MICHIGAN ROSS

How Avocado Forecasting Supports Walmart's Purchasing Decisions

TO 572 – Applied Business Forecasting

Final Project

Team #: 5

Written by Team 5:

Yuchen Gu

Quanfeng Li

Wenjie Wu

Tingting Xie

University of Michigan

Date: Feb 24th, 2021

1. Introduction

In recent years, as more and more people have raised concerns in environment, ecology, and health issues, they begin to go on vegan diets and restrain themselves from consuming animal protein as well as animal products. Avocado has turned out to become one of the substitutes of animal protein/products, since it can provide human beings with protein, energy as well as nutrition. According to the data, avocado consumption in the United States has increased by about 73% from 2000 (543 million pounds) to 2015 (938 million pounds).

Walmart, as one of the largest retail stores in the United States, provides its customers with avocado on a constant basis. The demand of avocado might be influenced by avocado's seasonality, popularity, and other external factors. To optimize Walmart's supply chain and reduce the cost of inventory, accurately forecasting the total volumes of avocados with models can be beneficial.

In this project, we focused on forecasting the demand for organic avocado in the Detroit region, so as to help local Walmart to make purchasing decisions on avocado based on the recent trend of avocado consumption. As an external consulting company, we don't have access to Walmart's historical consumption data of avocados during the current project pitch phase. However, we do have weekly consumption of avocado for all retailers in the Detroit region, so we used the following formula as an assumption to represent Walmart's avocado consumption.

$$\begin{aligned} &Organic\ Avocados\ Consumption\ for\ all\ Walmart\ Stores\ in\ Detroit \\ &\qquad = Total\ Organic\ Avocados\ Consumption\ in\ Detroit \\ &\qquad \times Market\ Share\ of\ Walmart\ in\ Detroit \end{aligned}$$

Specifically, we are looking into this problem from both strategic and operational aspects. From a strategic perspective, Walmart's purchasing department needs to know whether avocado's popularity keeps increasing in recent years despite the increasing popularity from 2000 to 2015, so as to adjust the supplier base and negotiate contracts in the long term. Besides, it needs to know whether the demand for avocados follows a certain seasonal pattern, so that they can communicate with suppliers and request flexibility in contracts for seasonal fluctuations. From the operational perspective, getting insights into the demand of avocados for the next month (four weeks) in advance would be helpful for satisfying the consumers' demand with less costs incurred.

2. Data

We used the dataset "Historical data on avocado prices and sales volume in multiple US markets" from Kaggle. It is a dataset with 18.2K rows and 14 columns, including avocado related information from 01/03/2015 to 03/24/2018 on a weekly basis (see *Table 1*). There is no missing data in this dataset.

Variable Name	**Description**	**Filtering Logic**
Date	the date of the observation	Select all
AveragePrice	the average price of a single avocado	Limited to "Organic"
type	conventional or organic	Limited to "Organic"
year	the year	Not use
Region	the city or region of the observation	Limit to "Detroit"
Total Volume	total number of avocados sold	Select all
4046	total number of avocados with PLU 4046 sold	Not use
4225	total number of avocados with PLU 4225 sold	Not use
4770	total number of avocados with PLU 4770 sold	Not use
Total Bags	Number of all bags sold	Not use
Small Bags	Number of small bags sold	Limited to "Organic"
Large Bags	Number of Large bags sold	Limited to "Organic"
XLarge Bags	Number of XLarge bags sold	Limited to "Organic"

Table 1: Description of variables in the data set

Specifically, the “Date” represents the date of observation, the “AveragePrice” reflects the average price of a single avocado in US Dollars, and the “Total Volume” is the total number of avocados sold. “4046”, “4225”, and “4770” are the product lookup codes which describe the size of an avocado. 4046 is small or medium-sized, 4225 is the large-sized one, and 4770 is the extra-large-sized. The values under these columns are the weights of each categorical avocado sale in pound units. “Total Bags”, “Small Bags”, “Large Bags”, and “XLarge Bags” are the measurement units for the avocados which were sold in pre-packaged containers or bags. The “type” variable is a binary variable which includes conventional avocado and organic avocado. Finally, the “Region” includes the main cities in each state of the US.

In this project, we placed our focus on Detroit and only extracted the data in Detroit to forecast Detroit’s exact demand of avocado in the next month.

3. Overview of the Approach

First, we transformed the dataset into a time series dataset using weekly intervals and visualized it to see if there are any patterns. Then we decomposed the time series data to do some initial trend analysis on the whole dataset and confirmed our guesses on certain patterns.

Next, we split the data into the training data and the testing data to try multiple forecasting methods, including naive, average, drift, SMA, SES, ETS, ARMA, ARIMA, and SARIMA. To further increase the accuracy of forecasting, we attempted to apply more advanced forecasting models, for example, dynamic linear regression, standard linear regression, dynamic harmonic regression, neural network, average combination of different models, and MLpol (Polynomial Potential Aggregation) and ODG (Online Gradient Descent) combination methods.

Finally, we evaluated all the forecasting methods and picked out the one with the highest accuracy to do the forecasting of future avocado consumption.

4. Analyses and Findings

4.1 Data Decomposition and Outlier Identification

After data visualization and decomposition (see *Figure 1*), we identified that there may be some outliers in the data set. In the total volume section of the Detroit area, there were several obvious outliers in March 2017 and March 2018, but such a pattern was not spotted in 2015 and

2016. The abnormal situation could not be explained even by searching for the special events hosted in Detroit in March 2017 and 2018. More importantly, March 2018 was involved in the test data, which would decrease the accuracy of the training model tests significantly because of the sudden increase in the total volume of avocado. Thus, we decided to adopt the tsclean method to adjust the value of outliers to get a more accurate forecasting model.

