Mediocre Social Network Apps Inc. Stock: Analysis and Forecast

Group: Dana Shan, Yifan Xia, Yujing Sun, Ziyan Xia

May 6, 2020

Blog Post: [https://rpubs.com/YifanXia0623/610562]

Executive Summary:

The first part of this report provides an analysis for the stock of Mediocre Social Network Apps Inc. from the beginning of 2015 through the end of September of 2019. The analysis is to be used to forecast the stock price for the first ten trading days of October 2019 which will be explained in the second part of this report. After comparing several models, we found that an ARIMA $(2, 1, 4)$ x $(0, 0, 1)_{22}$ model best captured the volatility in the stock price and gave the most accurate predictions.

1 Exploratory Data Analysis

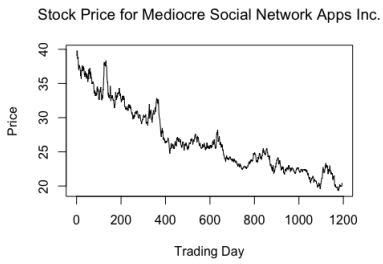

Figure 1: Mediocre Social Network Apps Inc.'s stock price from Jan 2nd 2015 to Sept 30th 2019.

From Figure 1, we see that the data appears to have a relatively strong negative linear trend. Since our timeplot displays price against trading days, we notice a lot of noise to our data which is normal due to the nature of the stock market having price fluctuations or the so-called stock market noise caused by the volatility in the stock market itself. Thus, the noise should not distort our overall trend.

Second, we notice that the magnitude of our data does not distinctly vary over time, so it is reasonable for our data to be considered as somewhat homoscedastic. However, we believe that further could be done to strengthen the homoscedasticity of our model. This will be further discussed in Section 1.1.

Notably, there is a steep increase in stock price roughly every 150-250 trading days. Since there is a large variation in the peaking periods over the years and that our 10 predicted data points are right after the peak phase in 2019, it could be reasonable to disregard these peaking periods when making short-term forecasting. Since the idea of seasonality, having the peaks and valleys in the graph occur at regular periods, isn't the most obvious in our case, we will model our data with the seasonality of our choice and without seasonality in general to understand the effect on our prediction.

1.1 Pursuing Stationarity

Since most statistical forecasting methods are based on the assumption that the time series can be rendered approximately stationary, we need to ensure that we perform the proper mathematical transformations so that our process obtains homoscedasticity with a constant mean.

In an effort to stabilize the mean, we take the first order differencing of the data to remove the changes in level of the time series and eliminate trend. This essentially means that we subtract the value

of each observation with the one directly before it ($\nabla X_t = X_t - X_{t-1}$). As mentioned in the previous section, it can be not as clear to whether our data is truly homoscedastic. Therefore, we would also like to test whether applying variance stabilizing transform has an effect on the residuals of our data. Thus, we are showing a side to side comparison of the residuals plots with first order differencing. Figure 2 shows the first order differencing residuals plot of the original time series data, and Figure 3 shows the plot with log() applied to the time series.

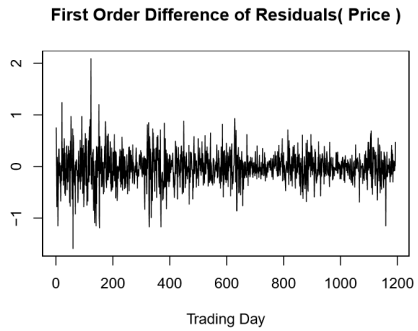

Figure 2: First Order Difference of Residuals. We use the original stock price.

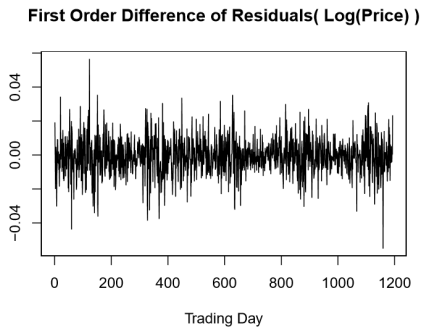

Figure 3: First Order Difference of Residuals. We use the logged stock price.

With the comparison, we see that the first order differencing residuals plot with the logged stock price yields a process that clearly looks much more stationary. Thus, we will proceed with our analysis with the variance stabilizing transform applied to our time series.

2 Models Considered

2.1 ARIMA(1,1,0)

For our logged time series data, we apply the auto.arima() function in R . This returns the best ARIMA model according to either AIC, AICc, or BIC value. The suggested model we get is an ARIMA(1,1,0). This is a differenced first-order autoregressive model. It is represented by the equation $X_t = X_{t-1} + \phi(X_{t-1} - X_{t-2}) + Z_t$, where ϕ is the autoregressive parameter to be estimated, and white noise is defined to be a zero mean, uncorrelated process with constant variance.

