps and 1 output.

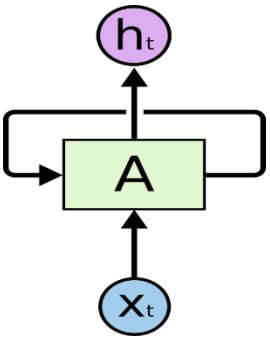
The model looks as follows,

- LSTM (units: 50, return sequence)
- Dropout layer
- LSTM (units: 50, return sequence)

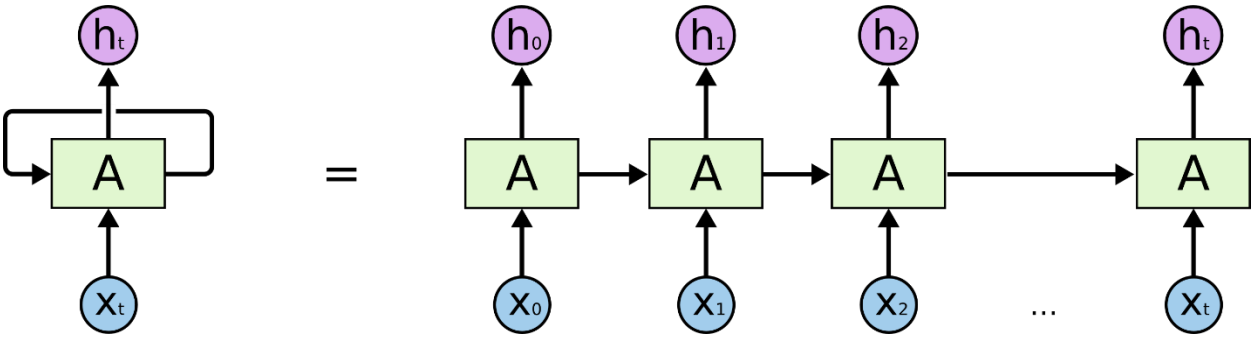
- Dropout layer
- LSTM (units: 50, return sequence)
- Dropout layer
- LSTM (units: 50)
- Dropout layer
- Output layer

Recurrent Neural Networks

Recurrent neural networks are networks with loops in them, allowing information to persist.



A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.



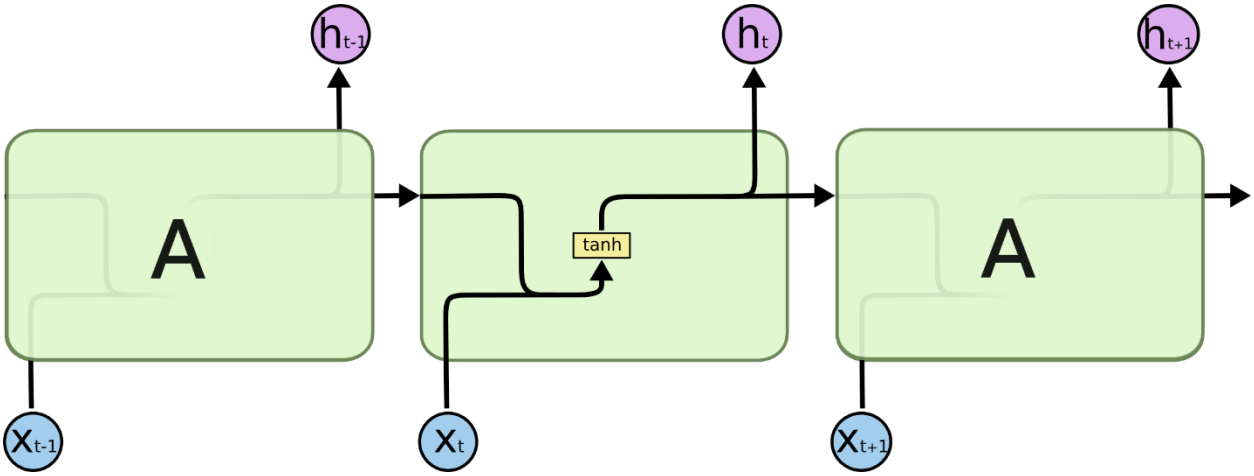
RNNs might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame.

