STAT 153 Project

Group: Charis Chan, Jeffrey Chen, Devin Hua, Eunice Sou, Anthony Yu

November 22, 2019

1 Overview

We analyzed the stock price data set using a combination of linear and quadratic models for a total of four models. Unfortunately, things seem to be rather grim for the Mediocre Social Network Apps Incorporated. From September of 2015 to September 2019, their stock has been on a steady decline. With our predictions for the first 10 trading days in the month of October using an ARIMA(1, 1, 1) model after differencing the data, things don't seem to improve for the company.

2 Exploratory Data Analysis

We begin by performing exploratory data analysis. Let's see the stock price starting from 2015 (Figure 1):

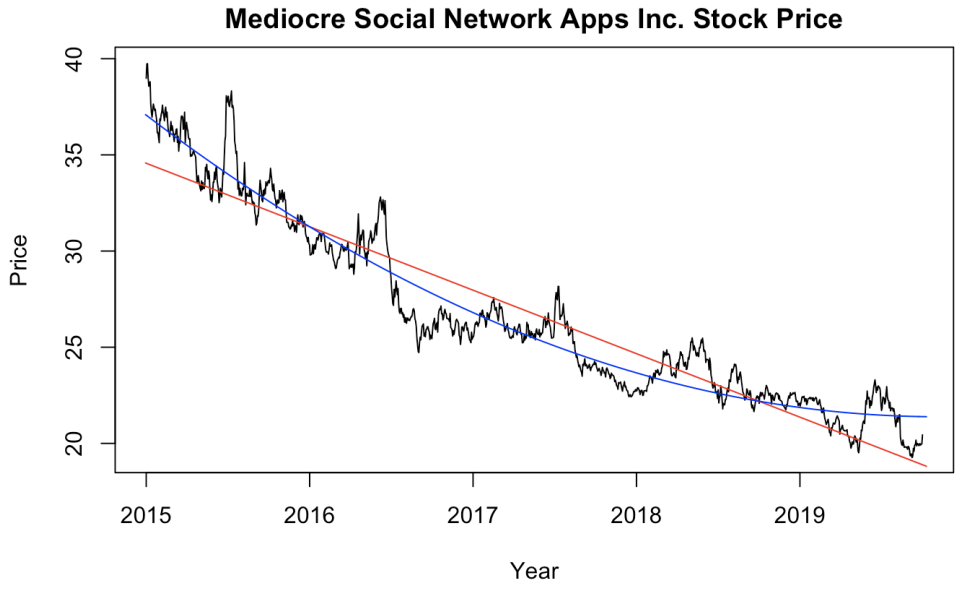

Figure 1: Mediocre Social Network Apps Inc. Stock Price from Jan. 2015 to Sep. 2019. The colored curves show fitted trends to the series: the red curve shows the fitted linear model, and the blue curve shows the fitted quadratic model.

We clearly see a negative trend, and on first glance it appears to be roughly linear. With the negative slope slowing down towards the end though, the series may be quadratic too.

We note that there are some hints of seasonality, with upward peaks at around the

middle of each year. However, it is not very distinct nor consistent (in terms of time of year and how these peaks are shaped), so we decided that these peaks were not due to seasonality.

We then proceed to make our data stationary in the next section.

3 Pursuing Stationarity

By the residual plots (Figure 2), we note slight heteroskedasticity, with more variance appearing in the beginning of the series. However, taking the log on the data seems to stabilize the variance, despite the variance decreasing with time, contrary to where we typically use the log transformation for quadratic increases in variance.

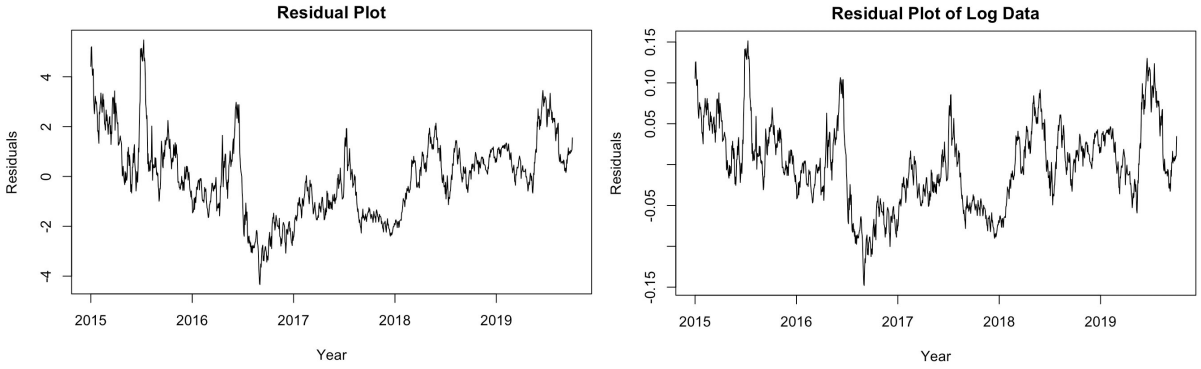

Figure 2: Residual plots of the original data and the log-transformed data.