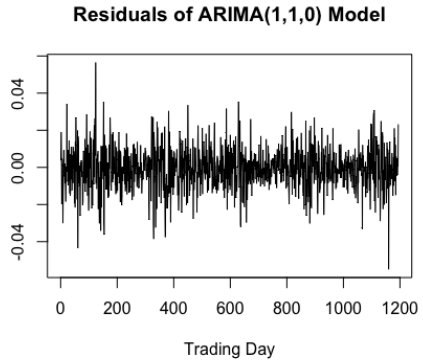

Figure 4: Residuals Plot

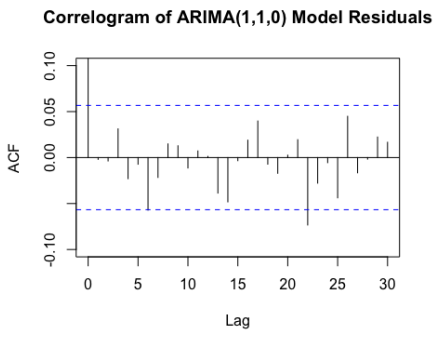

Figure 5: ACF of residuals with 95% confidence bands

From the residual plot of the model shown in Figure 4, we see that the model looks pretty stationary with roughly zero mean and constant variance, From the autocorrelation function or the ACF plot (Figure 5), which shows the correlation between data points based on lags ($acf(h) = Corr(X_t, X_{t-h})$), we see that almost all autocorrelations fall within the 95 % confidence bands that are generated under the white noise assumption. We notice that the autocorrelation at lag 22 fall a little outside of the confidence band. Since this is not a significant autocorrelation, we can reasonably conclude that the residuals of our model are more or less like white noise.

Furthermore, we also notice from the Ljung-Box statistic, a type of statistics test that determines whether the autocorrelations for the errors or residuals are non zero, that the p-values are all well above our chosen significance level. This implies that our model errors all have autocorrelations of zero, therefore, we might want to trust our model output and believe that this ARIMA(1,1,0) model maybe a good fit for our data.

2.2 ARIMA(1,1,0) x $(0,0,1)_{22}$

From the ARIMA(1,1,0) model given from the auto.arima() function earlier, we have noticed a single spike in the ACF plot that falls slightly out of the 95% confidence band at lag 22. To further improve our model, we can deduce a seasonal MA(1) model with lag 22, together with the ARIMA(1,1,0) given earlier, to get the SARIMA Model: ARIMA(1,1,0) x $(0,0,1)_{22}$. This is represented by the equation $(1-\phi_1 B)\nabla X_t = (1+\Theta_{22}B^{22})Z_t$ with, again, Z_t as white noise, and ϕ and Θ are the parameters to be estimated.

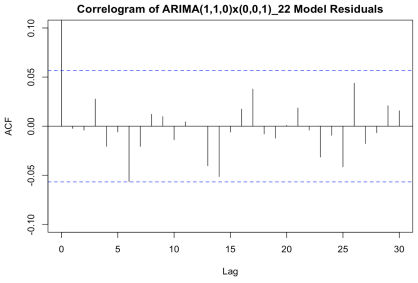

Figure 6: ACF of residuals with 95% confidence bands

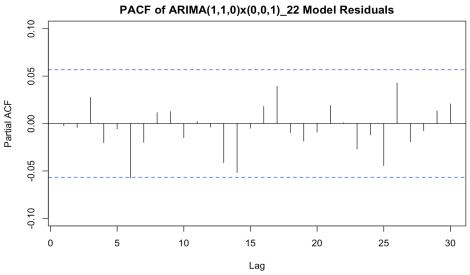

Figure 7: PACF of residuals with 95% confidence bands

The PACF plot here, or the partial autocorrelation function, gives the partial correlation of a stationary times series with its own lagged values. It is based on the correlation of observations with the effect of the intervening observations removed. From examining both the ACF and PACF plots in Figure 5 and 6, we do not see any significant lags, in contrast to the Correlogram of the ARIMA(1,1,0) plot shown in Figure 5. This means that the plots more or less simulate that of a white noise series. Similarly, the large p-values we see in the Ljung-Box statistics also support the adequacy of the model. Therefore, we have good reasons to believe in the sufficient fit of our model.

