

# Predicting Exchange Rate

Prasanti Das

Time Series Analysis

Aug 2018

# About Project

Goal :-

Develop a model which will help investors and business owners to decide about their investment opportunities across globe.

Benefits:-

- Maximize their profit
- Manage the cash flow and take business expansion decisions
- Helps to evaluate foreign borrowing decisions , investment opportunities
- Minimize the risk

# About the Data

The data set represents

- Exchange Rate Analysis from INR to USD
- Daily exchange rate information from the year 1973 till date
- The data source is <https://fred.stlouisfed.org>

# Data Cleaning

## Missing values

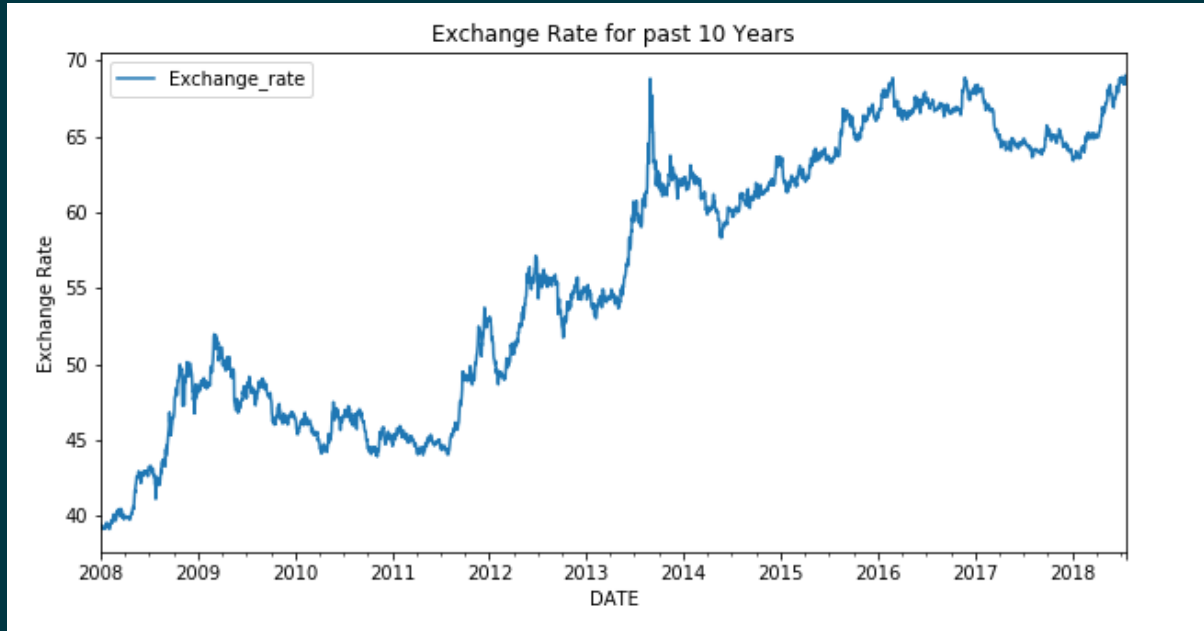
- 458 missing values in this dataset
- Used a forward-fill to propagate the previous value to next observation.

## Other:

- Given appropriate column name
- Changed the datatype of Date column from object to datetime
- Set the Date column as Index
- Changed the datatype of Exchange Rate column from object to numeric.

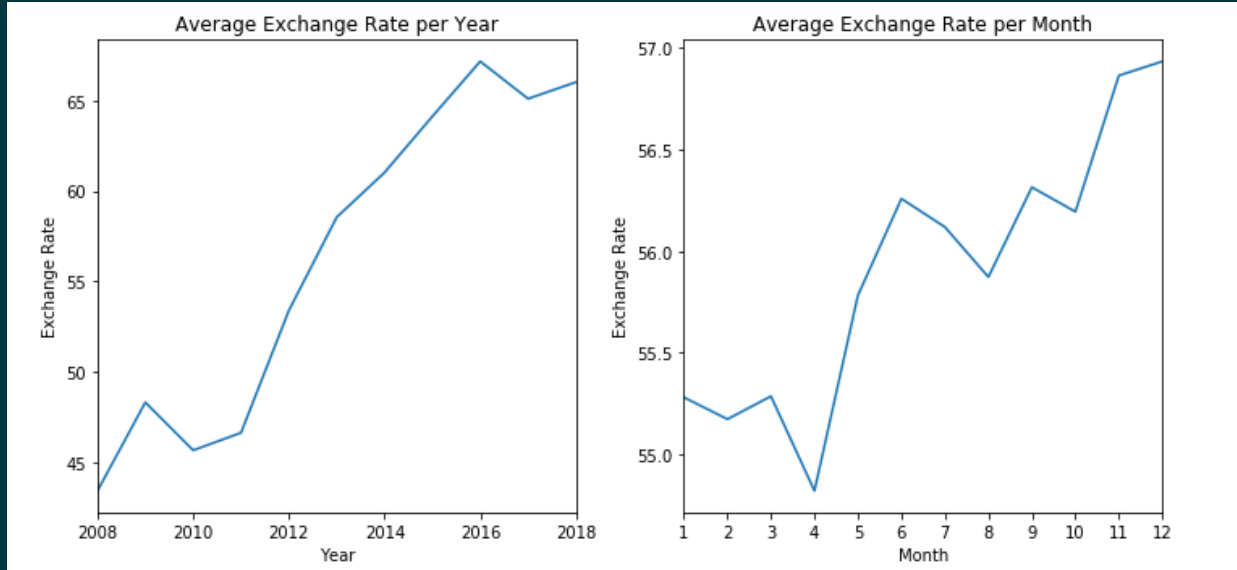
# Exploratory Data Analysis

# Exchange Rate for past 10 years



The exchange rates shows volatility, but illustrates an overall increasing trend.