Problem with RNN

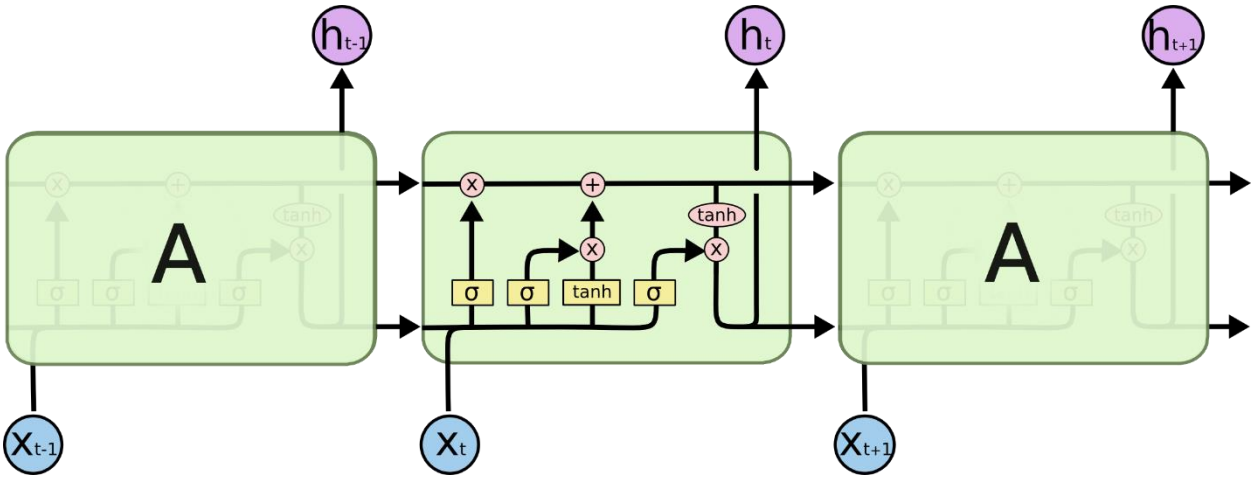
In cases, where the gap between the relevant information and the place that it's needed is small, RNNs can learn to use the past information. Unfortunately, as that gap grows, RNNs become unable to learn to connect the information.

Long Short-Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for extended periods of time is practically their default behavior, not something they struggle to learn. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.



LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



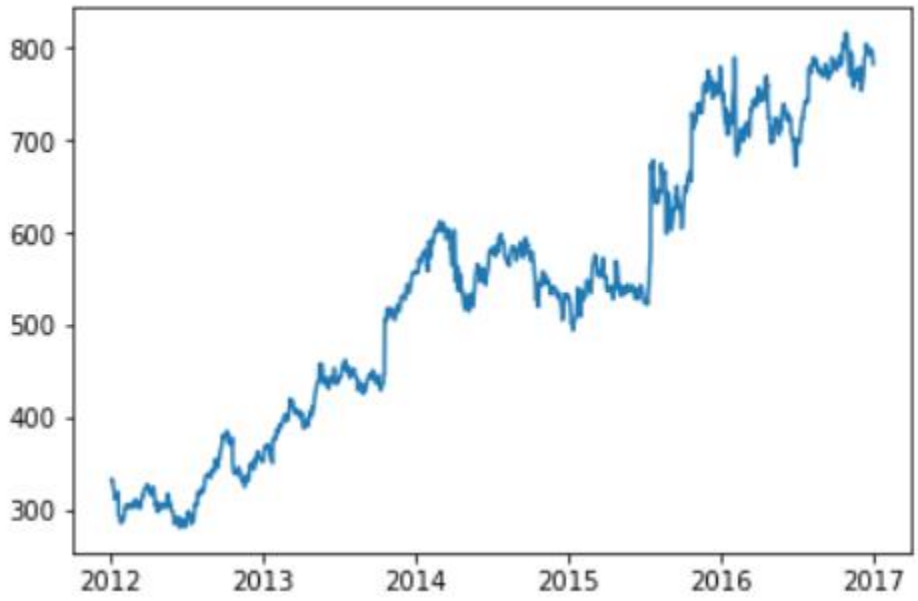
The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Time Series Analysis:

‘Time’ is the most crucial factor which ensures success in a business. It’s difficult to keep up with the pace of time. But, technology has developed some powerful methods using which we can ‘see things’ ahead of time. One such method, which deals with time based data is **Time Series Modeling**. As the

name suggests, it involves working on time (years, days, hours, minutes) based data, to derive hidden insights to make informed decision making.

Trend Analysis



A TS is said to be stationary if its statistical properties such as mean, variance remain constant over time. Most of the TS models work on the assumption that the TS is stationary. Intuitively, we can sat that if a TS has a particular behavior over time, there is a very high probability that it will follow the same in the future. Also, the theories related to stationary series are more mature and easier to implement as compared to non-stationary series.

Stationarity is defined using very strict criterion. However, for practical purposes we can assume the series to be stationary if it has constant statistical properties over time, i.e.. the following:

1.constant mean

2.constant variance

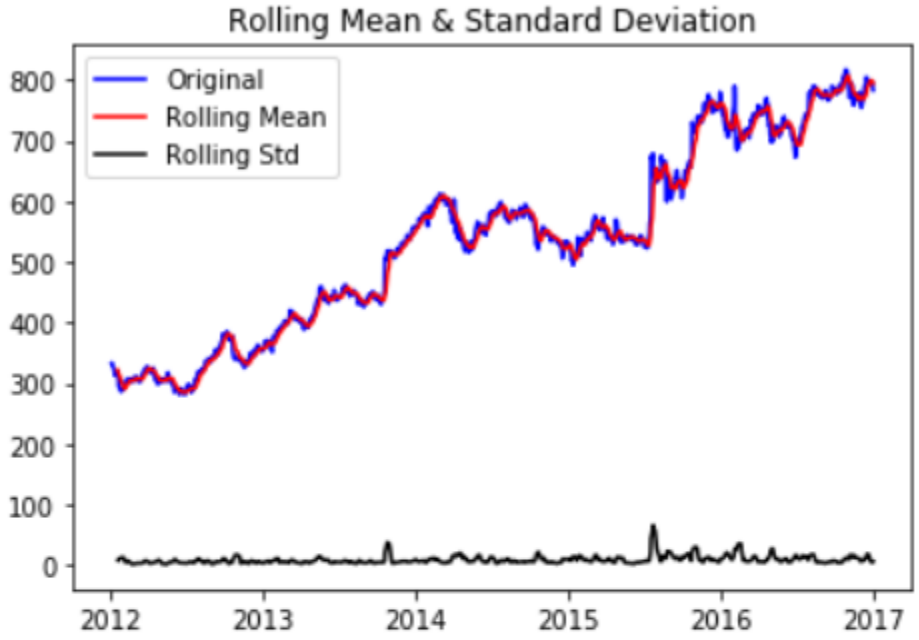
3.an autocovariance that does not depend on time.

Stationarity Check

It is clearly evident that there is an overall increasing trend in the data along with some seasonal variations. However, it might not always be possible to make such visual inferences. So, more formally, we can check stationarity using the following:

1.Plotting Rolling Statistics: We can plot the moving average or moving variance and see if it varies with time. By moving average/variance I mean that at any instant ‘t’, we’ll take the average/variance of the last year, i.e. last 12 months. But again this is more of a visual technique.