2.3 ARIMA(0,1,0) x $(1,0,0)_{22}$

Typically, in order to speculate the number of parameters needed in a model, we can examine the SARIMA diagnostics as well as look at both the ACF and PACF plots. Without using the suggested model given by the auto.arima() function, we started by looking at the SARIMA diagnostic without any parameters as input (Figure 8):

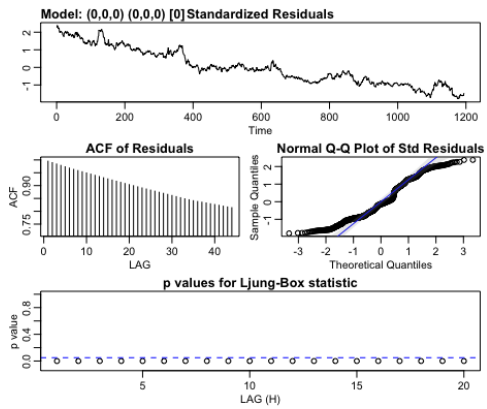

Figure 8: Sarima Diagnostics

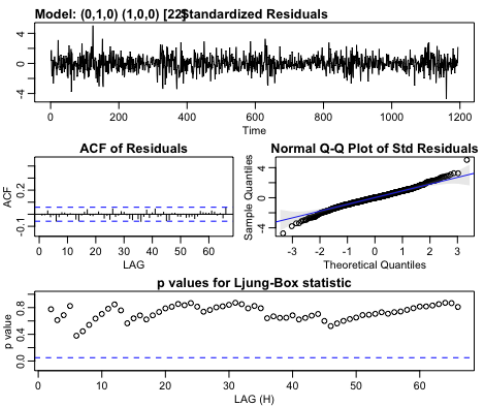

Figure 9: Sarima Diagnostics

Since the series is not stationary, the simplest possible model we want to try is a random walk model, which can be considered as a limiting case of an AR(1) model, or a series with infinitely slow mean reversion. With the PACF plot showing a spike at lag 22, we are adding a seasonal AR(1) component to the random walk model, producing our final model ARIMA(0,1,0) x $(1,0,0)_{22}$. This could be written as $\nabla X_t = (1 + \Phi_{22}B^{22})Z_t$ with, again, Z_t as white noise, and Θ is the parameter to be estimated.

The result of the diagnostic shown in Figure 9 with reasonably stationary residuals, zero significant lags in the ACF plot, as well as large p-values for the Ljung-Box statistics further infer that that this model could be considered as an adequate model.

2.4 ARIMA(2,1,4) x $(0,0,1)_{22}$

So far, we have used both auto.arima() and manual derivation to come up with the number of parameters needed for our model. Now, we would like to utilize a strategy or method commonly used by traders to help them with analyzing the stock market - moving average, specifically, 10-day moving average. 10-day moving average is a trend following indicator that gives you a visual of how strong a security a trending. We would like to test out both the exponential moving average (gives more weight to the most recent price action) and simple moving average (uses the unweighted mean of the previous n data) with a period of 10 to understand its effect our fitted model.

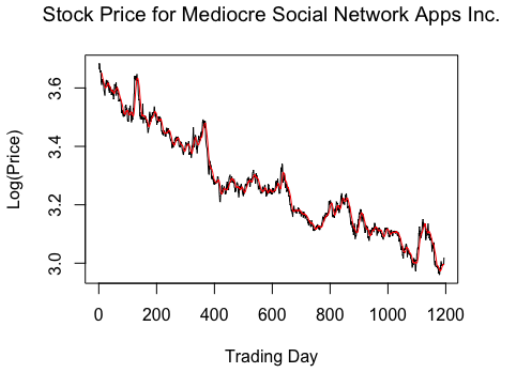

Figure 10: 10-Day Simple Moving Average

Applying, $SMA()$ to our logged time series data, we see in Figure 10 that the trend becomes smoother as the moving average can smooth out short-term fluctuations and highlight longer-term trends or cycles. Returning the smoothed out trend to auto.arima()function, we are given the ARIMA(2,1,4) model. However, to be clear, we will not fit a sarima model to the smoothed version of the data. We are only using the auto.arima() 's suggested model and will still fit it to the original logged time series data, so that it will be comparable to the other models described earlier. Using the same method but for EMA(), we actually obtain ARIMA(1,1,0) again from the auto.arima() function which is a model we have already used previously. Therefore, we will preceed with ARIMA(2,1,4).

