Predict_413_Discussion_Week_7

Singh, Gurjeet

February 21, 2018

Contents

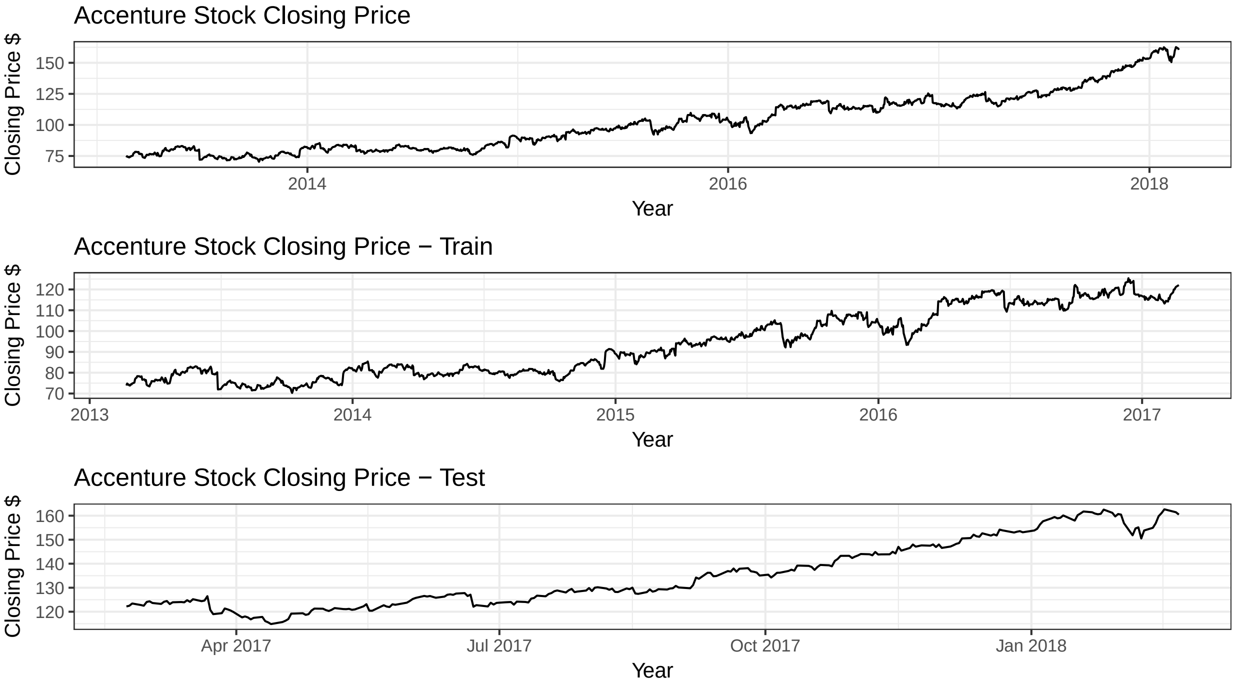
1. Introduction

For this week's discussion, we build the GARCH models using the fGarch packages. We then compare these models with other models developed in the previous weeks.

2. Data

For this week's discussion, I again chose ACN (Accenture) as my stock.

The figure below shows the Accenture stock closing price along with the plots for train and test data. Closing Price chart shows that the stock price has been increasing over the years. As expected from the stocks, there's no seasonality in the data. It is difficult to tell from the graph but can observe a lot of volatility in the data.