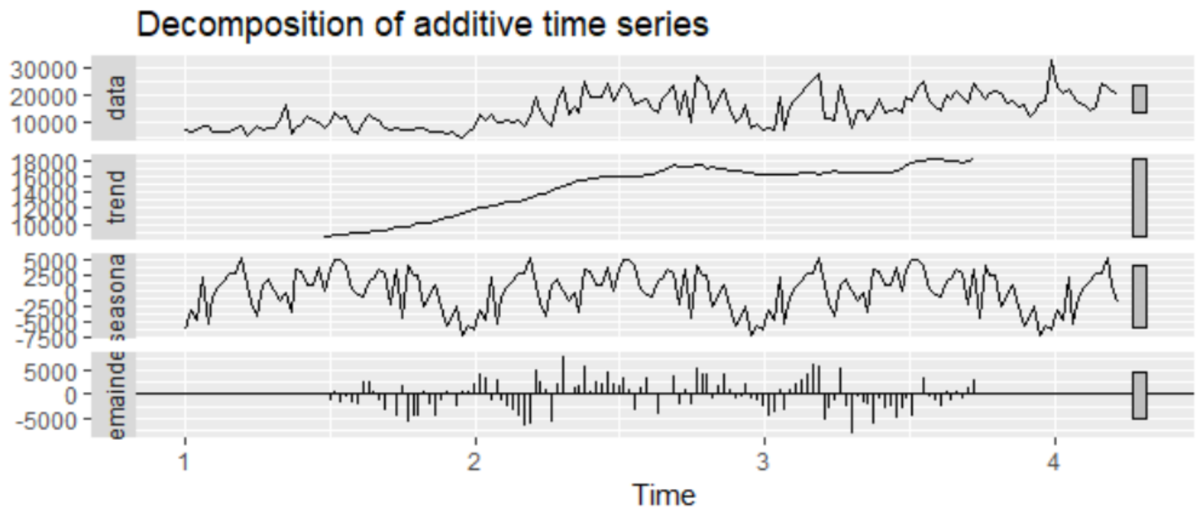

Figure 1: Result of decomposition by using the additive method

4.2 Models

Considering the leap year involved in the data set, the frequency was set as $365.27/7$. The plots of total volume (see *Figure 2*) and average price (see *Figure 3*) of Detroit show a big difference in terms of seasonality and trend. Total volume obviously keeps a clear seasonality and the increasing trend over time. Moreover, the average price is similar and stable across different cities (see *Figure 4*), which means there are no geographical differences and hard to get effective findings. The decomposition of the time series data also proved the above inference about total volume. For the 168 weeks in observation, the first 80% (135 weeks) were set to be the training data, and the latter 20% (33 weeks) were set to be the test data.