# Average Exchange Rate per year and Month

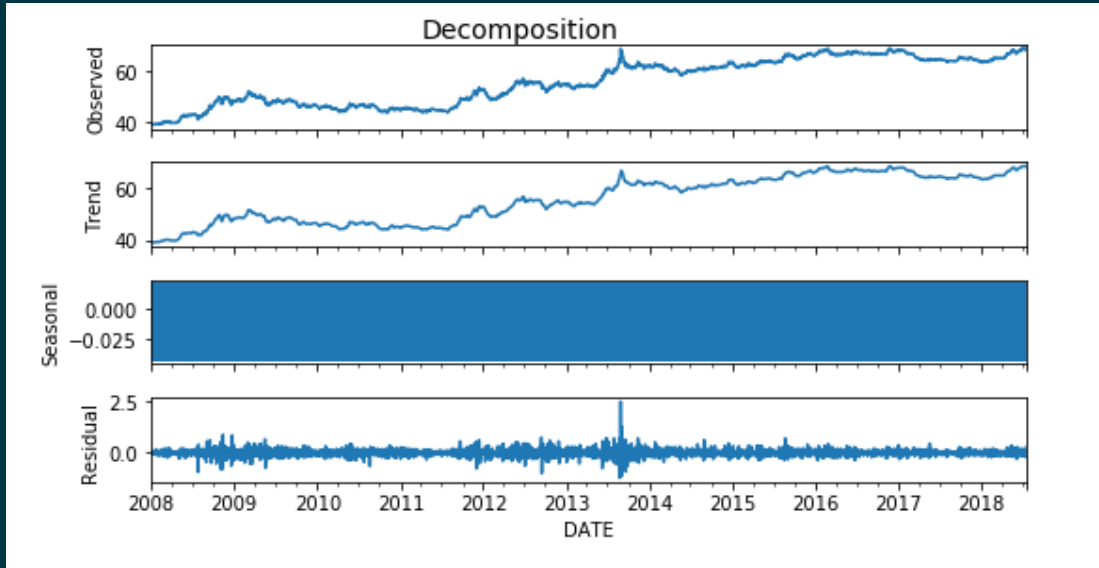


There is no seasonality, but we see an upward trend.

# Decomposition



# Decomposition



Decomposed the time series data to find patterns of Seasonality, Trend.

# Modelling

# Model Assumption and preprocessing:

Important Assumption :

Time Series Data is Stationary

Stationary:

Mean, variance and autocorrelation does not depend on time.

How to check stationarity ?

- Plotting Rolling Statistics
- Augmented Dickey-Fuller Test

How to achieve stationarity?

- First or second order difference
- Different transformation

# Augmented Dickey-Fuller (ADF) test

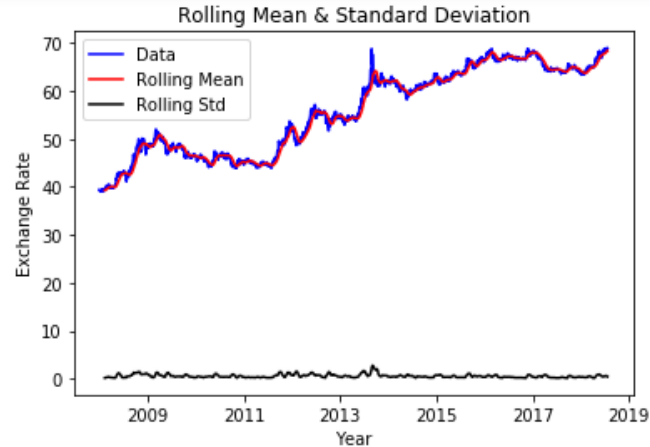
Used to test for a unit root in a time series sample.

Null hypothesis : there is a unit root,  
alternative hypothesis : there is no unit root.

Interpretation of ADF Test result:

- $p\text{-value} > 0.05$ :  
Accept the null hypothesis ( $H_0$ ),  
the data has a unit root and is non-stationary.
- $p\text{-value} < 0.05$ :  
Reject the null hypothesis ( $H_0$ ),  
the data does not have a unit root and is stationary.

# ADF test – Actual Data



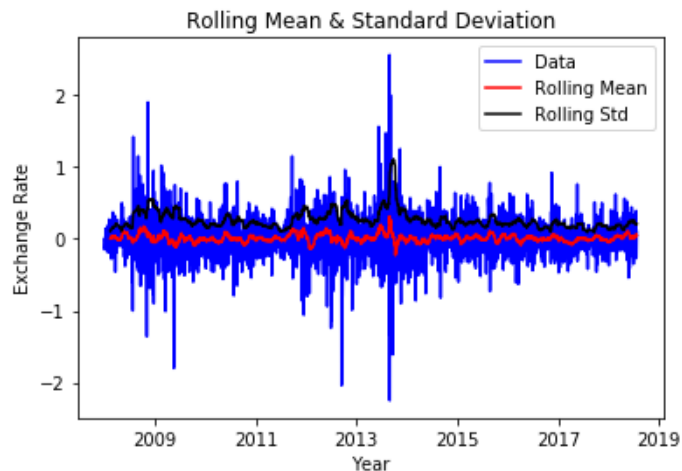
Results of Dickey-Fuller Test:

```
-----  
Test Statistic          -1.281828  
p-value                  0.637444  
#Lags Used               5.000000  
Number of Observations Used 2749.000000  
Critical Value (1%)      -3.432731  
Critical Value (5%)      -2.862592  
Critical Value (10%)     -2.567330  
dtype: float64
```

P value is more than 0.05

TS is not Stationary

# ADF test – Differenced Data



Results of Dickey-Fuller Test:

```
-----
Test Statistic      -22.092389
p-value             0.000000
#Lags Used          4.000000
Number of Observations Used  2749.000000
Critical Value (1%)   -3.432731
Critical Value (5%)   -2.862592
Critical Value (10%)  -2.567330
```

P value is less than 0.05  
TS is Stationary

# ACF and PACF

## ACF

Autocorrelation Function:

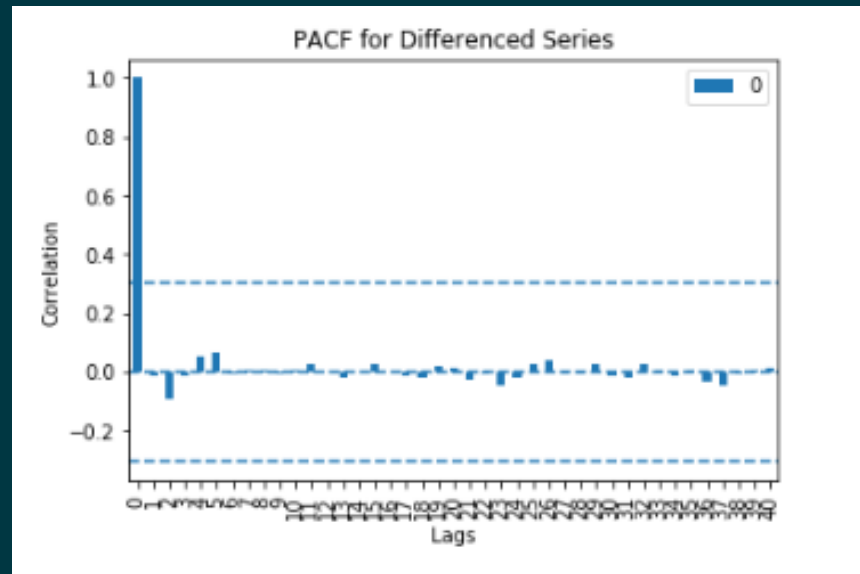
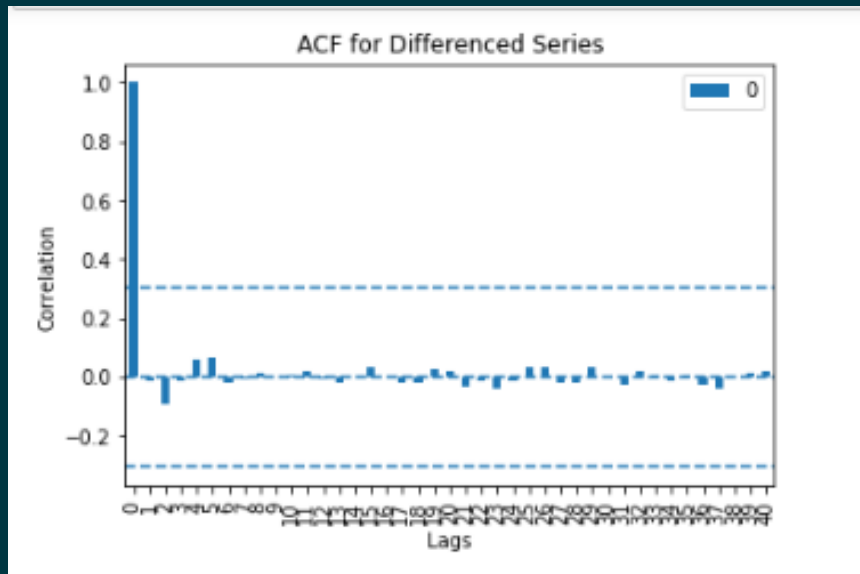
Measure of the correlation between the Time series with a lagged version of itself.

## PACF

Partial Autocorrelation Function:

Measures the correlation between the Time series with a lagged version of itself, but after eliminating the variations already explained by the shorter lags.

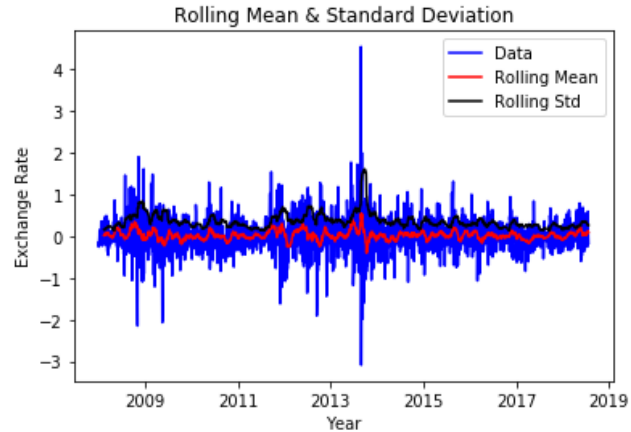
# ACF and PACF plot



No significant lags.



# ADF test – Differenced Data

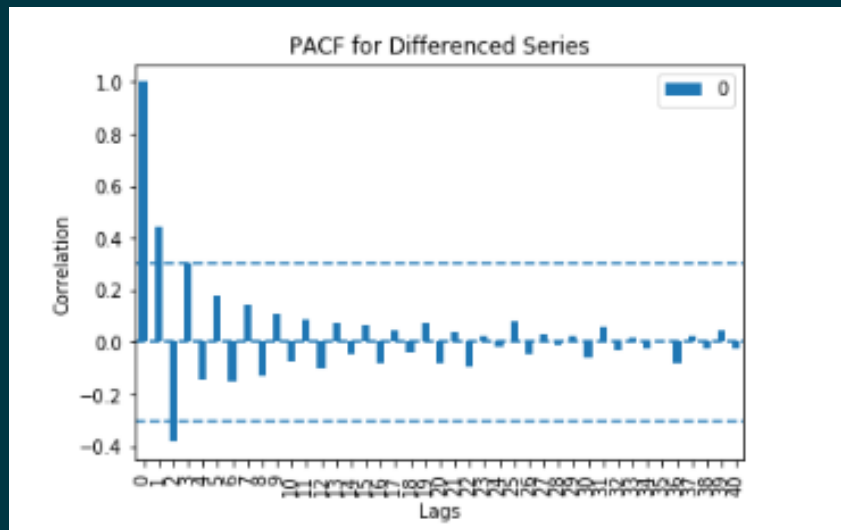
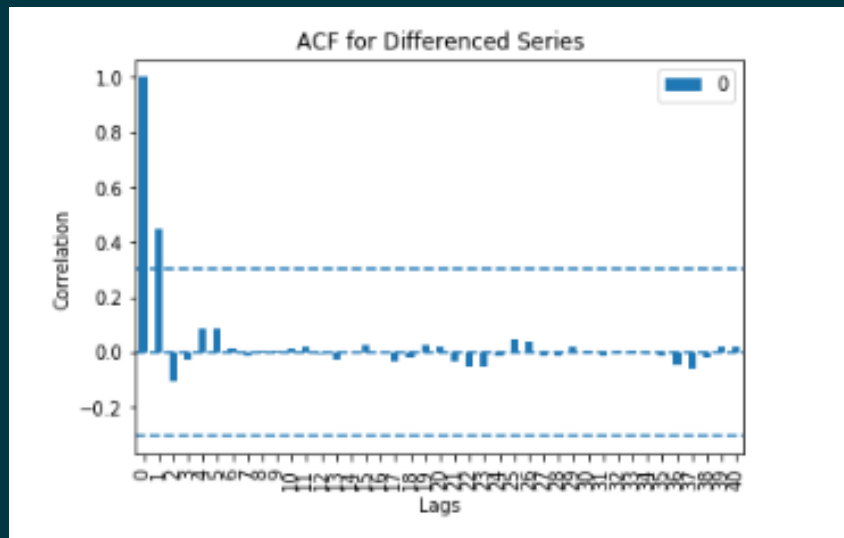


Results of Dickey-Fuller Test:

```
-----  
Test Statistic      -1.031000e+01  
p-value             3.203941e-18  
#Lags Used          2.500000e+01  
Number of Observations Used  2.727000e+03  
Critical Value (1%)  -3.432750e+00  
Critical Value (5%)  -2.862600e+00  
Critical Value (10%) -2.567335e+00  
dtype: float64
```

Stationary  
P value is less than 0.05

# ACF and PACF plot



ACF plot suggests MA of order 1  
PACF plot suggests AR of order 2 or 3

# ARIMA Modelling

Enables to use linear regression models on non-stationary data.