2.Dickey-Fuller Test: This is one of the statistical tests for checking stationarity. Here the null hypothesis is that the TS is non-stationary. The test results comprise of a Test Statistic and some Critical Values for difference confidence levels. If the 'Test Statistic' is less than the 'Critical Value', we can reject the null hypothesis and say that the series is stationary




```
Results of Dickey-Fuller Test:
Test Statistic                  -0.685244
p-value                          0.850535
#Lags Used                       4.000000
Number of Observations Used    1253.000000
Critical Value (1%)             -3.435580
Critical Value (5%)             -2.863849
Critical Value (10%)            -2.568000
dtype: float64
```

From the graph, we could find that the this is not stationary.

Making a time series stationary

Though stationarity assumption is taken in many TS models, almost none of practical time series are stationary. So, statisticians have figured out ways to make series stationary, which we'll discuss now. It's almost impossible to make a series perfectly stationary, but we try to take it as close as possible.

Let's understand what is making a TS non-stationary. There are 2 major reasons behind non-stationarity of a TS:

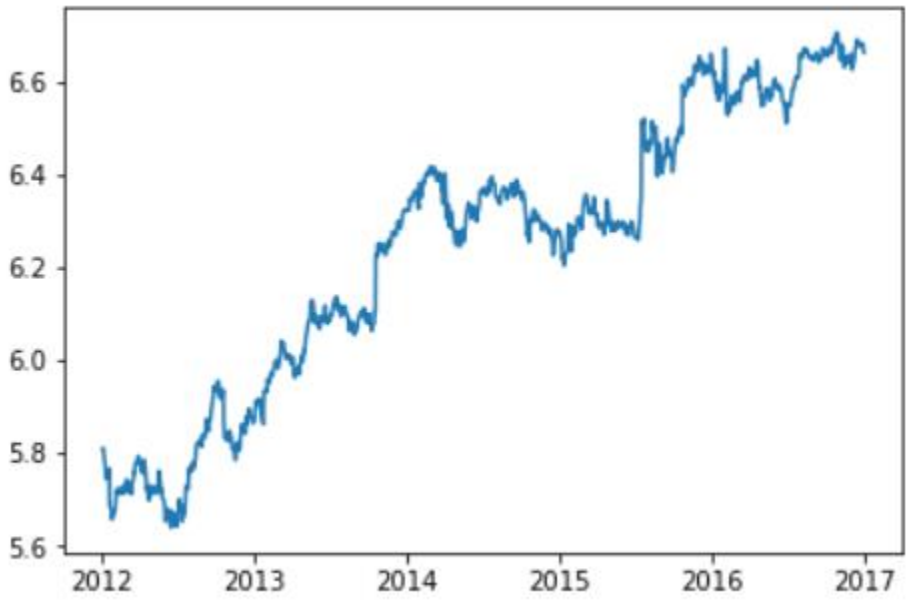
Trend – varying mean over time. For eg, in this case we saw that on average, the number of passengers was growing over time.

Seasonality – variations at specific time-frames. eg people might tend to buy cars in a month because of pay increment or festivals.

The underlying principle is to model or estimate the trend and seasonality in the series and remove those from the series to get a stationary series. Then statistical forecasting techniques can be implemented on this series. The last step would be to convert the forecasted values into the original scale by applying trend and seasonality constraints back.

Estimating and Eliminating Trend

One of the first tricks to reduce trend can be transformation. For example, in this case we can clearly see that the there is a significant positive trend. So, we can apply transformation which penalize higher values more than smaller values. These can be taking a log, square root, cube root, etc. Let's take a log transform here for *simplicity*:



In this simpler case, it is easy to see a forward trend in the data. But it's not very intuitive in presence of noise. So, we can use some techniques to estimate or model this trend and then remove it from the series. There can be many ways of doing it and some of most commonly used are:

Aggregation – taking average for a period like monthly/weekly averages

Smoothing – taking rolling averages

Polynomial Fitting – fit a regression model

Smoothing refers to taking rolling estimates, i.e. considering the past few instances.