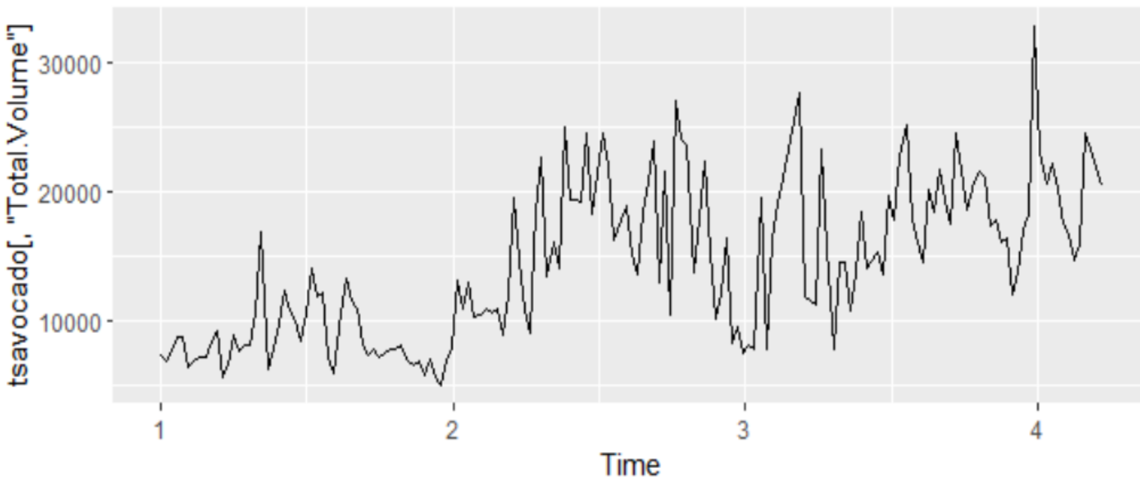

Figure 2: Trend of changing in total volume over time

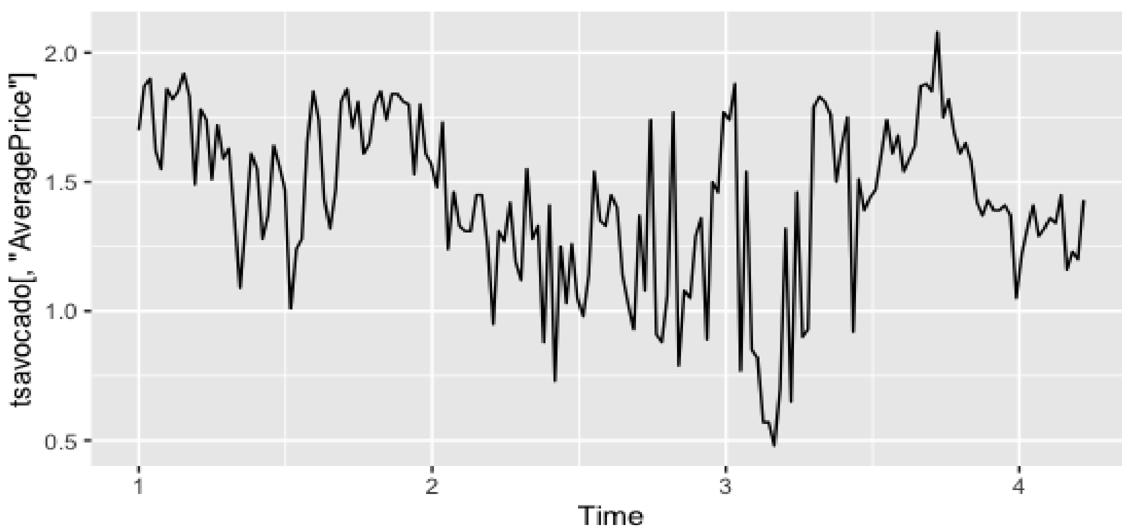

Figure 3: Trend of changing in average price over time

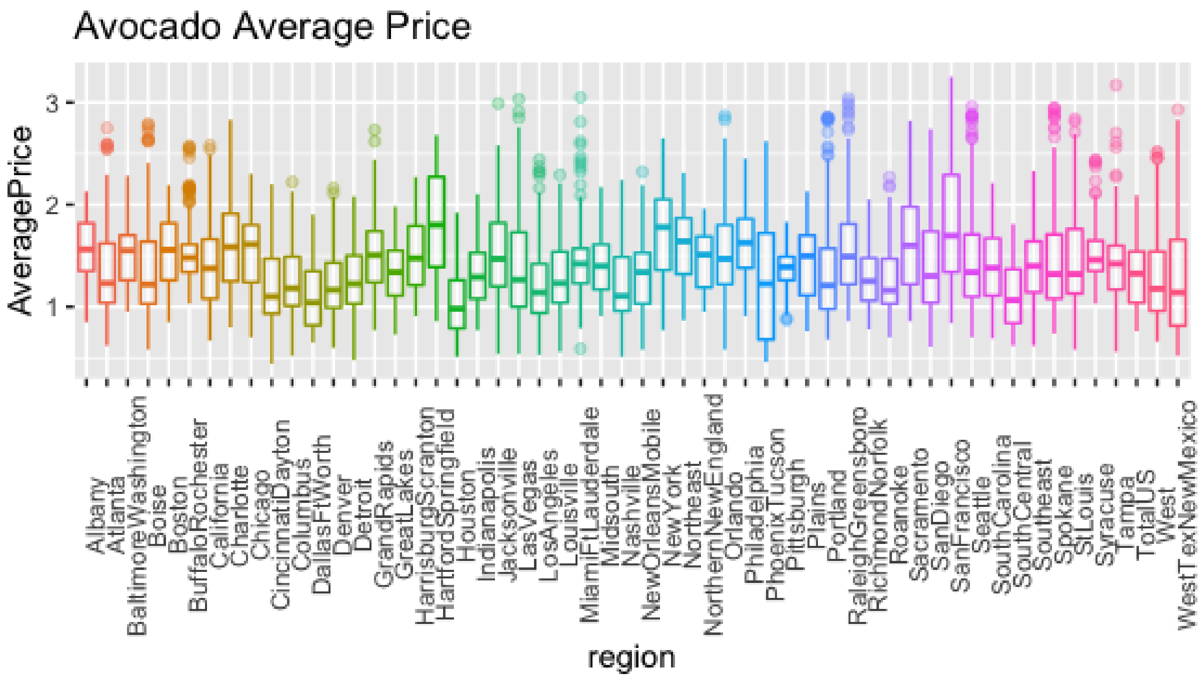

Figure 4: Average price in different cities

After running several forecasting models, the following tables and line graphs were generated by R. It was clear that SARIMA and ETS show better seasonality than other 7 methods (see *Figure 5)*. And ETS has the least MAPE whereas SARIMA has the highest MAPE among all (see *Table 2 and Figure 6*). For now, ETS gets the best performance with the nice seasonality and relatively lower forecast errors.