With the log-transformed data, we are now ready to fit different ARIMA models and compare their performances.

4 ARIMA Model Selection

We tried the following ARIMA models (we will refer these models as model 1 to model 4):

1. Differencing by 1 lag, ARIMA(1, 1, 1)
2. Differencing by 5 lags, ARIMA(1, 1, 4)x(1, 0, 1)[5]
3. Differencing by 5 lags, ARIMA(4, 0, 4)x(1, 0, 1)[5]
4. Differencing by 5 lags, fitting to a quadratic model, ARIMA(3, 0, 2)x(1, 0, 1)[5]

We differenced by 1 lag to remove the trend. We also tried differencing by 5 lags to consider data by the week, since the stock market is only open 5 days a week (Monday to Friday). We first check the residuals of taking this differenced data in Figure 3: if it looks like white noise, then taking this difference is sensible since we have successfully removed potential trends or seasonalities.

To justify why we fit the models above, we turn to ACF and PACF plots (Figure 4) to give insights if there are any autoregressive (AR) or moving average (MA) components.

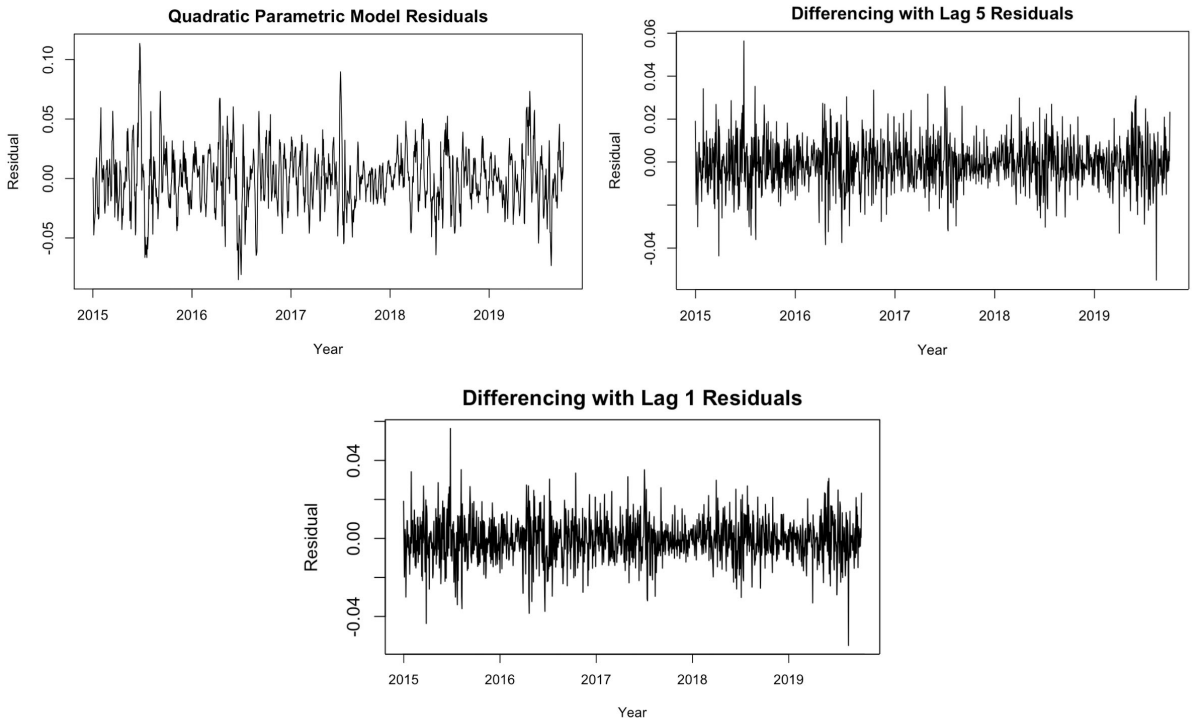

Figure 3: Residual plots after differencing data.

