STAT 153 Project

Group: Parul Madaan (3035332938)

May 6, 2020

Blog Post: Link for Blog Post

Executive Summary:

In this report, I have analyzed the stock price data for Mediocre Social Network Apps Incorporated (MSN Apps) from the beginning of 2015 through the end of September 2019. The report contains the discussion and analysis of two time series models fitted to model the stock price data: paramteric Linear fir model with AR(1) and SARIMA(d=1,D=1,Q=1,S=5) model. I have finally used the latter model to forecast the stock price of MSN Apps for the first ten trading days of October 2019. Based on the forecast, the future doesn't seem as bright as Mediocre Social Network Apps is currently hoping for. The start of the fourth quarter doesn't seem to be an end of the struggling era for the company.

1 Exploratory Data Analysis

An initial look at the time series plot of the MSN Apps stock data (Figure 1) shows an overall downward trend in the stock prices along with time.

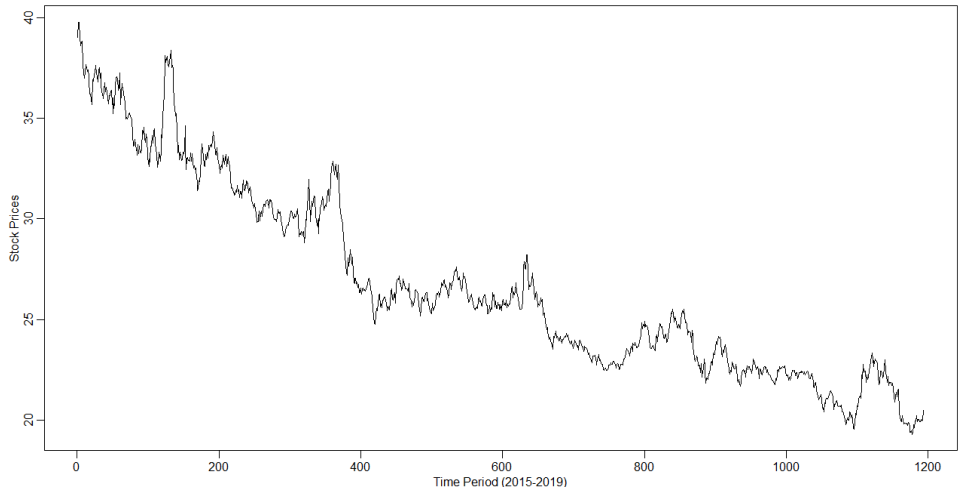

Figure 1: Stock Price for MSN Apps

The variable fluctuation in the prices (heteroscedasticity) is handled by perfoming log tranformation on the data to stabilize the variance before modelling it. Relatively homoscedastic, the series is not stationary as the mean of the trend is still changing. This might be indicative of some seasonal trends or auto regressing pattern in the data.

Then, I differenced the data to remove the linear trend and analyzed the differenced time series. As expected, the differenced series ∇X_t appears reasonably stationary.

Thus, I decided to fit a paramteric linear trend to the data and analyzed the residuals. As can be seen in the ACF/PACF (Figure 2) graph , there is a significant value at lag 1 in the PACF graph, and

also the significant values in the ACF are tapering off which indicates towards the presence of AR(1) process.

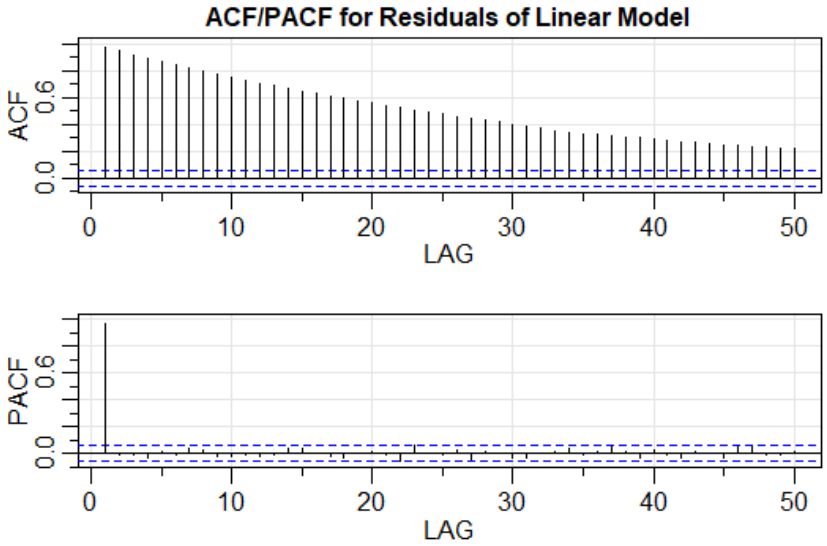

Figure 2: ACF/PACF: Residual after fitting Linear Trend

2 Models Considered

Based on the EDA, I tried multiple models to fit the data. Here I am presenting two of them that worked best in my opinion. Firstly, based on the data exploration, I fitted a autoregressive AR(1) model on the residuals of the log data:

2.1 Model A

As mentioned above, Model A represents the paramteric linear trend model with AR(1) residuals and it fits the data really well.