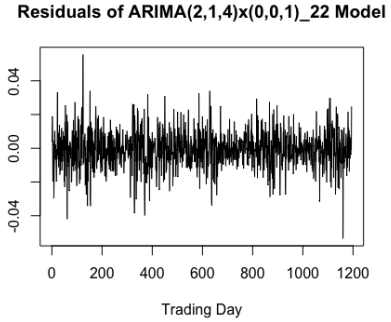

Figure 11: Residuals Plot

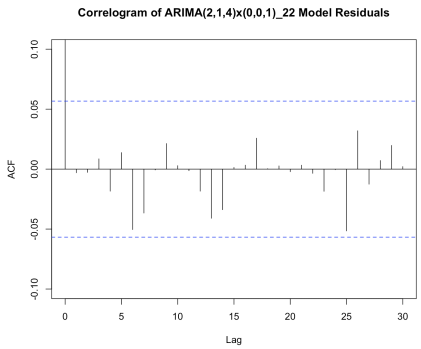

Figure 12: ACF of residuals with 95% confidence bands

Again, adding a seasonal MA(1) model with a period of 22 like our ARIMA(1,1,0) x (0,0,1) with period 22 earlier, we come to our resulting model which is ARIMA(2,1,4) x $(0,0,1)_{22}$. The equation is: $(1-\phi_1 B-\phi_2 B^2)\nabla X_t = (1+\Theta_{22}B^{22})(1+\theta_1 B+\theta_2 B^2+\theta_3 B^3+\theta_4 B^4)Z_t$ with, again, Z_t as white noise, and ϕs, θs, and Θ are the parameters to be estimated.

To ensure that this model is a good fit, we once again look at the residuals of the model as well as the correlogram. Figure 11-12 shows that the residuals look somewhat like white noise and the there are no significant lags in the ACF of the residuals. Even though there are one or two p-values in the Ljung-Box statistics that are relatively smaller compare to the previous models, they are mostly still large and statistically insignificant. Therefore, we can consider this model to be an adequate model.

2.5 Linear Model with AR(2) residuals

As mentioned previously in the Exploratory Data Analysis section, we notice a large variation in the peaking periods over the years. To further understand the underlying seasonality, we will split our data by year (Figure 13):

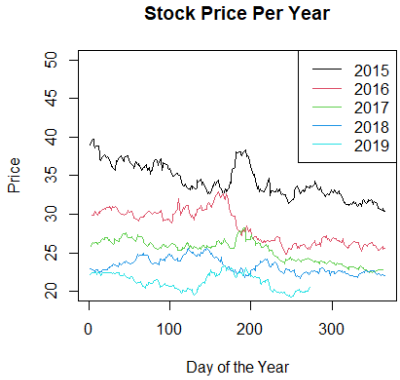

Figure 13: The stock price of each year from 2015-2019

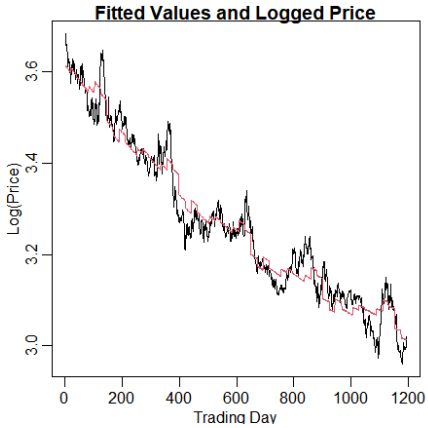

Figure 14: linear model of logged price

From Figure 13, we notice that almost every year contains one peak except for 2018, which has two consecutive and relatively smoother peaks. Since our 10 predicted data will occur right after a complete peak phase in 2019, we will treat 2018 as a regular year, one with only one peak. We want to take these peaks that occur during certain months of the year into consideration while also including weekdays as indicator varaibles. This is because volatility in the stock market can vary depending on the day of the week.

We first fit a linear model to the logged stock price with month and weekday indicators to account for the potential seasonality in month and weekday. The linear model seems to capture the linear trend and part of the seasonality (Figure 14). Then we use auto.arima() function to fit an ARIMA model to the residuals and we get the ARIMA(2,0,0) or the AR(2) model.

The parameters for our linear model with residuals of AR(2) model are shown in Figure 15. The ARIMA model for residuals is $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$ where Z_t is white noise and ϕ_1 is 0.9787 and ϕ_2 is -0.0296.

```
Coefficients:
(Intercept)         time    I(time^2)       month2       month3       month4       month5       month6
  3.615e+00   -7.823e-04    2.537e-07    1.009e-02    6.012e-03    4.220e-03    1.751e-02    4.189e-02
     month7       month8       month9      month10      month11      month12     weekday3     weekday4
  3.166e-02   -1.254e-02   -2.887e-02   -7.686e-05   -1.498e-02   -1.655e-02   -1.276e-03   -1.347e-03
    weekday5     weekday6
 -1.341e-03   -2.337e-04
```

Figure 15: parameters of the linear model

The Ljung-Box statistics plot indicates that the model fits quite well as almost all p-values are above the line. The acf plot also shows all of the plot being inside the confidence bands. The residual plot also yields a process that looks somewhat stationary. However, in the normal Q-Q plot, a tool that helps to assess if a set of data plausibly came from some theoretical distribution such as Normal, the points fall along a line in the middle, but curve off in the extremities. This behavior means that the data might

have more extreme values than would be expected if they came from a Normal distribution. In general, this model could suffice as a good fit.