AR (Autoregression):

Uses the relationship between an observation and its lagged observations.

I (Integrated):

Order of differencing of observations to make the time series stationary.

MA (MovingAverage):

Uses the dependency between an observation and past error.

# ARIMA Modelling

Denoted with the notation  $ARIMA(p, d, q)$

$p$  : Number of AR terms

$d$  : the degree of differencing.

$q$  : Number of MA terms

Tried Grid search on

$p - 0, 1, 2, 3$

$d - 0, 1, 2$

$q - 0, 1, 2, 3$

# Train Test Split and Evaluation

Split:

The last 90 observations as the test data and remaining as training data

AIC:

Akaike information criteria : provides a way of model selection.

MSE:

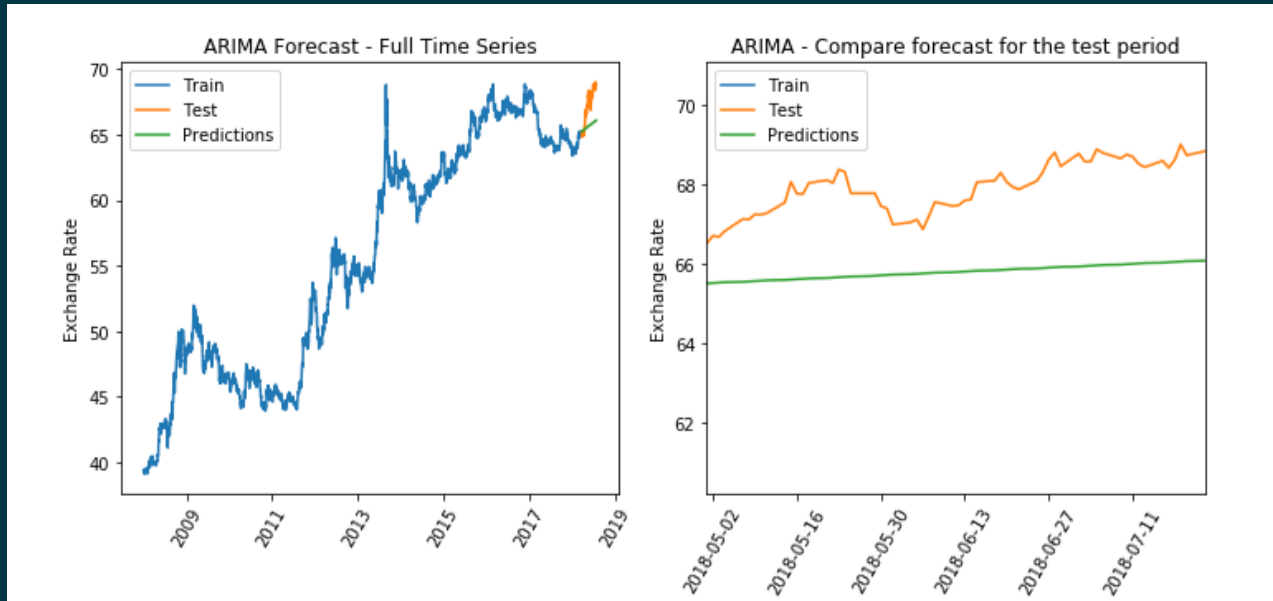
Mean squared error

Validation:

walk-forward model validation

# ARIMA Best Model - Prediction

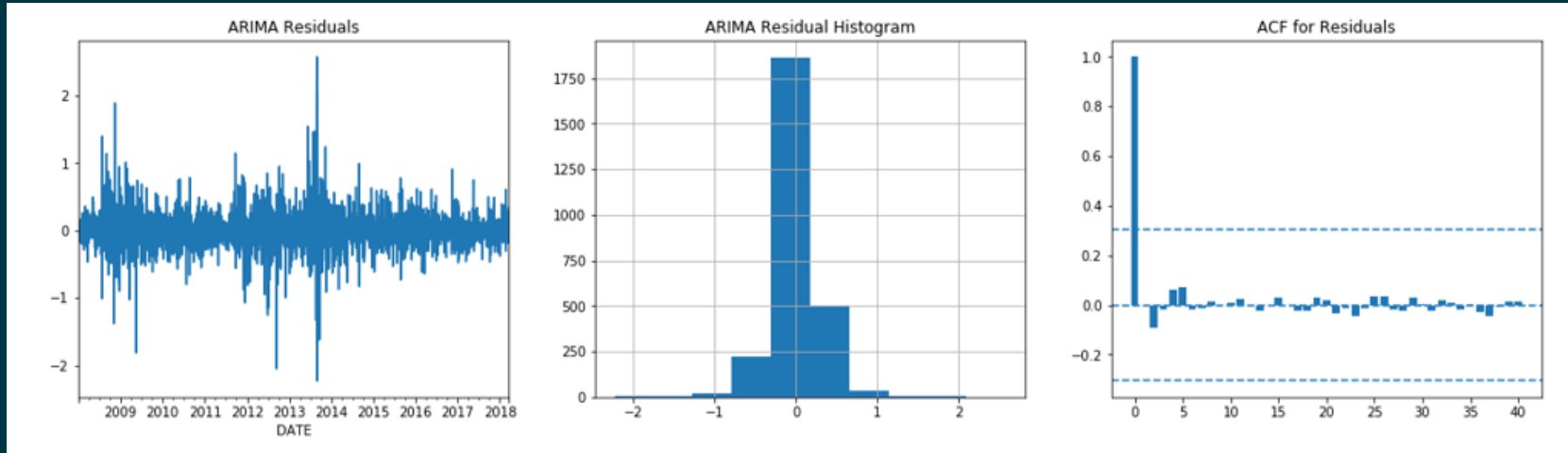
Lowest MSE



Best Model  
Chosen by  
ARIMA- 1,1,0

AIC: 940.7  
MSE: 3.33

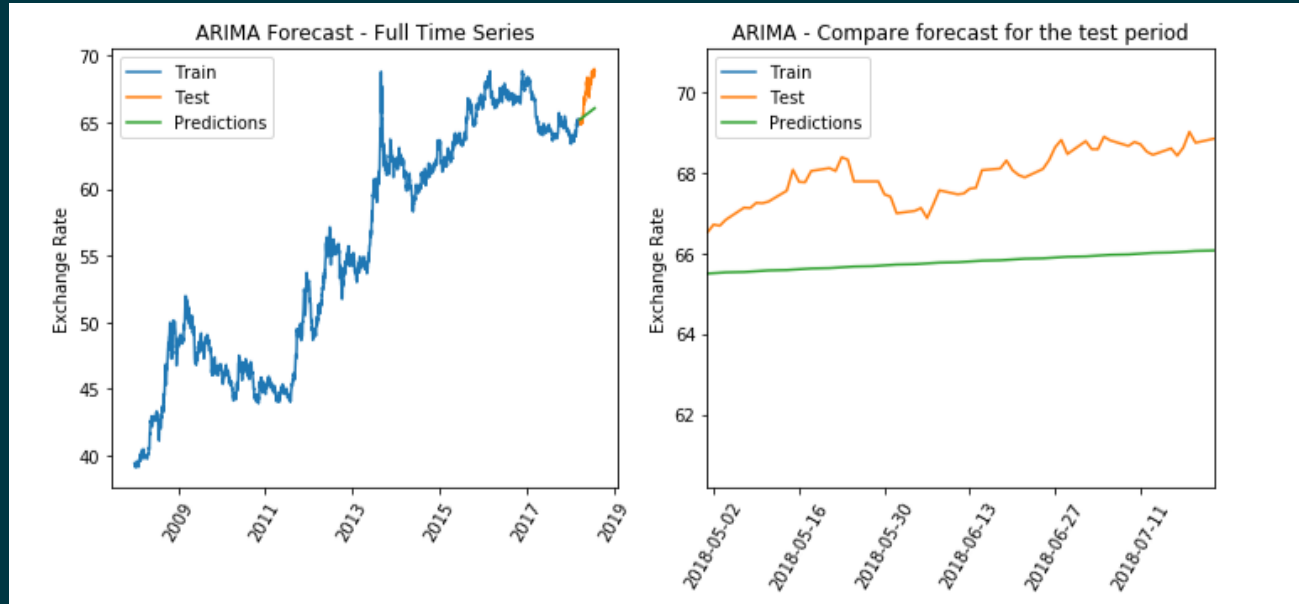
# ARIMA Best Model - Residual



The residuals appear to be normally distributed and the ACF plot for residuals also don't show any significant lag values.

# ARIMA Best Model - Prediction

Lowest AIC

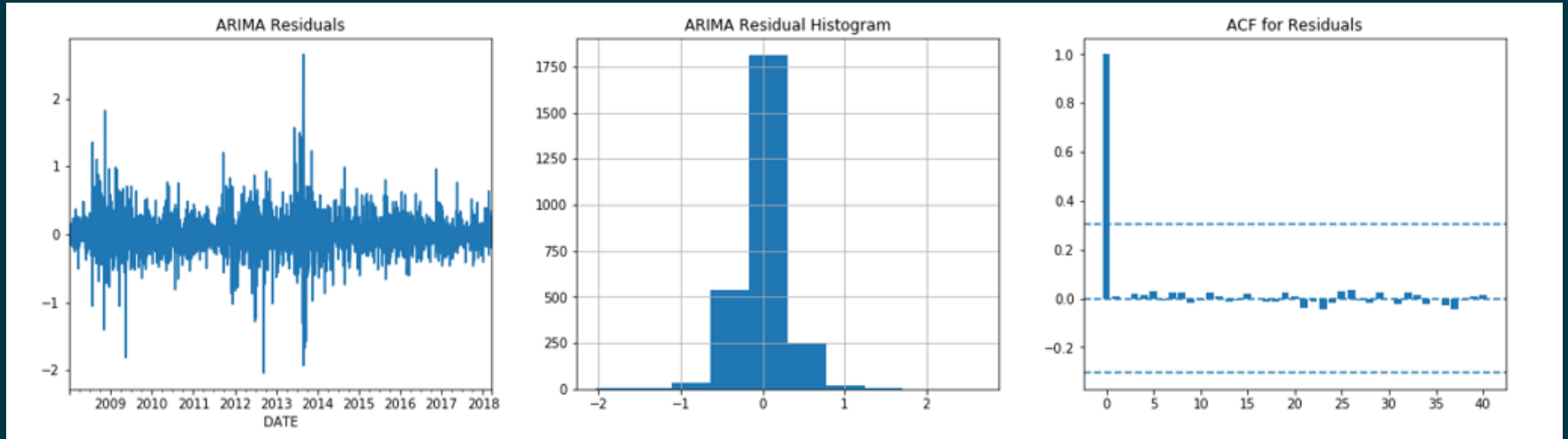


Best Model  
Chosen by  
ARIMA- 2,1,2

AIC: 907.2  
MSE: 3.37



# ARIMA Best Model - Residual



The residuals appear to be normally distributed and the ACF plot for residuals also don't show any significant lag values.

# AUTO ARIMA Modelling

- Wraps statsmodels' well-tested ARIMA and SARIMAX estimators in a single, easy-to-use package
- Uses pyramid-arima library for Python
- allows to quickly perform grid search

Installation Requirement :

pyramid-arima should be installed

Other Parameters:

m : The period for seasonal differencing.

seasonal : Whether to fit a seasonal ARIMA. Default is True.

stepwise : less likely to over-fit the model.