The output of the AR(1) model is shown below (Figure 3) which clearly shows that the residuals are stable and this appears to be an AR(1) process.

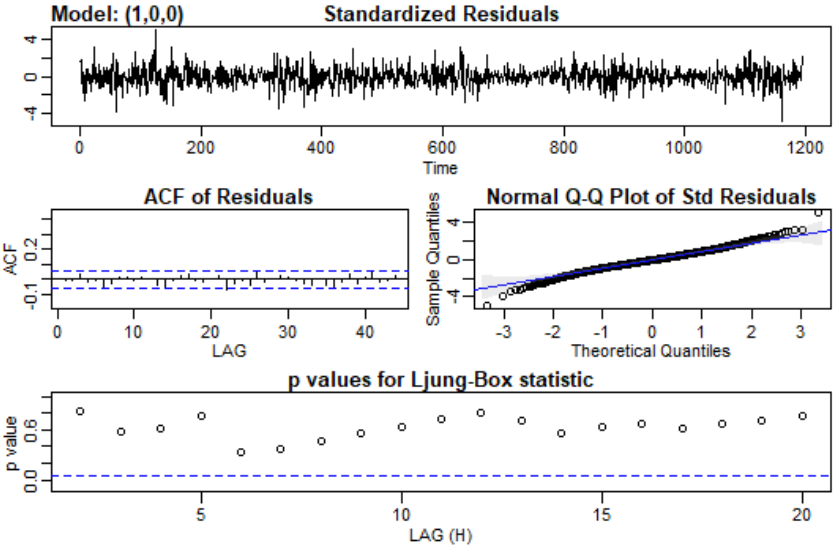

Figure 3: Model A Diagnostics

As all the ACF values are within the blue dashed line, there doesn't seem to be any correlation left in the residuals. Referring to the Ljung-Box statistics plot at the bottom, the p-values are quite high which represents a good model fit.

2.2 Model B

Model B represents the SARIMA model with the following parameters: d=1,D=1,Q=1,S=5

A closer look at the original time series data indicates the presence of a seasonal trend on a weekly basis (5-6 days for the given data). The ACF and PACF of differenced data with (d=1 and d=5) subsequently validates this finding (Figure 4).

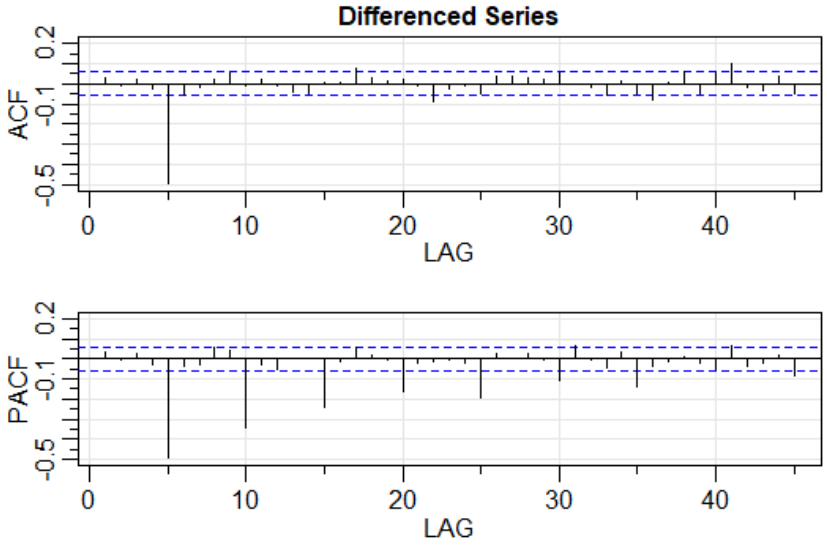

Figure 4: ACF/PACF of Differenced Log Series (d=1 and d=5)

A large spike in the ACF plot at 5 and a decaying trend in the PACF with a interval of 5 directs towards using a seasonal MA(1) model with period =5, thus the choice of using SARIMA(0,1,0)(0,1,1)[5]. The output of the model fit is represented below:

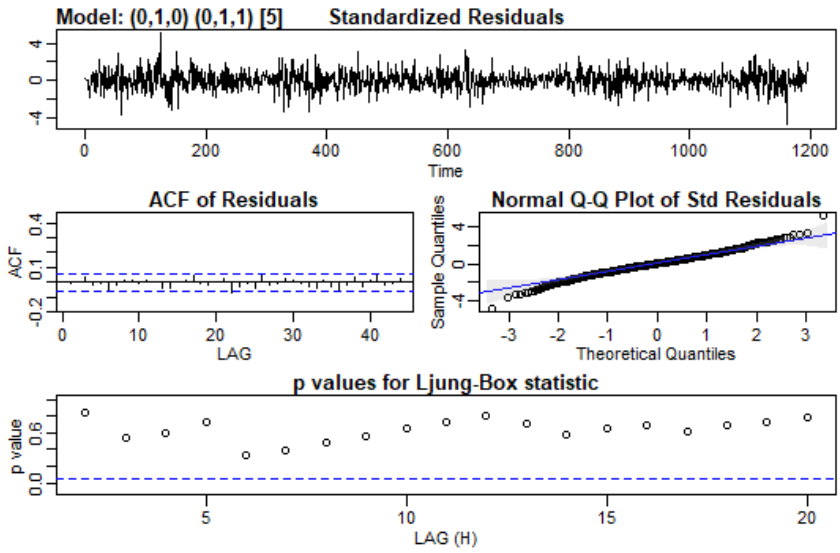

Figure 5: Model B Diagnostics

The model diagnostic tells that the fitted SARIMA model is a good fit to the data, since the residuals are quite stable and the p-value for the Ljung-Box statistic is also high.

3 Model Comparison and Selection

Since the final task at hand is to forecast, to compare the two models, I used cross-validated (k = 19) Mean Square Error for prediction as the evaluation criterion. Starting with minimum 1000 data points, I fitted multiple models with a rolling window of 20 points and computed the mean squared error in the prediction of upcoming 10 days for the two models. Mentioned below is the comparison of the average MSE for logged series and average MSE of actual stock prices predicted by the two models across all model fits:

Model Name	Description	Avg Log Price MSE	Avg Price MSE
Model A	Linear Trend and AR(1)	0.000432	0.1978
Model B	SARIMA(0,1,0)(0,1,1)[5]	0.000432	0.1973

Table 1: Cross-validated MSE for Model A and B

Based on the results above, there is a marginal difference in the performance of the two models in terms of cross- validated Mean Squared Error. I decided to go ahead with the **Model B** for further analysis and forecast.

4 Results

The final SARIMA model (Model B) is of the following form:

$$\nabla_5 \nabla log(X_t) = Z_t + \theta Z_{t-5} \tag{1}$$

where Z_t represents white noise. There is only one paramters, θ which is equal to -0.988 with a standard error of 0.0065.

4.1 Prediction

The following figure (Figure 6) represents the forecast for the first 10 days of the October 2019 along with the confidence interval using Model B:

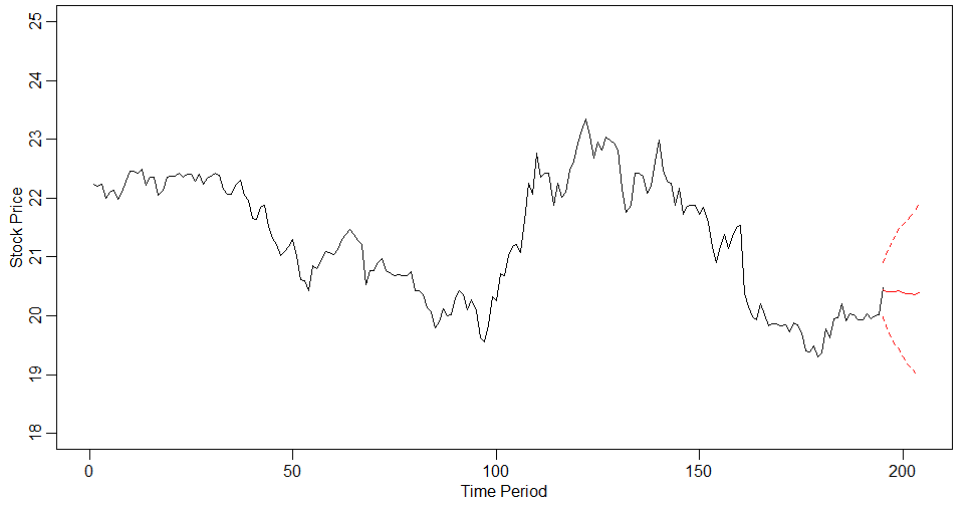

Figure 6: Forecast for the upcoming 10 Days

Model's forecast doesn't give much hope about the start of the fourth quarter. Model predicts the decreasing trend in the stock prices of Mediocre Social Network Apps Incorporated would be continued and thus the executives or investors shouldn't be much hopeful about the future of the company.