Accenture Stock Closing Price

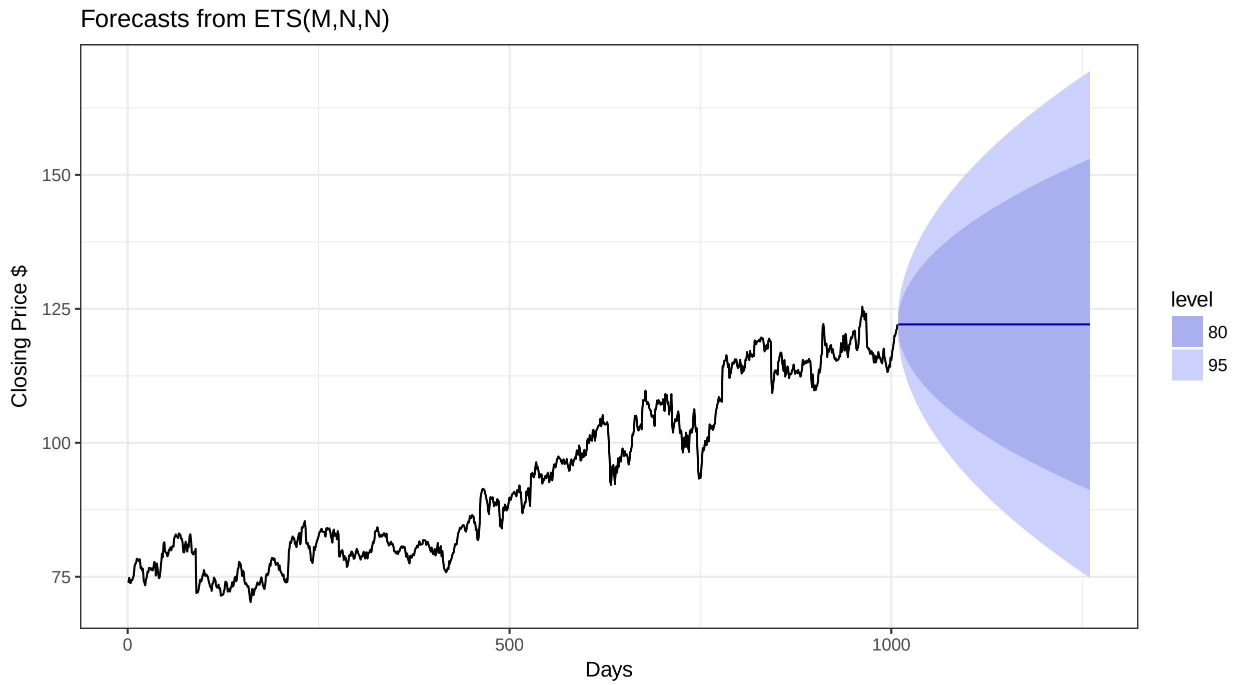
Accenture Stock Closing Price − Train

Accenture Stock Closing Price − Test

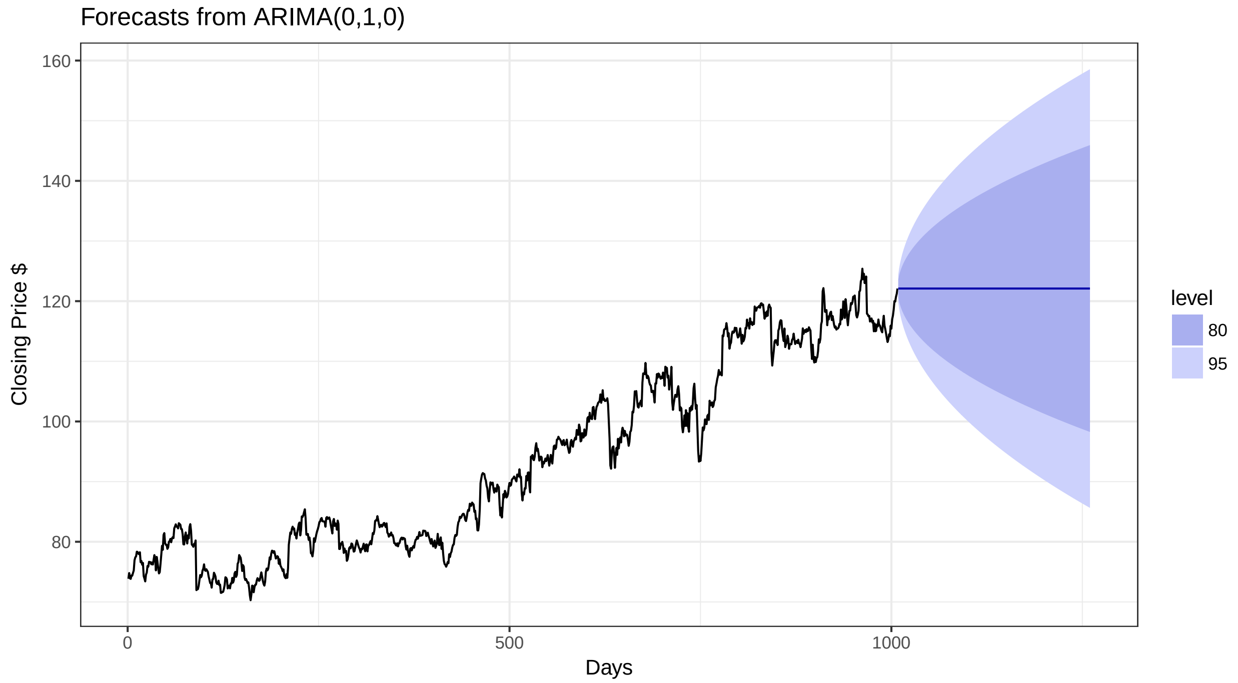
3. Model

We created four models using auto ETS, auto.arima, GARCH with ETS residuals, and GARCH model with auto.arima residuals.

The figure below shows the auto ETS(M,N,N) model. We notice that the forecast it pretty flat and just around $123.



The figure below shows the auto ARIMA(0,1,0) model. We notice that the forecast is similar to ETS model and is flat as well. Forecasted amount is around $123 as well.



We then create two GARCH models. One of the models uses ETS residuals and the other model uses ARIMA residuals. AIC values from all four models are shown in the table below. The smallest AIC here is the GARCH with ETS. However, the AIC value for this model is in negative. Not sure if picking a model with negative metric value is appropriate here. Further investigation will be needed. Therefore, we will ignore that for now and pick the next best model. Our next best model with the smallest AIC value is the GARCH with ARIMA.

Table 1: AIC Values

	ETS Model	ARIMA Model	GARCH ETS Model	GARCH ARIMA Model
AIC	7232.429	3179.555	-5.963627	3.12271

4. Metrics

The table below shows the various metrics from each model ran on out of sample data. The RMSE value is pretty close for all the models. ETS and ARIMA models have the same values for all metrics. GARCH ARIMA model that we selected earlier based on AIC values proves to be the correct choice because it beats the rest of the model in all the metrics.

Table 2: Metrics from each model

	ME	RMSE	MAE	MPE	MAPE
ETS Model - Test set	12.844	18.652	13.722	8.645	9.389
ARIMA Model - Test set	12.844	18.652	13.722	8.645	9.389
GARCH ETS - Test set	12.843	18.651	13.721	8.644	9.389
GARCH ARIMA - Test set	12.793	18.617	13.687	8.607	9.365

5. Conclusion

In conclusion, we would select GARCH(1,1) with ARIMA(0,1,0) model as our final model because it got beats the other models by performing better in all the metrics.

RMarkdown code is available on my GitHub page (https://github.com/Singh-Gurjeet) in the Time Series and Forecasting repository.