3 Model Comparison and Selection

3.1 Information Criterion

Model Name	Description	AIC	BIC	AICc
model1	ARIMA(1,1,0)	-6.140505	-6.12772	-6.140496
model2	ARIMA(1,1,0) x $(0,0,1)_{22}$	-6.144138	-6.127091	-6.144121
model3	ARIMA(0,1,0) x $(1,0,0)_{22}$	-6.14596	-6.133175	-6.145952
model4	ARIMA(2,1,4) x $(0,0,1)_{22}$	-6.151631	-6.113275	-6.151529

Table 1: These are information criterion (AIC, BIC and AICc) for our model.

Information criterion (eg. AIC, BIC and AICc) helps us measure in-sample fit and the complexity of each model. Therefore, the model which has the best information criterion indicates the best in-sample fit without overfitting. AIC, AICc, and BIC are three different numerical criterion we used to evaluate the quality of our models. Calculation of each information criterion contains a negative log component on the likelihood of the model and a penalty function that takes into account of factors that determine sample size and the number of parameters used. The difference is that AIC prefers a more complicated and less parsimonious model compared to BIC since BIC penalizes model complexity more heavily. Similarly, AICc also corrects for AIC's tendency to overfit, which can result from adding too many parameters in order to increase likelihood. For all three information criterion, a smaller value is preferred.

Table 1 shows that the model 4 technically has the lowest AIC and AICc. For BIC, model 3 tops all four options. That is to say, we do not have enough information to determine which model is the best by simply comparing information criterion.

3.2 Cross Validation Model Comparison

In order to predict the future stock price with our best model, we also need to conduct a cross validation test to understand the ability of each model to make out-out-sample prediction. This test is necessary because it is a very useful technique in assessing the effectiveness of our models, particularly in limiting problems like overfitting, underfitting, and understanding how our models will generalize to an independent data set. This also helps determine the hyperparameters in our model, or which parameters will result in lowest test error.

To do this, we set the proportion of the training set and test set to be 8:2 and use training data to train the model. Recall that four of our five models are SARIMA models with various possible parameters and the other one is a linear model with an AR model used on its residuals.

Model Name	Description	MSE
model1	ARIMA(1,1,0)	0.2994924
model2	ARIMA(1,1,0) x $(0,0,1)_{22}$	0.2918652
model3	ARIMA(0,1,0) x $(1,0,0)_{22}$	0.2913488
model4	ARIMA(2,1,4) x $(0,0,1)_{22}$	0.2892622
model5	lm(month+weekday)+AR(2)	0.3893195

Table 2: These are the out-of-sample mean squared erros for our models of interest. There are many ways to do cross validation and there are many diagnostics you can look at. MSE is one example.

Table 2 shows that model 4 has the lowest MSE, which suggests that this model is the most robust when forecasting out-of-sample. Recall from the result of the model chosen by information criterion, we know that model 4 also performs well according to AIC and AICc. Notably, even though model 4 is the most complicated SARIMA model of all candidates, it still performs well when we include the complexity test by the information criterion, exhibiting the superiority of the model. Therefore, we will use model 4 as our chosen model to conduct further analysis.

4 Results

Based on our analysis, we have modeled the stock price over time as the model Our model is defined in equation (1).

$$(1 - \phi_1 B - \phi_2 B^2)\nabla X_t = (1 + \Theta_{22} B^{22})(1 + \theta_1 B + \theta_2 B^2 + \theta_3 B^3 + \theta_4 B^4) Z_t \tag{1}$$

where Z_t is the white noise and $\phi_1, \phi_2, \phi_3, \phi_4, \Theta_{22}, \theta_1, \theta_2, \theta_3, \theta_4$ are the parameters. (Table 3).

4.1 Estimation of model parameters

Parameter	Estimate	(s.e)
ϕ_1	0.3709	(0.0103)
ϕ_2	-0.9640	(0.0086)
θ_1	-0.3793	(0.0319)
θ_2	0.9839	(0.0356)
θ_3	0.0160	(0.0356)
θ_4	-0.0115	(0.0326)
Θ_{22}	-0.0723	(0.0291)

Table 3: These are our parameter estimates and corresponding standard errors for the ARIMA(2,1,4) x $(0, 0, 1)_{22}$ model in Equation 1.