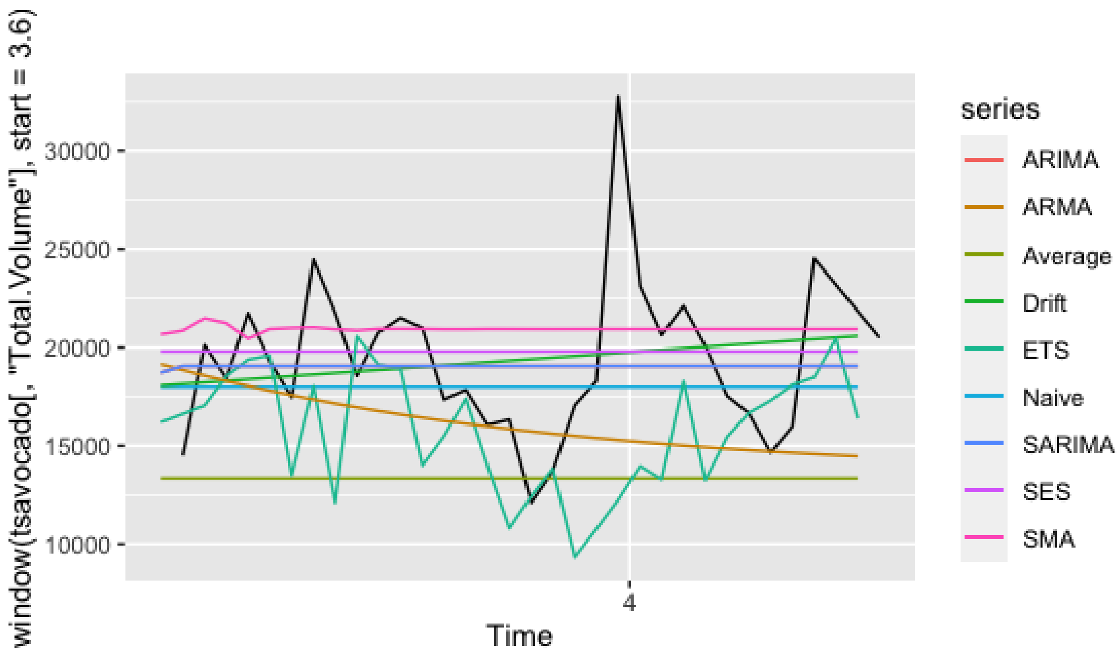

Figure 5: Forecast by using different models

Model Name	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
linear model	-6793.018	8244.797	7094.992	-36.284	37.52	0.601	1.876
Standard linear regression	-5543.705	7365.320	5949.101	-32.279	33.71	0.266	1.865
Average	6199.104	7329.229	6269.060	29.266	29.84	0.373	1.647
Dynamic harmonic regression	4754.574	6512.891	5050.965	21.878	23.90	0.478	1.472
ETS	3431.868	5736.168	4155.278	15.156	19.80	0.353	1.310
ARMA	3592.789	5563.992	4171.563	15.288	19.60	0.421	1.231
SMA	-1529.611	4202.708	3287.616	-12.073	18.70	0.376	1.069
comb2	2763.127	4921.590	3638.736	11.258	17.43	0.392	1.125
ARIMA	-645.997	3963.082	3062.891	-7.348	16.86	0.373	0.993
SARIMA	-645.997	3963.082	3062.891	-7.348	16.86	0.373	0.993
SES	-620.175	3958.955	3061.277	-7.210	16.83	0.373	0.991
Neural Network	102.937	4275.599	3043.781	-3.261	16.57	0.302	1.153
comb4	1891.263	4431.900	3260.524	6.521	16.14	0.353	1.049
Drift	53.703	3908.758	3006.136	-3.566	16.08	0.389	0.984
response2	2699.860	4768.847	3433.122	10.872	16.06	0.393	1.101
comb3	1289.156	4222.305	3084.185	3.302	15.74	0.325	1.037
Naïve	1430.905	4163.676	3094.274	3.761	15.32	0.373	0.982
response1	1787.031	4243.993	3145.410	6.161	15.24	0.307	1.016
comb1	419.936	3743.100	2604.431	-1.171	13.51	0.286	0.898

Table 2: Accuracy evaluation for different models

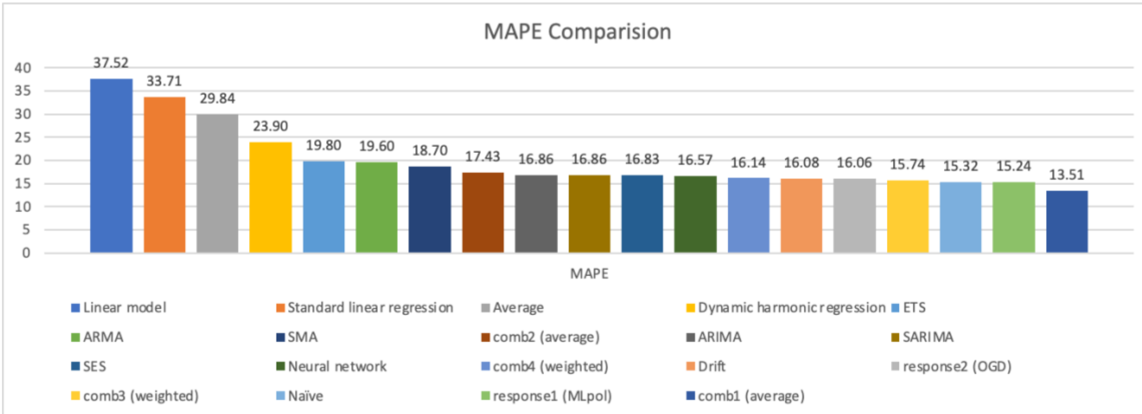

Figure 6: Visualization of MAPE

To find out the forecast models with higher fitting accuracy, the DLR, DHR, SLR and neural networks were further adopted when doing forecasts. The correlation plot helps in identifying the highly related external variables, which could be added into the forecast to increase the forecast accuracy. Obviously, average price, large bags and small bags have the better correlation with total volumes and will be considered in the linear regression (see *Figure 7)*.