6. Appendix I: R Code

```r
knitr::opts_chunk$set(echo = TRUE, warning = F, message = F)

library(tidyverse)
library(fpp)
library(quantmod) # use for gathering and charting Stock data
library(fGarch)
options(scipen = 999)

#Download the Accenture data from Google Finance.
K <- getSymbols("ACN", src="google", return.class = "xts", warnings = F)
#Extract only closing price of the stock.
ACN_Close <- ACN$ACN.Close # Get the closing rate
#Extract only closing price of the stock.
#filter data for 5 years only
ACN_Close_5yr <- ACN_Close["2013-02-21/"]
ACN_train <- ACN_Close_5yr["2013-02-21/2017-02-21"]
ACN_test <- ACN_Close_5yr["2017-02-22/"]
```

```r
p1 <- autoplot(ACN_Close_5yr) + ggtitle("Accenture Stock Closing Price") +
        xlab("Year")+ ylab("Closing Price $") + theme_bw()
p2 <-autoplot(ACN_train) + ggtitle("Accenture Stock Closing Price - Train") +
        xlab("Year")+ ylab("Closing Price $") + theme_bw()
p3 <-autoplot(ACN_test) + ggtitle("Accenture Stock Closing Price - Test") +
        xlab("Year")+ ylab("Closing Price $") + theme_bw()
gridExtra::grid.arrange(p1,p2,p3, layout_matrix = rbind(1,2,3))

#Auto ETS Model
fit_ets <- ets(ACN_train)
fcast_ets <- forecast(fit_ets, h=nrow(ACN_test))
acc.ETS <- accuracy(fcast_ets,ACN_test)

aic_ETS <- data.frame(AIC(fit_ets))
rownames(aic_ETS) <- "AIC"
colnames(aic_ETS) <- "ETS Model"
#aic_ETS

autoplot(fcast_ets) +
  xlab("Days") +
  ylab("Closing Price $") +
  theme_bw()

#Auto ARIMA Model
fit_arima <- auto.arima(ACN_train)
#summary(fit_arima)
fcast_arima <- forecast(fit_arima, h=nrow(ACN_test))
acc.arima <- accuracy(fcast_arima,ACN_test)


aic_ARIMA <- data.frame(AIC(fit_arima))
rownames(aic_ARIMA) <- "AIC"
colnames(aic_ARIMA) <- "ARIMA Model"
#aic_ARIMA

autoplot(fcast_arima) +
  xlab("Days") +
  ylab("Closing Price $") +
  theme_bw()

#GARCH Model with ETS Residual
garch_ETS <- garchFit(formula = ~garch(1,1), data = fit_ets$residuals, trace = F)
#summary(garch_ETS)
garch_pred_ets <- predict(garch_ETS,n.ahead=nrow(ACN_test))
fcast_garch_ets <- fcast_ets$mean + garch_pred_ets$meanForecast
acc_garch_ets <- accuracy(fcast_garch_ets, ACN_test)

aic_gETS <- data.frame((garch_ETS@fit)$ics[1])
colnames(aic_gETS) <- "GARCH ETS Model"
#aic_gETS

#GARCH Model with ARIMA Residual
garch_arima <- garchFit(formula = ~garch(1,1), data = fit_arima$residuals, trace = F)
```

```r
#summary(garch_arima)
garch_pred_arima <- predict(garch_arima,n.ahead=nrow(ACN_test))
fcast_garch_arima <- fcast_arima$mean + garch_pred_arima$meanForecast
acc_garch_arima <- accuracy(fcast_garch_arima, ACN_test)

aic_gARIMA <- data.frame((garch_arima@fit)$ics[1])
colnames(aic_gARIMA) <- "GARCH ARIMA Model"
#aic_gARIMA

knitr::kable(cbind(aic_ETS, aic_ARIMA, aic_gETS, aic_gARIMA), caption = "AIC Values")

m_ets <- t(data.frame(acc.ETS[2,1:5]))
rownames(m_ets) <- "ETS Model - Test set"

m_arima<- t(data.frame(acc.arima[2,1:5]))
rownames(m_arima) <- "ARIMA Model - Test set"

rownames(acc_garch_ets) <- "GARCH ETS - Test set"
rownames(acc_garch_arima) <- "GARCH ARIMA - Test set"

knitr::kable(round(rbind(m_ets,m_arima,
                   acc_garch_ets, acc_garch_arima),3),
                  caption = "Metrics from each model")
```