4.2 Prediction

Using our model to predict the stock price for the next 10 trading days, or the initial 10 weekdays from October 1st, 2019 , we generate the following forecast:

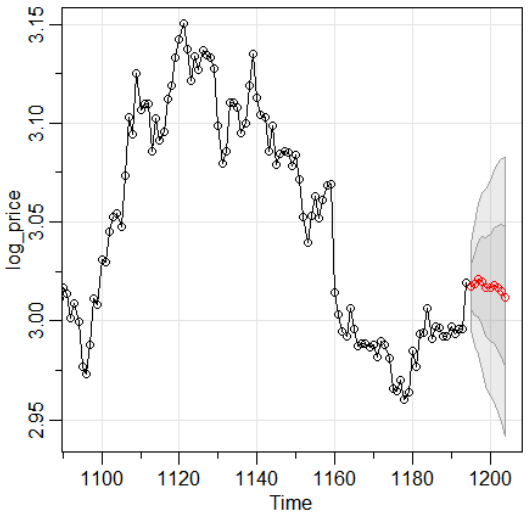

Figure 16: Forecast of Logged Price

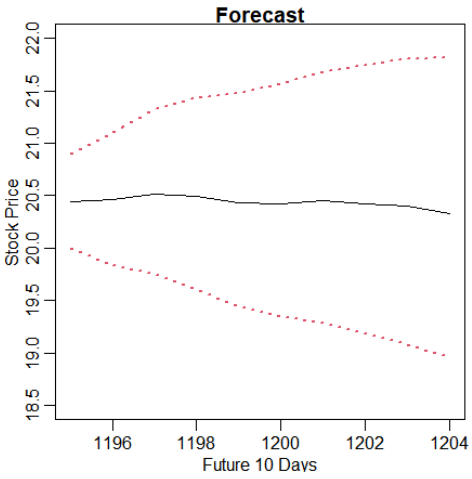

Figure 17: Forecast of Price with 95% Confidence Intervals

The forecast of Mediocre Social Network Apps Inc.'s stock price for the first ten trading days of October 2019 helps us to understand what the company's fourth quarter turnout could possibly be. Realistically, it isn't very likely that things will turn around in the next 10 days as the stock prices we predict still trend downward. This makes sense due to the long-term downward trend the stock has been facing for the past few years.

However, if we examine the 95% confidence intervals, we see that it is still possible for the company to experience a short-term upward trend. Likely, this would require the company to go through a restructuring of its business or a long-term strategic change. If the company does nothing going forward, it would be difficult to expect a satisfactory fourth quarter.

5. Appendix

```{r}
library(astsa)
library(ggplot2)
stocks <- read.csv("stocks.csv")
tail(stocks)

plot(stocks$X,stocks$Price,type='l',xlab="Trading Day",ylab="Price",main = "Stock Price for Mediocre Social Network Apps Inc.", font.main=1)

#Variance Transform
price_ts <- as.ts(stocks$Price)
log_price <- log(price_ts)
plot.ts(price_ts,type="l")
plot.ts(log_price,type="l")

#differencing
d1_price <- diff(log_price)
plot.ts(d1_price)
acf2(d1_price)
acf2(diff(d1_price),period=22)

#ACF/PACF
acf2(d1_price,max.lag = 200)
```

```{r}
#Indicator
library(lubridate)
stocks$month<-as.factor(month(stocks$Date))
stocks$year<-as.factor(year(stocks$Date))
stocks$day<-yday(stocks$Date)
stocks$weekday<-as.factor(wday(stocks$Date))
levels<-levels(stocks$year)
time<-1:nrow(stocks)
stocks$log_price<-log(stocks$Price)
stocks$time<-time
```

#10day moving average

```{r}
price_ts
log_price
ma10

library(TTR)
ma10 <- EMA(log_price) #Exponential Moving Average - More weights on more recent data
ma10_SMA <- SMA(log_price) #Simple Moving Average - Equal Weights
auto.arima(ma10_SMA)

plot(log_price,xlab="Trading Day",ylab="Log(Price)",main = "Stock Price for Mediocre Social Network Apps Inc.", font.main=1)
lines(as.numeric(ma10_SMA),col=2)
```