Eliminating Trend and Seasonality

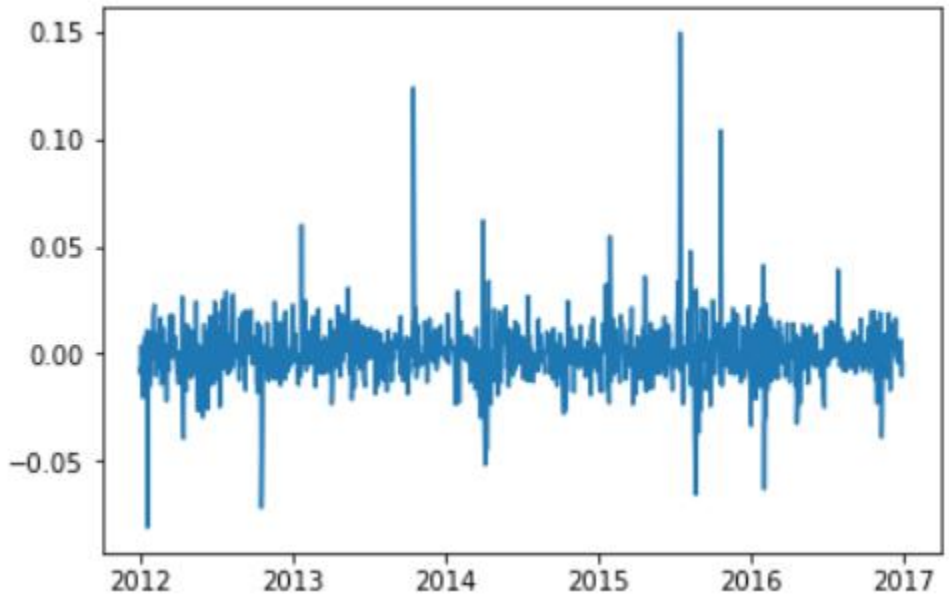
The simple trend reduction techniques discussed before don’t work in all cases, particularly the ones with high seasonality. Let’s discuss two ways of removing trend and seasonality:

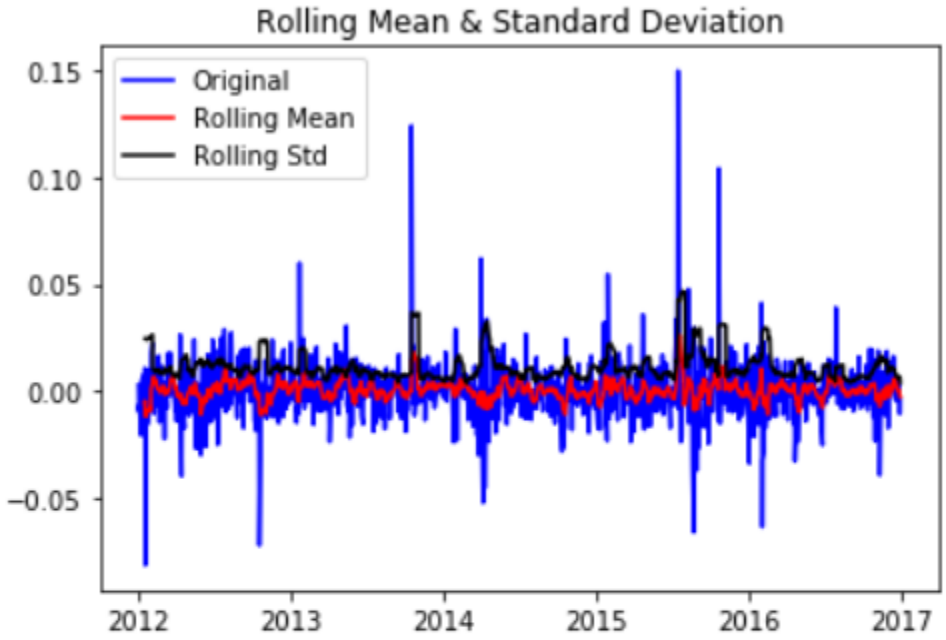
Differencing – taking the difference with a time lag

Decomposition – modeling both trend and seasonality and removing them from the model.