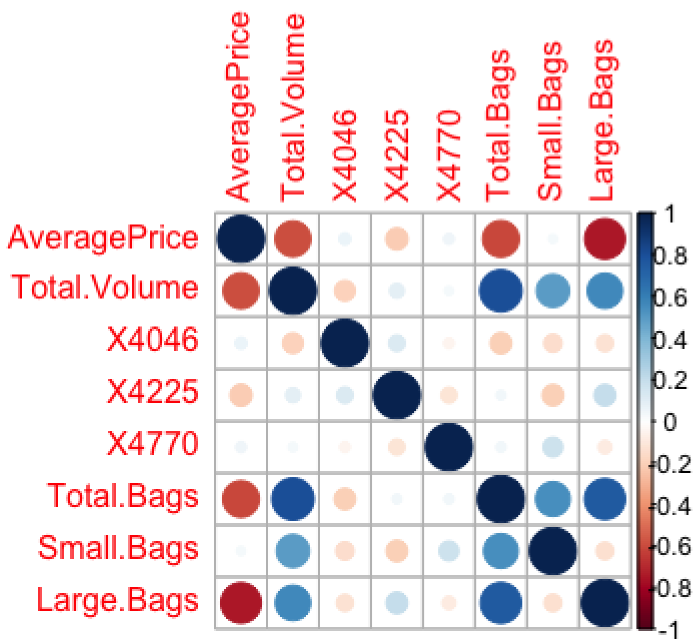

Figure 7: Correlation plot between total volumes and all external variables

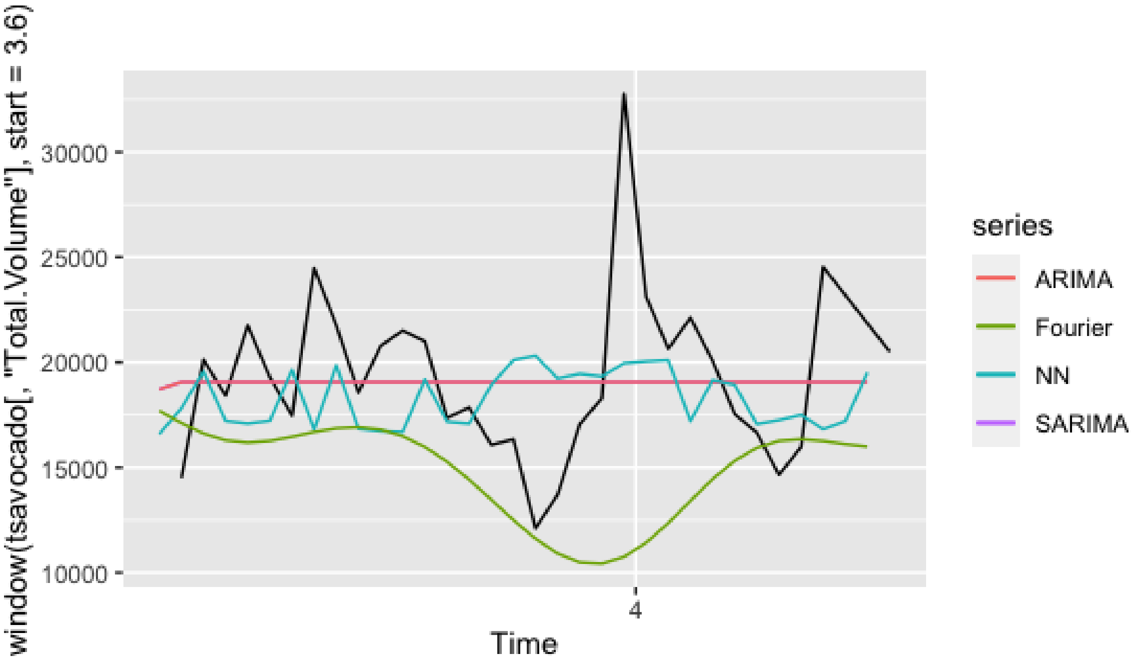

Figure 8: Visualization of neural network and Dynamic harmonic regression

Last but not least, we tried the forecast combination, which is a commonly used way for enhancing the forecast accuracy by combining several well performed forecast models together. Recalling the MAPE value of all the models, we chose ETS, DHR, ARIMA, Neural Network and DLR as combination 1, and ETS, Neural Network, and DHR as combination 2. Combination 3 and 4 used the weight calculated by MLpol and OGD methods (see *Figure 9).* Overall, the combination 1 gets the smallest MAPE (see *Figure 10 and Table 2)*.

```
> tail(weights1)
      ETS NN DHR
[28,]   0  1   0
[29,]   0  1   0
[30,]   0  1   0
[31,]   0  1   0
[32,]   0  1   0
[33,]   0  1   0
> tail(weights2)
            ETS        NN       DHR
[28,] 0.2986844 0.5118278 0.1894878
[29,] 0.2985845 0.5113869 0.1900286
[30,] 0.2984208 0.5112448 0.1903344
[31,] 0.3006629 0.5117681 0.1875691
[32,] 0.3037278 0.5108984 0.1853738
[33,] 0.3075391 0.5098171 0.1826438
```