#Model
```{r}
library(forecast)
auto.arima(log_price)
acf2(diff(log_price),max.lag = 50)
acf(log_price)

#####MODEL 1
model1 <- sarima(log_price,p=1,d=1,q=0,S=0,P=0,D=0,Q=0)

test1<- arima(log_price, order= c(1,1,0))
plot(residuals(test1),ylab="",xlab="Trading Day",main="Residuals of ARIMA(1,1,0) Model")
p1 <- acf(residuals(test1))
plot(p1, main= "Correlogram of ARIMA(1,1,0) Model Residuals",ylim = c(-0.1,0.1))
pacf(residuals(test1),ylim=c(-0.1,0.1))


######MODEL2
model2 <- sarima(log_price,p=0,d=1,q=0,S=22,P=1,D=0,Q=0)
test2 <- arima(log_price, order= c(0,1,0), seasonal = list(order=c(1,0,0),period=22))
acf2(test2$residuals)

test2_1<-sarima(log_price,p=0,d=0,q=0,S=0,P=0,D=0,Q=0)
test2_2<- arima(log_price, order= c(0,1,0),seasonal = list(order=c(1,0,1),period=22))
pacf(test2_2$residuals,ylim=c(-0.1,0.1))
acf(test2_2$residuals,ylim=c(-0.1,0.1))
```

```
plot(test2_2$residuals,xlim=c(0,200))

acf2(log_price,max.lag = 40)

######MODEL3
model3 <- sarima(log_price,p=1,d=1,q=0,S=22,P=0,D=0,Q=1)
test3 <- arima(log_price, order= c(1,1,0), seasonal = list(order=c(0,0,1),period=22))
plot(test3$residuals,ylab="",xlab="Trading Days",main="Residuals of ARIMA(1,1,0)x(0,0,1)_22",font.main=1)

acf(test3$residuals,ylim=c(-0.1,0.1))
title("Correlogram of ARIMA(1,1,0)x(0,0,1)_22 Model Residuals")

pacf(test3$residuals,ylim=c(-0.1,0.1))
title("PACF of ARIMA(1,1,0)x(0,0,1)_22 Model Residuals")


######MODEL 4
model_SMA10 <- sarima(log_price,p=2,d=1,q=4,S=22,P=0,D=0,Q=1)

model_SMA10_base <- arima(log_price, order= c(2,1,4),seasonal = list(order=c(0,0,1),period=22))
plot(model_SMA10_base$residual,ylab="",xlab="Trading Day",main="Residuals of ARIMA(2,1,4)x(0,0,1)_22 Model")

p2 <- acf(model_SMA10_base$residuals)
plot(p2, ylim=c(-0.1,0.1), main = "Correlogram of ARIMA(2,1,4)x(0,0,1)_22 Model Residuals")

pred<-sarima.for(log_price,p=2,d=1,q=4,S=22,P=0,D=0,Q=1,n.ahead=10)
predicted<-exp(pred$pred)
predict<-data.frame(predicted)
upper<-exp(pred$pred+2*pred$se)
lower<-exp(pred$pred-2*pred$se)

plot(1195:1204,predicted,type="l",xlab="Future 10 Days",ylab="Stock Price",ylim=c(18.5,22),main="Forecast")
lines(1195:1204,upper,col=2,lty=3,lwd=2)
lines(1195:1204,lower,col=2,lty=3,lwd=2)


######Model 5
library(forecast)
model5_lm<-lm(data=stocks,log_price~time+I(time^2)+month+weekday)
```

```
model5_lm
plot.ts(log_price,xlab="Trading Day",ylab="Log(Price)",main="Fitted Values and Logged Price")
lines(time,model5_lm$fitted.values,col=2)
auto.arima(model5_lm$residuals)
residuals<-model5_lm$residuals
sarima(residuals,p=2,d=0,q=0,P=0,D=0,Q=0)
model5_arima<-arima(residuals,order=c(2,0,0))
model5_residuals<-model5_arima$residuals
plot(model5_residuals,ylab="",xlab="Trading Day",main="Residuals of lm_AR(2) Model")
```

#price over time of each year
```{r}
library(dplyr)
dat_2015<-filter(stocks,year==levels[1])
dat_2016<-filter(stocks,year==levels[2])
dat_2017<-filter(stocks,year==levels[3])
dat_2018<-filter(stocks,year==levels[4])
dat_2019<-filter(stocks,year==levels[5])
plot(dat_2015$day,dat_2015$Price,type="l",col=1,ylim=c(20,50),xlab="Day of the Year",ylab="Price",main="Stock Price by Year")
lines(dat_2016$day,dat_2016$Price,col=2)
lines(dat_2017$day,dat_2017$Price,col=3)
lines(dat_2018$day,dat_2018$Price,col=4)
lines(dat_2019$day,dat_2019$Price,col=5)
legend("topright",c("2015","2016","2017","2018","2019"),
col=c("1","2","3","4","5"),lty=1,cex=1)
```

```{r}
model1$AIC
model2$AIC
model3$AIC
model_SMA10$AIC

model1$BIC
model2$BIC
model3$BIC
```

```
model_SMA10$BIC


model1$AICc
model2$AICc
model3$AICc
model_SMA10$AICc
```