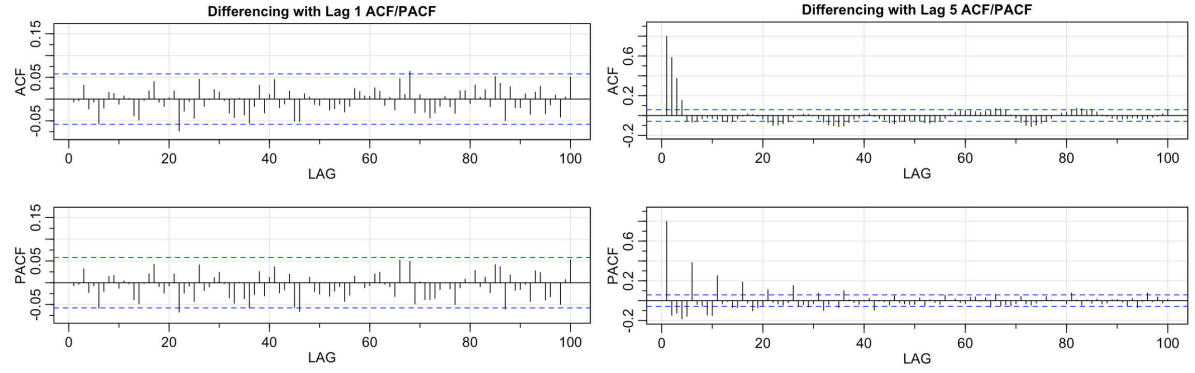

Figure 4: ACF and PACF plots for first differences and fifth differences of the data.

For the first difference, the ACF/PACF looks like white noise, as they are fluctuating around 0 with less than 5% of the spikes exceeding the blue confidence bands. Since this model has a non-zero mean for the price, we tried an ARIMA(0, 1, 0) model, but there were peaks in the ACF after we fitted this model, so we tried ARIMA(1, 1, 1). This model performed better, so we used this model instead.

For the fifth difference, we had more peaks, and subsequently more models, to consider. Without considering a quadratic curve, we decided to try both ARIMA(1, 1, 4)x(1, 0,

1)[5] and ARIMA(4, 0, 4)x(1, 0, 1)[5]. There are spikes in the ACF plot from 1 to 4, so we chose to include 4 MA components. For the PACF, there is a spike at 1, but it seems to be repeated every 5 lags. This is why we included an AR component and a seasonal AR component, denoted in the second multiplicative component. When we included a quadratic component, we fit a ARIMA(3, 0, 2)x(1, 0, 1)[5] model, which seemed to fit slightly better than other ARIMA models and had better AIC/AICc/BIC values.

After getting these candidate models, we can now compare them by observing p-values of Ljung-Box statistics, AIC/AICc/BIC, and the sum of squared errors (SSE) from cross validation.

4.1 p-value Comparison of Ljung-Box Statistics

One way we can compare evaluate the validity of our models is by looking at the p-values of the Ljung-Box statistic for each model (Figure 5). Unlike hypothesis testing, we want the p-values to exceed 0.05, which represents that the residuals are independent (the alternate hypothesis assumes the residuals are dependent, which we do not want in a stationary model).