Figure 9: Weights calculated by MLpol and OGD

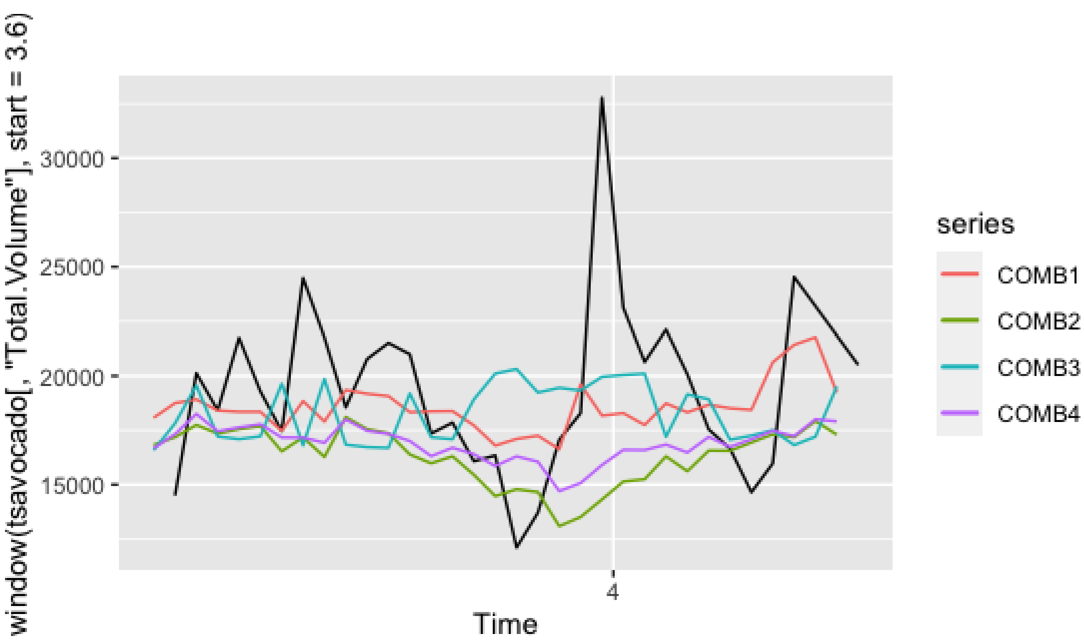

Figure 10: Visualization of different combinations

Since the MAPE value of the combination 1 is the smallest, we used combination 1 to forecast the total volume for the next 4 weeks. After running the model, the forecast of next 4 week's avocado total volume in the Detroit region would be :17539.04, 17464.18, 19003.78, 17633.91 (see *Figure 11)*.

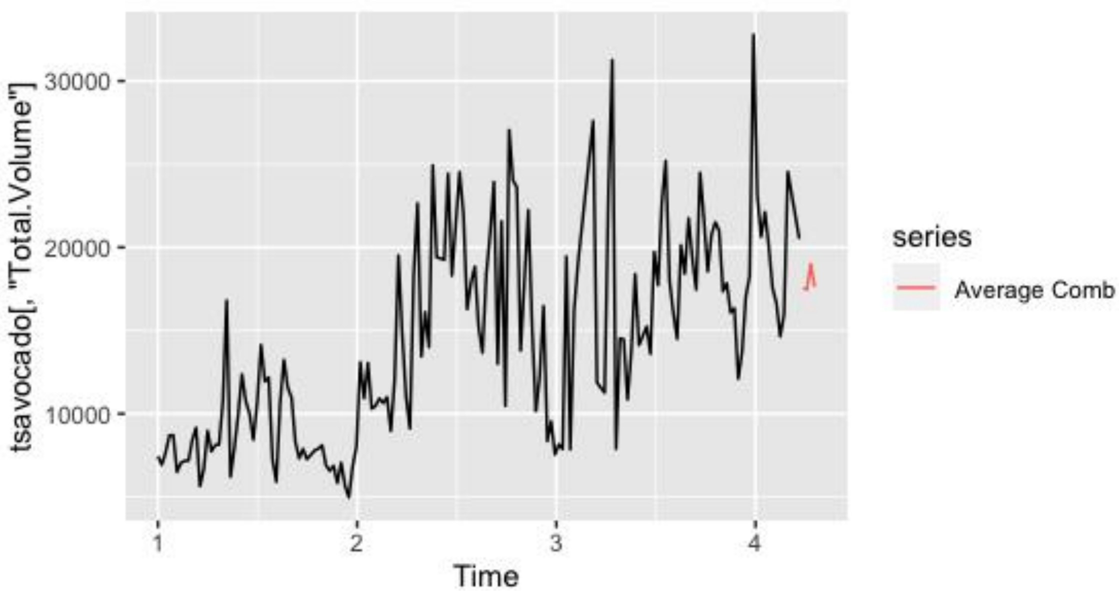

Figure 11: Visualization of forecasted total volume for the next four weeks

5. Recommendations

Walmart would use the forecast combination 1 model, which averages the ETS, DHR, ARIMA, Neural Network and DLR models with the smallest MAPE to forecast the consumption of avocados in three horizons: weekly consumption, monthly consumption, yearly consumption, and make short-term and long-term decision making accordingly.

5.1 Adjust Sourcing Strategy

According to next year's consumption forecast and trend decomposition, Walmart can see the long-term trend and overall fluctuations and adjust sourcing strategy accordingly. We can see the consumption of organic avocado in Detroit shows a dramatic increasing trend from 2015 to 2016. And the increasing trend becomes less obvious from 2017 to 2018.

Walmart should adjust the supplier base portfolio, including number of suppliers, level of interaction, etc. Since the volume of consumption of organic avocado is generally increasing in Detroit, we suggest Walmart consolidate avocado supply for the 6 Walmart stores in Detroit to maximize the benefits brought by economies of scale and reduce the complexity of upstream supply chain management.

Besides, Walmart should re-negotiate with suppliers on pricing, capacity, flexibility. Since the purchasing volume of avocado demand is increasing, Walmart, as a buyer, will have more buying power in contract negotiations to request a lower unit price. Also, since Walmart shares the supply of wholesalers with lots of competitors like Target or Wholefood in Detroit and the demand of organic avocado is still increasing, we expect there may be shortage risks of avocado in Detroit. So, we suggest Walmart request customer prioritization in supply capacity or shortage penalties in contract when the wholesaler is facing difficulty in meeting all its customers' demand. According to the seasonality and remainders decomposition plot, we could see there is high variability of organic avocado demand. Therefore, Walmart should negotiate flexibility of supply in contracts.

5.2 Improve Purchasing Operations

Based on weekly and monthly forecasts, Walmart stores in Detroit are able to make short-term purchasing decisions. For the monthly forecast we get from our forecasting model, Walmart stores could publish monthly rolling forecasts to the supplier, which will help the supplier with its sales, inventory and operations (SIOP) planning process cycle.

Our model is able to generate weekly forecasts. Walmart stores in Detroit areas may refer to the weekly forecast and place weekly orders of organic avocado to suppliers. Also, since avocado is a fresh product and spoils quickly, Walmart could divide the weekly order and allocate daily volume to each day. So, Walmart can request the supplier to deliver the volume allocated to each day daily to each store in Detroit.