Differencing:

One of the most common methods of dealing with both trend and seasonality is differencing. In this technique, we take the difference of the observation at a instant with that at the previous instant. This mostly works well in improving stationarity.






```
Results of Dickey-Fuller Test:
Test Statistic                  -24.364365
p-value                           0.000000
#Lags Used                        1.000000
Number of Observations Used    1255.000000
Critical Value (1%)              -3.435571
Critical Value (5%)              -2.863846
Critical Value (10%)             -2.567998
```

We can see that the mean and standard variations have small variations with time. Also, the Dickey-Fuller test statistic is less than the 20% critical value, thus the TS is stationary with 90% confidence.

OUTCOMES

Forecasting a Time Series

Having performed the trend and seasonality estimation techniques, there can be two situations:

A strictly stationary series with no dependence among the values. This is the easy case wherein we can model the residuals as white noise. But this is very rare.

A series with significant dependence among values. In this case we need to use some statistical models like ARIMA to forecast the data.

Let me give you a brief introduction to ARIMA. I won't go into the technical details but you should understand these concepts in detail if you wish to apply them more effectively. ARIMA stands for Auto-Regressive Integrated Moving Averages. The ARIMA forecasting for a stationary time series is nothing but a linear (like a linear regression) equation. The predictors depend on the parameters (p,d,q) of the ARIMA model:

Number of AR (Auto-Regressive) terms (p): AR terms are just lags of dependent variable. For instance if p is 5, the predictors for x(t) will be x(t-1)….x(t-5).

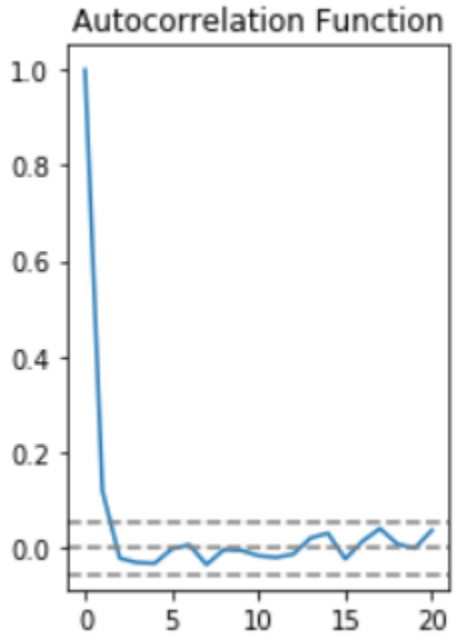
Number of MA (Moving Average) terms (q): MA terms are lagged forecast errors in prediction equation. For instance if q is 5, the predictors for x(t) will be e(t-1)….e(t-5) where e(i) is the difference between the moving average at ith instant and actual value.

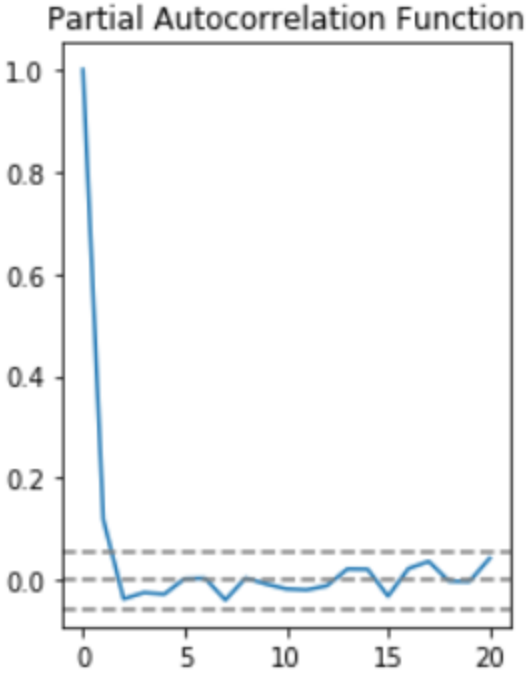
Number of Differences (d): These are the number of no seasonal differences, i.e. in this case we took the first order difference. So, either we can pass that variable and put d=0 or pass the original variable and put d=1. Both will generate same results.