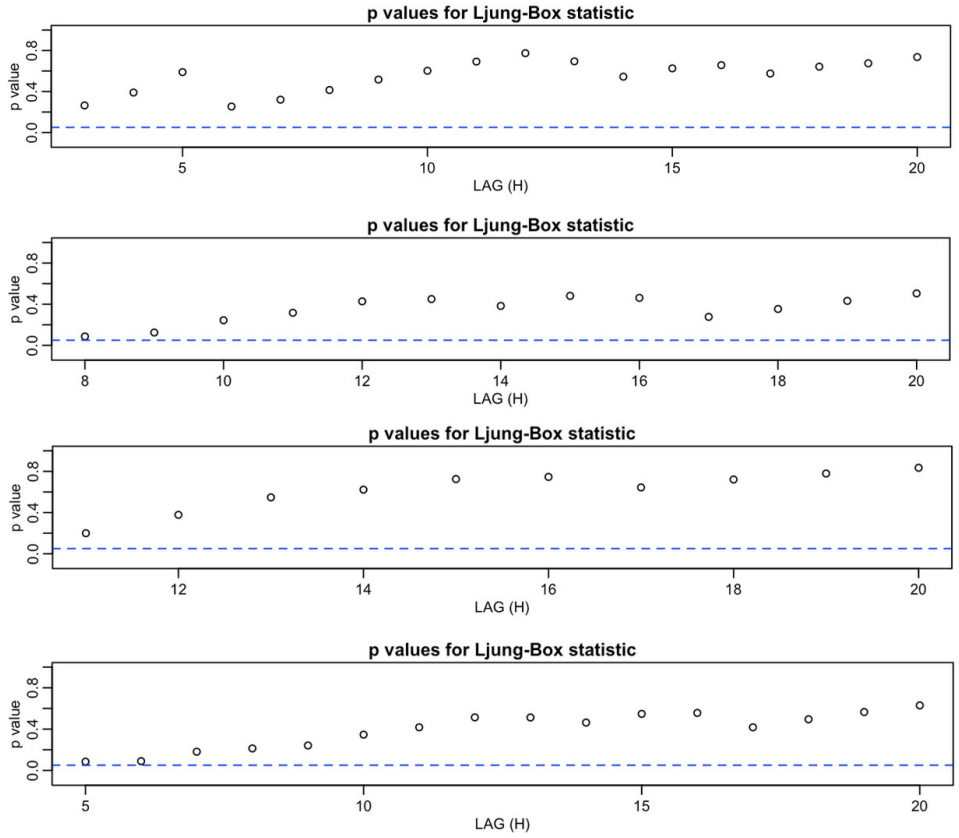

Figure 5: p-values of the Ljung-Box statistic for each model. The first plot corresponds to Model 1, the second to Model 2, and so on.

All of these models seem to fit well, but it is worth noting that Model 1 & 3 have higher p-values (ranging from 0.2 to 0.8) compared to the other models, some of which get close to the blue dashed line at 0.05, and Model 2 specifically only has the highest p-value at 0.5. By these plots, we would think Model 1 and Model 3 would be better picks.

4.2 AIC/AICc/BIC Comparison

Another way we can compare our models is using different information criterion (IC): AIC, AICc, and BIC. Each evaluate models in a similar way: it takes the negative log likelihood and adds a penalty for model complexity, so we want to minimize these IC's as much as possible.

Model	AIC	AICc	BIC
1. First difference, ARIMA(1, 1, 1)	-6.1324	-6.1323	-6.1153
2. Fifth difference, ARIMA(1, 1, 4)x(1, 0, 1)[5]	-6.1201	-6.1200	-6.0816
3. Fifth difference, ARIMA(4, 0, 4)x(1, 0, 1)[5]	-6.1211	-6.1209	-6.0698
4. Fifth difference, quadratic, ARIMA(3, 0, 2)x(1, 0, 1)[5]	-6.1312	-6.1311	-6.0928

Table 1: The IC's for each model.

From Table 2, we see that Model 1 has the smallest AIC, AICc, and BIC. Although the differences seem very small, since we have log-transformed the data, the differences in these metrics are also scaled down by the log transformation. It is also worth noting that Model 1 is much simpler than the others in terms of the number of parameters, which may be why it performs better here.

4.3 Cross Validation Comparison

Finally, we can use cross validation to verify each model's performance. The idea is to hold out some data, then use the remaining data to predict the price the next day (see Section 6.4.3 in the Appendix for the code implementation). We incrementally increase the amount of data to predict future data points, then sum the squared errors to evaluate each model. Lower errors means the model fits better.

Model	SSE
1. First difference, ARIMA(1, 1, 1)	0.02303
2. Fifth difference, ARIMA(1, 1, 4)x(1, 0, 1)[5]	0.09813
3. Fifth difference, ARIMA(4, 0, 4)x(1, 0, 1)[5]	0.07898
4. Fifth difference, quadratic, ARIMA(3, 0, 2)x(1, 0, 1)[5]	0.08179

Table 2: The sum of squared errors (SSE) for each model.

We see that Model 1 again performs the best with the smallest SSE by a considerable margin. Based on all our comparisons, Model 1 always performed well, so we conclude that taking the first difference of the data, then fitting an ARIMA(1, 1, 1) model is the most appropriate.

5 Results

Our simplest model turned out to work the best by our analysis. This ARIMA model, written out, is defined in equation (1).

$$(1 - \phi_1 B)\nabla(X_t - \mu) = (1 - \theta_1 B)Z_t \tag{1}$$

5.1 Estimation of model parameters

After fitting the model, these are the parameter estimates for the model:

Parameter	Estimate (s.e)
ϕ_1	-0.0065 (0.0291)
θ_1	-1.0000 (0.0023)
μ	5.9×10^{-7} (0.00003)

Table 3: Parameter estimates and corresponding standard errors for the ARIMA model in equation 1.