# Auto ARIMA Modelling - Result

## Grid Search Parameters

$P = 0, 1, 2$

$D = 0, 1$

$Q = 0, 1, 2$

$m = 1, 7, 30$

$p = 0, 1, 2, 3$

$d = 0, 1, 2$

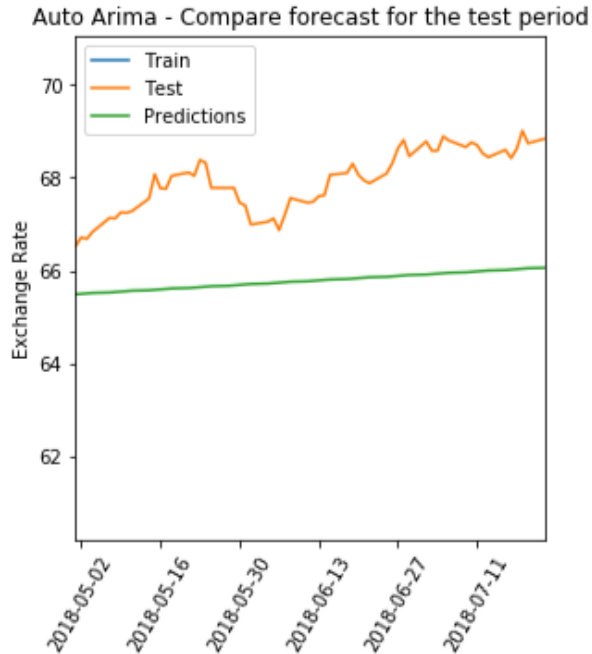
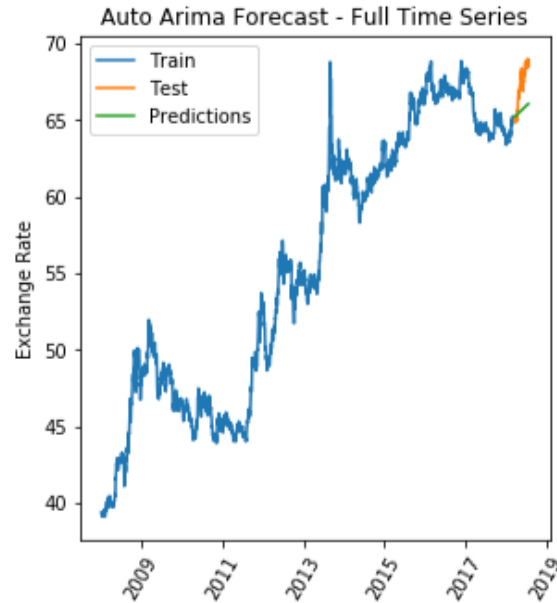
$q = 0, 1, 2, 3$

Best Configuration: (2, 1, 0)

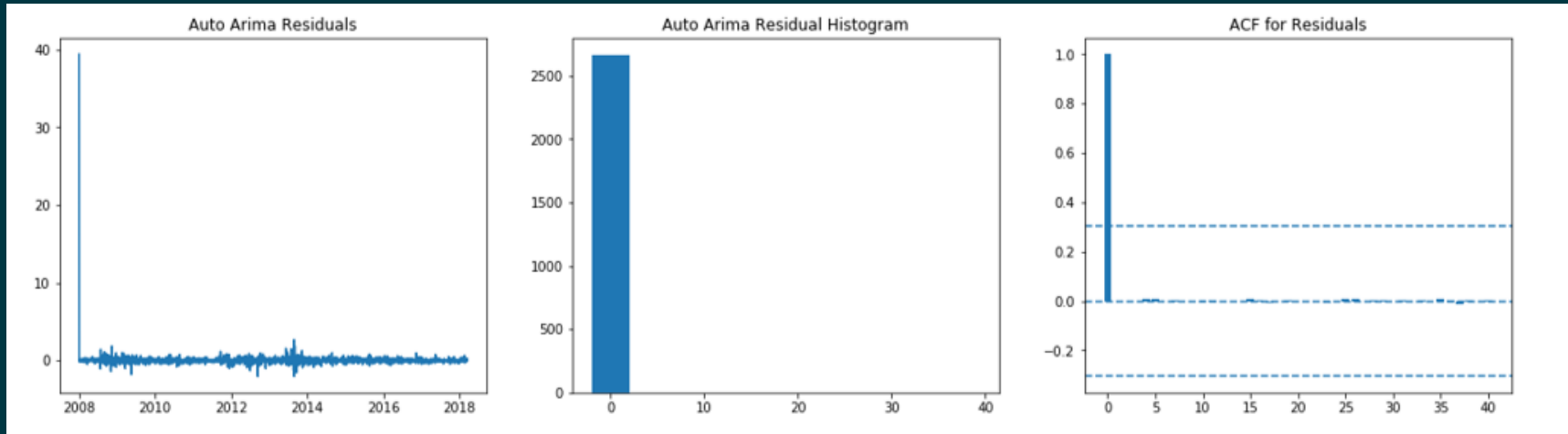
AIC : 976.28

MSE : 3.39

# Auto ARIMA Best Model - Prediction



# Auto ARIMA Best Model - residual



The residuals appear to be normally distributed and the ACF plot for residuals also don't show any significant lag values.

# Prophet Model

- Developed by Facebook for forecasting time series data.
- Based on an additive model
- Non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.
- Works best with time series that have strong seasonal effects and several seasons of historical data.
- Robust to missing data and shifts in the trend, and typically handles outliers well.

# Prophet Model

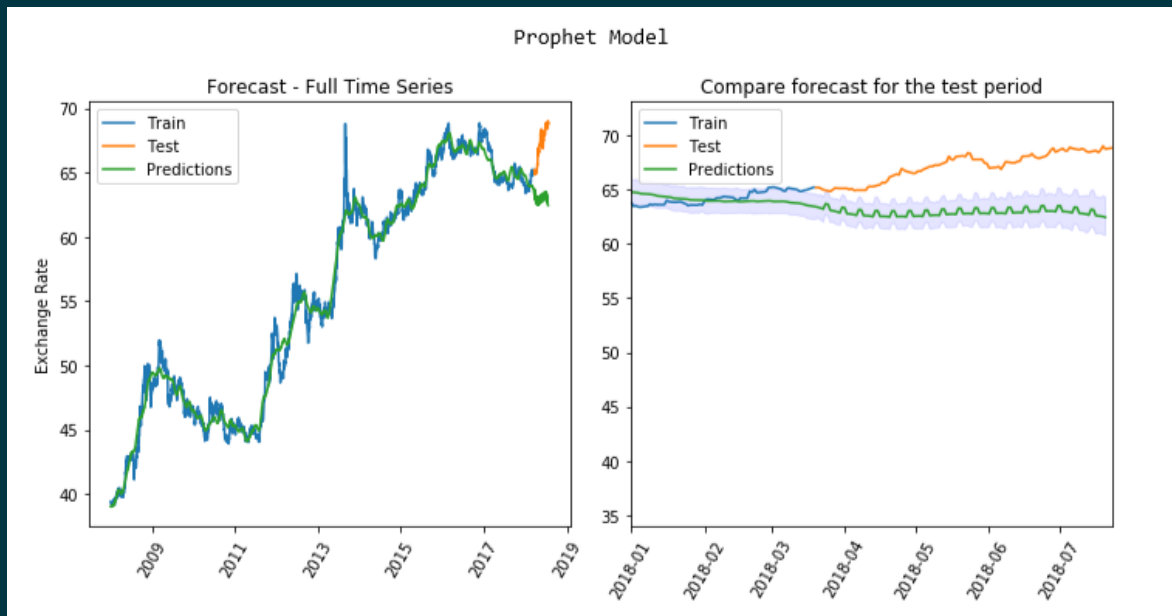
## Installation Requirements:

Need pystan, cython and fbprophet modules installed .

## Input Requirements:

- Requires specific input format.
- Columns names must be lowercase,
- date column should be named as 'ds', data as 'y'

# Prophet Model - prediction

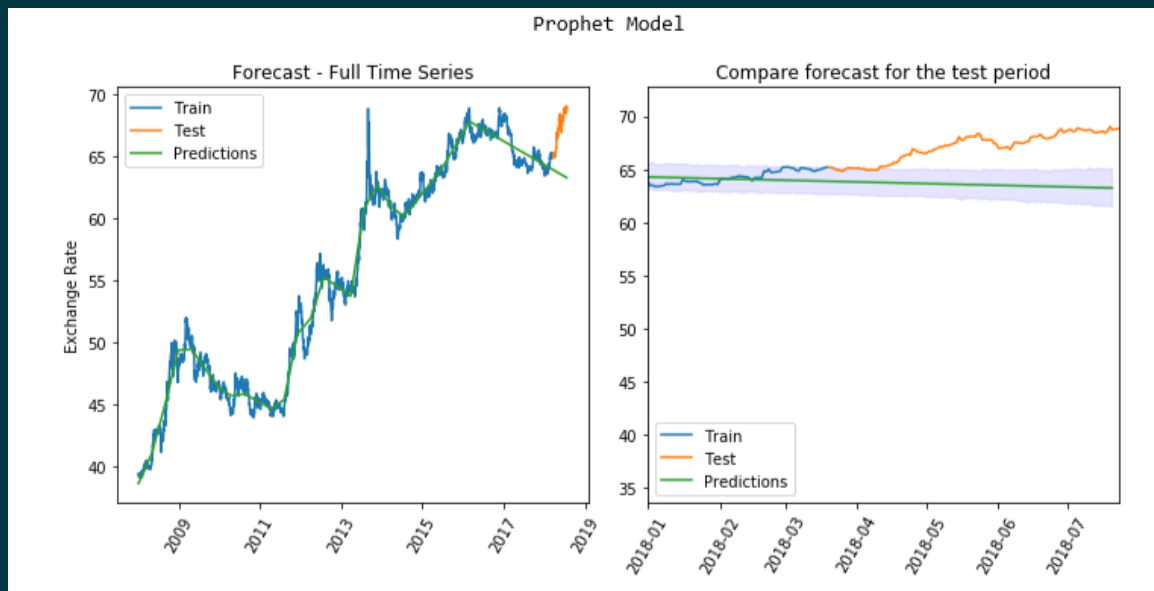


Prophet Model  
(Default parameters)

MSE : 1.43



# Prophet Model - prediction

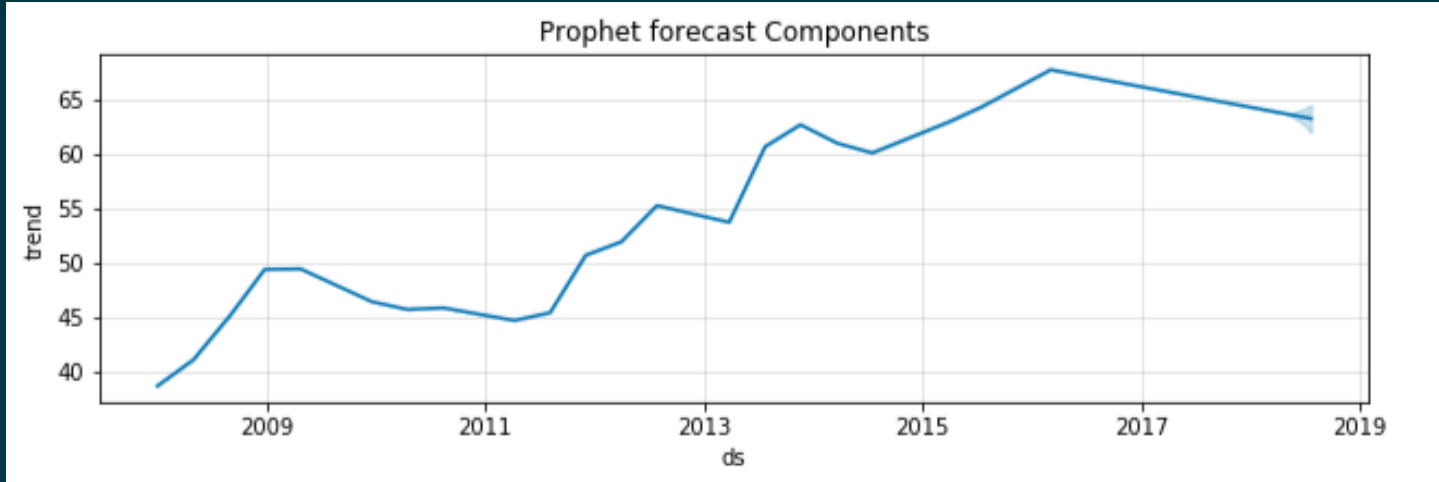


Prophet Model  
(parameter tuning)

Daily, weekly and yearly  
seasonality=False

MSE : 1.39

# Prophet Model – forecast components



Prophet forecast a decreasing Trend

# LSTM network

- Stands for Long Short-Term Memory Network
- A recurrent neural network
- Trained using Backpropagation Through Time
- Overcomes the vanishing gradient problem