#Forecast
```{r}
#calculate cross validation for each model
start_day <- 900
end_day <- 1194
#10-20% for test set

sum_squared_errors <- c(model1=0, model2=0, model3=0, model_SMA10=0,model5=0)
# calculate cv for 4 sarima model
for (day in seq(start_day,end_day,by=10)) {
  train_set <- window(log_price, end=day-0.001)
  test_set<- window(stocks$Price,start=day,end=day+10-0.01)

  forecast1 <- sarima.for(train_set, n.ahead=10, p=1,d=1,q=0,S=0,P=0,D=0,Q=0)$pred
  forecast2 <- sarima.for(train_set, n.ahead=10, p=0,d=1,q=0,S=22,P=1,D=0,Q=0)$pred
  forecast3 <- sarima.for(train_set, n.ahead=10, p=1,d=1,q=0,S=22,P=0,D=0,Q=1)$pred
  forecast4 <- sarima.for(train_set, n.ahead=10, p=2,d=1,q=4,S=22,P=0,D=0,Q=1)$pred

  sum_squared_errors[1] = sum_squared_errors[1] + sum((exp(forecast1) - test_set)^2)
  sum_squared_errors[2] = sum_squared_errors[2] + sum((exp(forecast2) - test_set)^2)
  sum_squared_errors[3] = sum_squared_errors[3] + sum((exp(forecast3) - test_set)^2)
  sum_squared_errors[4] = sum_squared_errors[4] + sum((exp(forecast4) - test_set)^2)
}
#calculate lm_AR(2) model
for (day in seq(start_day,end_day,by=10)) {
  # spilt our data
  train_order<-window(1:1194, end=day-0.001)
  test_order<-window(1:1194,start=day,end=day+10-0.01)
  train_set<-stocks[train_order,]
  test_set<-stocks[test_order,]
```

```
  #lm model with indicator month and weekday
  model5_lm<-lm(data=train_set,log_price~time+I(time^2)+month+weekday )
  forecast51<-predict(model5_lm,test_set,n.ahead=10)
  model5_arima<-arima(model5_lm$residuals,order=c(2,0,0))
  forecast52<-predict(model5_arima,n.ahead=10)
  forecast5<-forecast51+forecast52$pred
  sum_squared_errors[5] = sum_squared_errors[5] +
sum((exp(forecast5)-test_set$Price)^2)
  }
sum_squared_errors/(end_day - start_day + 1)
```

```{r}
stock_price<-stocks$Price
zy_pred <- sarima.for(stock_price[1:1184], n.ahead=10,
p=1,d=1,q=0,S=0,P=0,D=0,Q=0)$pred

yf_pred <- sarima.for(stock_price[1:1184], n.ahead=10,
p=1,d=1,q=0,S=22,P=0,D=0,Q=1)$pred

ds_pred <- sarima.for(stock_price[1:1184], n.ahead=10,
p=0,d=1,q=0,S=22,P=1,D=0,Q=0)$pred

sma_pred <- sarima.for(stock_price[1:1184], n.ahead=10,
p=2,d=1,q=4,S=22,P=0,D=0,Q=1)$pred

model5_lm<-lm(data=stocks[1:1184,],Price~time+I(time^2)+month+weekday )
forecast51<-predict(model5_lm,stocks[1185:1194,],n.ahead=10)
model5_arima<-arima(model5_lm$residuals,order=c(2,0,0))
forecast52<-predict(model5_arima,n.ahead=10)
linear_pred<-forecast51+forecast52$pred
```

draw prediction plot of the last 10 points
```{r}
plot(x=c(1185:1194),y=stock_price[1185:1194],type='l',col=1,ylim =c(19.8,20.5),xlab
='Trading Day',ylab ='Price',main='Stock Price Prediction of Each Model')
lines(x=c(1185:1194),y=zy_pred,type='l',col=2)
lines(x=c(1185:1194),y=yf_pred,type='l',col=3)
lines(x=c(1185:1194),y=ds_pred,type='l',col=4)
lines(x=c(1185:1194),y=sma_pred,type='l',col=5)
lines(x=c(1185:1194),y=linear_pred,type='l',col=6)
```

```
legend('topleft', legend=c("True Value","ARIMA(1,1,0)","ARIMA(1,1,0)*(0,0,1)_22","ARIMA(0,1,0)*(1,0,0)_22","ARIMA(2,1,4)*(0,0,1)_22","lm(month+weekday)+AR(2)"),col=c(1:6),lty=1,cex=0.7)
```