5 Appendix: R Code

```
library(forecast)
library(astsa)

#Read the data
stocks <- read.csv("D:/Berkeley/Stat153_TimeSeries/Project/stocks.csv")

#Plotting the given data
plot.ts(stocks$Price,ylab = "Stock Prices", xlab= "Time Period (2015-2019)")

#VST - log transformation
log_stocks = log(stocks$Price)
plot.ts(log_stocks, ylab = "Log (Stock Prices)", xlab= "Time Period (2015-2019)")

#Differencing to see if removing linear trend makes the data stationary
D1X = diff(log_stocks)
plot.ts(D1X)

# ACF/PACF of differenced series
a = acf2(D1X,main="Differenced Series")

#Models Considered

#Model A: Linear Trend with AR(1)
time = 1:length(log_stocks)
modelA = lm(log_stocks~time)
residA = modelA$residuals

acf2(residA,max.lag = 50,main = "ACF/PACF for Residuals of Linear Model")

ar1 = arima(residA, order = c(1,0,0), include.mean = FALSE)

forecasts_modelA= predict(ar1,n.ahead = 10)

new_time = 1195:1204
predictions = modelA$coefficients[1] + modelA$coefficients[2]*new_time +
forecasts_modelA$pred
var = forecasts_modelA$se

exp(predictions)
exp(predictions+ 2*var)
exp(predictions- 2*var)

plot(stocks$Price[1000:1194],type='l',xlab='Time Period',ylab='Stock Price',
xlim=c(0,204),ylim = c(18,25))
lines(195:204,exp(predictions),col=2)
lines(195:204,exp(predictions + 2*var),lty=2,col=2)
lines(195:204,exp(predictions - 2*var),lty=2,col=2)

#Model B: SARIMA(d=1,D=1,Q=1,S=5)

# Forecasts and Confidence Interval - Model B

modelB = sarima(log_stocks,p=0,d=1,q=0,P=0, D=1, Q=1,S=5)

predictions = sarima.for(log_stocks, n.ahead=10, p=0, d=1, q=0, P=0, D=1, Q=1,
                         S=5)$pred
var = sarima.for(log_stocks, n.ahead=10, p=0, d=1, q=0, P=0, D=1, Q=1,
                 S=5)$se

exp(predictions)
```

```
exp(predictions+ 2*var)
exp(predictions- 2*var)

write.csv(exp(predictions),"stocks_3035332938_NA_NA_NA_NA.csv",
row.names = FALSE,quote = FALSE)

plot(stocks$Price[1000:1194],type='l',xlab='Time Period',ylab='Stock Price',
xlim=c(0,204),ylim = c(18,25))
lines(195:204,exp(predictions),col=2)
lines(195:204,exp(predictions + 2*var),lty=2,col=2)
lines(195:204,exp(predictions - 2*var),lty=2,col=2)

#Cross- Validation - MSE
mse_log_modelA = list()
mse_log_modelB= list()
mse_modelA=list()
mse_modelB=list()

for (i in seq(1000,1184,10)) {

  train_set <- window(log_stocks,end=i)
  test_set <- window(log_stocks,start=i+1,end=i+10)

  #AR(1)
  time = 1:length(train_set)
  req_model = lm(train_set~time)
  req_resid = req_model$residuals
  forecast_residual <- sarima.for(req_resid, n.ahead=10, p=1, d=0, q=0)$pred

  new_time = (length(train_set)+1):(length(train_set)+10)
  forecast1 = req_model$coefficients[1] + req_model$coefficients[2]*new_time +
    forecast_residual

  #SARIMA(0,1,0)(0,1,1)[5]
  forecast2 <- sarima.for(train_set, n.ahead=10, p=0, d=1, q=0, P=0, D=1, Q=1,
                          S=5)$pred

  #Mean Squared Error for Logged Prediction
  mse_log_modelA =  c(mse_log_modelA,mean((forecast1 - test_set)^2))
  mse_log_modelB =  c(mse_log_modelB,mean((forecast2 - test_set)^2))

  #Mean Squared Error for Actual Data Point Prediction
  mse_modelA =  c(mse_modelA,mean((exp(forecast1) - exp(test_set))^2))
  mse_modelB =  c(mse_modelB,mean((exp(forecast2) - exp(test_set))^2))

}

#Log MSE
mean(unlist(mse_log_modelA))
mean(unlist(mse_log_modelB))

#MSE
mean(unlist(mse_modelA))
mean(unlist(mse_modelB))
```