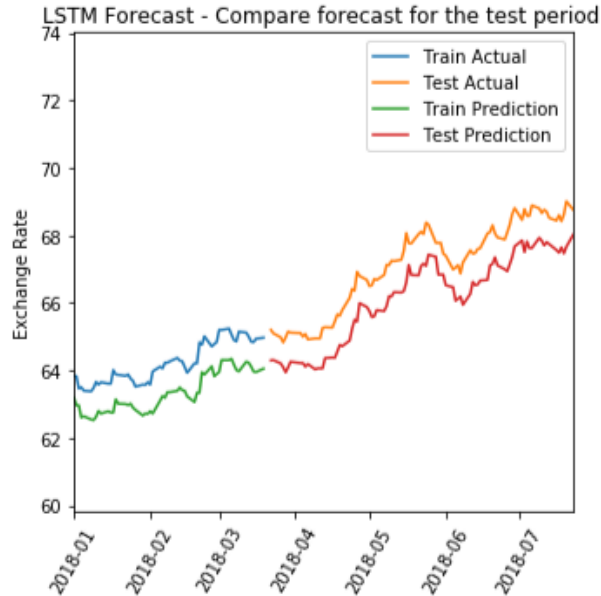
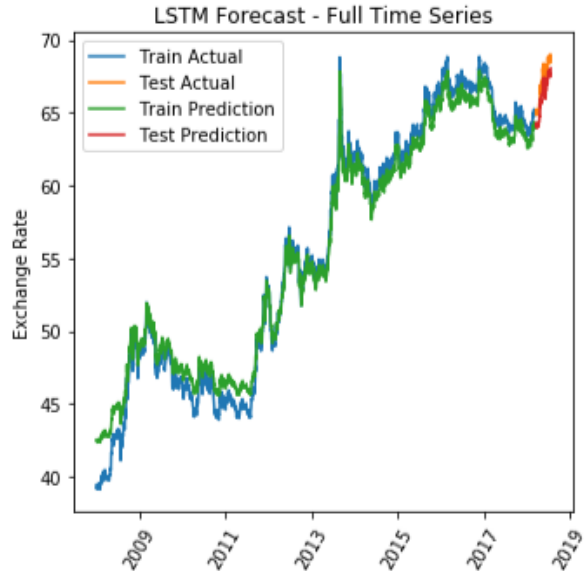
# Preprocessing - LSTM

- Sensitive to the scale of the input data
- Took last 90 observations as test data and rest as training data
- Needs input in the form of: [samples, time steps, features]
- Wrote a simple function to convert our single column of data into a multi-column dataset and consider the last column as dependent variable.

# LSTM Model

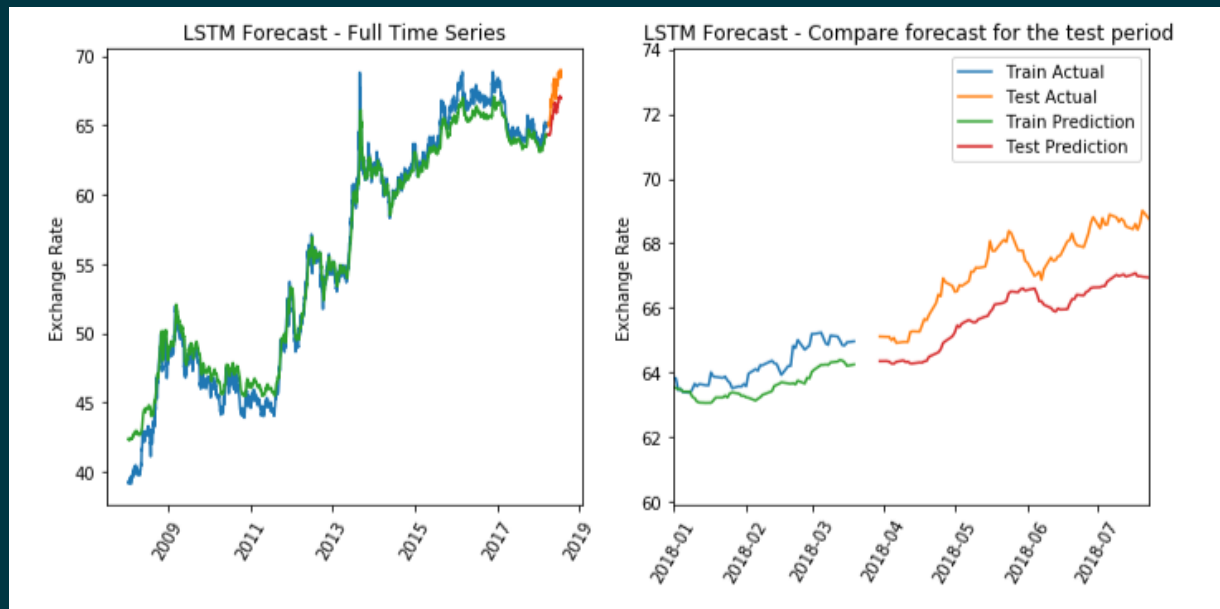
- Fitted a Model with 1 input layer, 1 hidden layer with 4 LSTM blocks, and 1 output layer
- default sigmoid activation function is used for the LSTM blocks.
- Dropout – for addressing overfitting problem
- The network is trained for 100 epochs and a batch size of 64.

# LSTM network – Prediction – 1 day Look Back



- Look back of 1 day data-  
MSE = 0.98

# LSTM network – Prediction – 7 days Look Back



- Look back of 7 days data-  
MSE = 2.3

# Model Comparison – Using MSE

```
model_comp_df.sort_values(by= 'Mean_Squared_error')
```

	Model	AIC_Value	Mean_Squared_error
3	LSTM_1_look_back	NaN	0.980652
2	PROPHET	NaN	1.389718
4	LSTM_7_look_back	NaN	2.300942
0	ARIMA	940.706970	3.335052
1	AUTO_ARIMA	976.279331	3.399531

The LSTM model with 1 day look back, provided the lowest mean squared error



# Conclusion

We looked at a large range of ARIMA configurations as well as a more powerful forecasting tool developed by the Facebook. We also looked at the LSTM Neural Network model. All of the models did fairly well.

## Model improvements:

- For prophet – adding Holiday calendar
- For LSTM models - try parameter tuning and adding More layers

THANK YOU

# Questions