5.2 Prediction

Using the selected ARIMA model, we now make predictions for the next 10 trading days for the stock in Figure 6:

Figure 6: Predictions (in red) of the next 10 trading days.

Because of the model's simplicity and the volatility of stock prices in general, it's no surprise that we would not capture all fluctuations, which would be more like noise than an actual pattern. Considering the downward trend by our predictions, the stocks for Mediocre Social Network Apps Incorporated will continue to drop in the beginning of October, and it does not seem like things will turn around for them.

6 Appendix

6.1 Dependencies

```
library(astsa)
```

6.2 Exploratory Data Analysis

```
# Run `setwd()`, passing in the file path of
# the directory your R script is in
stocks <- read.csv("stocks.csv")

quadratic <- function(x) {
  return (1.057e-05 * x^2 + -2.577e-02 * x + 37.13)
}
linear <- function(x) {
  return (-0.01314 * x + 34.61602)
}
plot.ts(stocks$Price)
curve(linear, from = 0, to = 1200, add = T, col = 'red')
curve(quadratic, from = 0, to = 1200, add = T, col = 'blue')
```

6.3 Pursuing Stationarity

```
time <- 1:length(stocks$Price)
no_transform_model <- lm(log(stocks$Price) ~ time)
plot(no_transform_model$residuals, type = 'l',
     main = "Residual Plot",
     ylab = "Residuals")
log_transform_model <- lm(log(stocks$Price) ~ time)
plot(log_transform_model$residuals, type = 'l',
     main = "Residual Plot of Log Data",
     ylab = "Residuals")
```

6.4 ARIMA Model Selection

6.4.1 ACF/PACF plots

```
differenced <- diff(log(stocks$Price), lag = 1)
acf2(differenced)
differenced5 <- diff(log(stocks$Price), lag = 5)
acf2(differenced5)
```

6.4.2 SARIMA output

```
# Model 1
sarima(differenced, 1, 1, 1)
# Model 2
sarima(differenced5, 1, 1, 4, P = 1, D = 0, Q = 1, S = 5)
```

```
# Model 3
sarima(differenced5, 4, 0, 4, P = 1, D = 0, Q = 1, S = 5)
# Model 4
time <- 1:length(differenced5)
time_sq <- time^2
param_model <- lm(differenced5 ~ time + time_sq)
sarima(param_model$residuals, 3, 0, 2, P = 1, D = 0, Q = 1, S = 5)
```

6.4.3 Cross Validation

```
start_index <- 1160
end_index <- length(time)
sum_squared_errors <- c(model1 = 0, model2 = 0, model3 = 0, model4 = 0)
for (index in start_index:end_index) {
  train_set1 <- window(differenced, end = index)
  test_set1 <- window(differenced, start = index, end = index + 1-0.1)

  train_set2_3 <- window(differenced5, end = index)
  test_set2_3 <- window(differenced5, start = index, end = index + 1-0.1)

  train_set4 <- window(param_model$residuals, end = index)
  test_set4 <- window(param_model$residuals, start=index, end= index + 1-0.01)

  forecast1 <- sarima.for(train_set1, n.ahead=10, p=1, d=1, q=1,
                          P=0, D=0, Q=0, S=0)$pred
  forecast2 <- sarima.for(train_set2_3, n.ahead=10, p=1, d=1, q=4,
                          P=1, D=0, Q=1, S=5)$pred
  forecast3 <- sarima.for(train_set2_3, n.ahead=10, p=4, d=0, q=4,
                          P=1, D=0, Q=1, S=5)$pred
  forecast4 <- sarima.for(train_set4, n.ahead=10, p=3, d=0, q=2,
                          P=1, D=0, Q=1, S=5)$pred

  sum_squared_errors[1] = sum_squared_errors[1] + sum((forecast1 - test_set1)^2)
  sum_squared_errors[2] = sum_squared_errors[2] + sum((forecast2 - test_set2_3)^2)
  sum_squared_errors[3] = sum_squared_errors[3] + sum((forecast3 - test_set2_3)^2)
  sum_squared_errors[4] = sum_squared_errors[4] + sum((forecast4 - test_set4)^2)
}
sum_squared_errors
```

6.5 Results

6.5.1 Estimation of Model Parameters

```
arima_model <- sarima(differenced, 1, 1, 1)
arima_model$ttable
```

6.5.2 Predictions

```
sarima.for(stocks$Price, 10, 1, 1, 1)
```