6. Summary and Conclusion

In conclusion, we use the model "combination 1", and it achieves an MAPE of 13.51% to support Walmart's purchasing and sourcing decision-making.

6.1 Limitations

In our model, we only considered internal parameters (time and consumption) to forecast future consumption. However, several external factors may impact the consumption of Avocado in the Detroit region as well. First, the substitute's information like price may play a significant role. Since our forecasting model is focusing on the organic avocado, we identify regular avocado as the main substitute. Second, marketing campaign investment may impact the demand of avocado in Detroit. We suppose avocado's demand may show positive correlation with how much Walmart' investment in healthy diet and avocado nutrition. Third, the availability of avocado in Walmart stores may seriously affect customer's consumption. In retailing store scenarios, consumers will just leave when retailers don't have what they want. So, we may use service level (the probability of running out stock of organic avocados) to reflect the availability if we want to consider availability in forecasting.

The dataset only contains less than four years of data points about Avocado sales, which increases the difficulty of explaining the outliers and finding the patterns. Keeping the outliers in the dataset might influence the accuracy of forecast, on the other hand, adjusting the outliers would also negatively affect the accuracy of forecast in another way if they turned out to be meaningful data with more evidence supports.

6.2 Next Steps

Considering the limitations of our current model, our next steps are to add those affecting factors to our model and request more data on substitute, marketing, and availability from specific departments of Walmart. With those data on hand, we may conduct correlation and regression analysis to build a regression model. After that we could combine the regression model and our current model to add appropriate complexity and deliver better accuracy. At the same time, we need to pay attention to the overfit issues and omit non-significant factors.

Moreover, since we only have data for less than four years, we would request more historical data and keep updating the newest data. This will help us to better identify outliers and capture patterns more precisely.