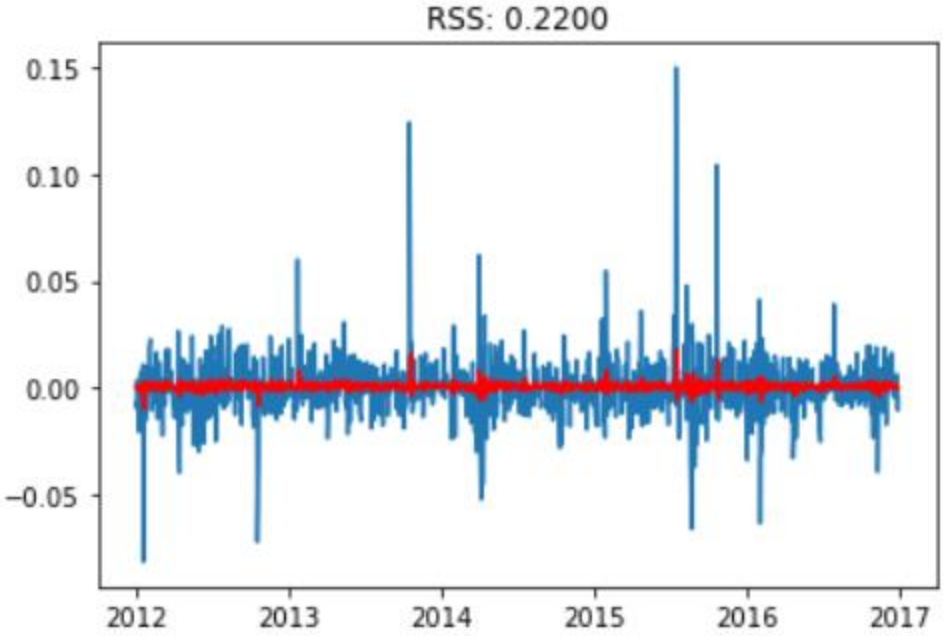
An importance concern here is how to determine the value of 'p' and 'q'. We use two plots to determine these numbers. Let's discuss them first.

Autocorrelation Function (ACF): It is a measure of the correlation between the TS with a lagged version of itself. For instance, at lag 5, ACF would compare series at time instant 't1'…'t2' with series at instant 't1-5'…'t2-5' (t1-5 and t2 being end points).

Partial Autocorrelation Function (PACF): This measures the correlation between the TS with a lagged version of itself but after eliminating the variations already explained by the intervening comparisons. Eg at lag 5, it will check the correlation but remove the effects already explained by lags 1 to 4.




Partial Autocorrelation Function
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20


RSS: 0.2200
0.15
0.10
0.05
0.00
−0.05
2012
2013
2014
2015
2016
2017

PREDICTION USING RECURRENT NEURAL NETWORK

Acknowledgment

We would like to show our gratitude to professor Nik Bear Brown for guiding us during this project.

Reference:

- Oscar Sharp & Benjamin, 2016, Sunspring
- Sepp (Josef) Hochreiter, 1991, Untersuchungen zu dynamischen neuronalen Netzen
- Yoshua Bengio, 1994, Learning Long-Term Dependencies with Gradient Descent is Difficult
- Razvan Pascanu, 2013, On the difficulty of training recurrent neural networks
- Sepp Hochreiter & Jurgen Schmidhuber, 1997, Long Short-Term Memory
- Christopher Olah, 2015, Understanding LSTM Networks
- Shi Yan, 2016, Understanding LSTM and its diagrams
- Andrej Karpathy, 2015, The Unreasonable Effectiveness of Recurrent Neural Networks
- Andrej Karpathy, 2015, Visualizing and Understanding Recurrent Networks
- Klaus Greff, 2015, LSTM: A Search Space Odyssey
- Xavier Glorot, 2011, Deep sparse rectifier neural network