Appendix

Code in R

```
# import libraries
install.packages("corrplot")

library(gridExtra)
library(smooth)
library(fpp2)
library(corrplot)

# read data
avocado = read.csv('avocado.csv')

####################################################################
################### Part 1: describe the data ######################
####################################################################

summary(avocado)
names(avocado)


# Plot Average Price.
ggplot(avocado) +
  geom_boxplot(aes(x=region, y=AveragePrice, color = region), alpha=0.3) +
  theme(axis.text.x = element_text(angle = 90))+
  ggtitle("Avocado Average Price") +
  theme(legend.position="none")

avocado_detroit_org = avocado[avocado$region == "Detroit" & avocado$type == "organic",]
```

```
#######################################################################
################### Part 2: decompose ###############################
#######################################################################

tsavocado = ts(avocado_detroit_org, freq=365.25/7)
tsavocado
autoplot(tsavocado[, 'AveragePrice'])
autoplot(tsavocado[, 'Total.Volume'])
add.decomp = decompose(tsavocado[, 'Total.Volume'], "additive")
autoplot(add.decomp)

avocado_detroit_org['Total.Volume'] = tsclean(avocado_detroit_org$Total.Volume)
tsavocado = ts(avocado_detroit_org, freq=365.25/7)

#######################################################################
################### Part 3: Models ##################################
#######################################################################

train_data = window(tsavocado, end = 3.6-1/(365.25/7))
test_data  = window(tsavocado, start = 3.6)

nrow(train_data)
nrow(test_data)

Volume = train_data[,"Total.Volume"]

volume.naive = naive(Volume, h=length(test_data[,"Total.Volume"]))
volume.ave   = meanf(Volume, h=length(test_data[,"Total.Volume"]))
volume.drift = rwf(Volume, drift=TRUE, h=length(test_data[,"Total.Volume"]))
volume.sma   = sma(Volume, h=length(test_data[,"Total.Volume"]))
volume.ses   = ses(Volume, h=length(test_data[,"Total.Volume"]))
```

```
volume.ets   = forecast(Volume, h=length(test_data[,"Total.Volume"]))

fit.arma = auto.arima(Volume, d=0, seasonal=FALSE)
volume.arma  = forecast(fit.arma, h=length(test_data[,"Total.Volume"]))

fit.arima = auto.arima(Volume, seasonal=FALSE)
volume.arima  = forecast(fit.arima, h=length(test_data[,"Total.Volume"]))

fit.sarima = auto.arima(Volume)
volume.sarima = forecast(fit.sarima, h=length(test_data[,"Total.Volume"]))

autoplot(tsavocado[,"Total.Volume"]) +
  autolayer(volume.naive$mean, series="Naive") +
  autolayer(volume.ave$mean, series="Average") +
  autolayer(volume.drift$mean, series="Drift") +
  autolayer(volume.sma$forecast, series="SMA") +
  autolayer(volume.ses$mean, series="SES") +
  autolayer(volume.ets$mean, series="ETS") +
  autolayer(volume.arma$mean, series="ARMA") +
  autolayer(volume.arima$mean, series="ARIMA") +
  autolayer(volume.sarima$mean, series="SARIMA")

# Zoom in
autoplot(window(tsavocado[,"Total.Volume"], start = 3.6)) +
  autolayer(volume.naive$mean, series="Naive") +
  autolayer(volume.ave$mean, series="Average") +
  autolayer(volume.drift$mean, series="Drift") +
  autolayer(volume.sma$forecast, series="SMA") +
  autolayer(volume.ses$mean, series="SES") +
  autolayer(volume.ets$mean, series="ETS") +
  autolayer(volume.arma$mean, series="ARMA") +
```

```
  autolayer(volume.arima$mean, series="ARIMA") +
  autolayer(volume.sarima$mean, series="SARIMA")


accuracy(volume.naive$mean, test_data[,"Total.Volume"])
accuracy(volume.ave$mean, test_data[,"Total.Volume"])
accuracy(volume.drift$mean, test_data[,"Total.Volume"])
accuracy(volume.sma$forecast, test_data[,"Total.Volume"])
accuracy(volume.ses$mean, test_data[,"Total.Volume"])
accuracy(volume.ets$mean, test_data[,"Total.Volume"])
accuracy(volume.arma$mean, test_data[,"Total.Volume"])
accuracy(volume.arima$mean, test_data[,"Total.Volume"])
accuracy(volume.sarima$mean, test_data[,"Total.Volume"])


#########################################################################
################ Part 4: Dynamic linear regression #################
#########################################################################

corrMatrix = as.matrix(subset(avocado_detroit_org,
select=c('AveragePrice','Total.Volume','X4046','X4225','X4770','Total.Bags','Small.Bags','Large.Bags')))
corrplot(cor(corrMatrix), method="circle")

options(warn=-1)

Price = train_data[,"AveragePrice"]
SmallBag = train_data[,"Small.Bags"]
LargeBag = train_data[,"Large.Bags"]

extvar = cbind(Price, SmallBag, LargeBag)
```

```
extvar_test = cbind(test_data[,"AveragePrice"], test_data[,"Small.Bags"],
test_data[,"Large.Bags"])

dlr.fit = auto.arima(Volume, xreg = extvar)
summary(dlr.fit)
dlr.fc = forecast(dlr.fit, xreg = extvar_test)
accuracy(dlr.fc$mean, test_data[,"Total.Volume"])
checkresiduals(dlr.fit)

df = data.frame(Volume, Price, SmallBag, LargeBag)
lm.fit = lm(Volume ~ Price + SmallBag + LargeBag, data = df)
summary(lm.fit)
testData = data.frame(extvar_test)
lm.fc = predict(lm.fit, newdata = data.frame(Price=testData[,1], SmallBag=testData[,2],
LargeBag=testData[,3]))
accuracy(lm.fc, test_data[,"Total.Volume"])
checkresiduals(lm.fit$residuals)
dlr.fc$mean
lm.fc

autoplot(window(tsavocado[,"Total.Volume"], start = 3.6)) +
  autolayer(volume.ets$mean, series="ETS") +
  autolayer(volume.sarima$mean, series="SARIMA") +
  autolayer(dlr.fc$mean, series="DLR")


######################################################################
############### Part 5: Dynamic harmonic regression ################
######################################################################

fit  = list()
```

```
fc   = list()
aicc = list()
p    = list()

for (i in seq(6)) {
  fit[[i]] = auto.arima(Volume, xreg = fourier(Volume, K=i), seasonal=FALSE)
  fc[[i]] = forecast(fit[[i]], xreg = fourier(Volume, K=i, h = length(test_data[,"Total.Volume"])))
  aicc[[i]] = fit[[i]]$aicc

  p[[i]] = autoplot(fc[[i]]) +
    ggtitle(paste("K =",i))  +
    ylab("")
}

aicc

autoplot(ts(aicc)) # minimum is k = 5

grid.arrange(p[[1]],p[[2]],p[[3]],p[[4]],p[[5]],p[[6]], nrow = 3, ncol = 2)

fit[[5]]

# Since using K = 1 gives us the smallest AICc, we use this as our final model.
accuracy(fc[[5]]$mean, test_data[,"Total.Volume"])

autoplot(window(tsavocado[,"Total.Volume"], start = 3.6)) +
  autolayer(volume.sarima$mean, series="SARIMA") +
  autolayer(volume.arima$mean, series="ARIMA") +
  autolayer(dlr.fc$mean, series="DLR") +
  autolayer(fc[[5]]$mean, series="Fourier")
```

```
#####################################################################
#################### Part 6: Neural Network ########################
#####################################################################

fit_nn = nnetar(Volume)
fc.nn = forecast(fit_nn, h = length(test_data[,"Total.Volume"]))
accuracy(fc.nn$mean, test_data[,"Total.Volume"])

autoplot(window(tsavocado[,"Total.Volume"], start = 3.6)) +
  autolayer(volume.sarima$mean, series="SARIMA") +
  autolayer(volume.arima$mean, series="ARIMA") +
  autolayer(fc.nn$mean, series = "NN") +
  autolayer(fc[[5]]$mean, series="Fourier")

#####################################################################
################ Part 7: Forecast combination ######################
#####################################################################

# Forecast combination using averaging
fc.comb1 = (volume.arima$mean + volume.ets$mean + fc.nn$mean + fc[[5]]$mean +
dlr.fc$mean)/5
accuracy(fc.comb1, test_data[,"Total.Volume"])

fc.comb2 = (volume.ets$mean + fc.nn$mean + fc[[5]]$mean)/3
accuracy(fc.comb2, test_data[,"Total.Volume"])

#####################################################################
###### Part 8: More sophisticated forecast combinations ############
#####################################################################
```

```
install.packages("opera")
library(opera)

X = cbind(ETS = volume.ets$mean, NN = fc.nn$mean, DHR = fc[[5]]$mean)

mixt1 = mixture(model = "MLpol", loss.type = "square")
mixt2 = mixture(model = "OGD", loss.type = "square")

weights1 = predict(mixt1, X, test_data[,"Total.Volume"], type="weights")
weights2 = predict(mixt2, X, test_data[,"Total.Volume"], type="weights")

response1 = predict(mixt1, X, test_data[,"Total.Volume"], type="response")
response2 = predict(mixt2, X, test_data[,"Total.Volume"], type="response")

mixt1
mixt2

# the last row is the optimal weight
tail(weights1)
tail(weights2)

fc.comb3 = 0*volume.ets$mean + 1*fc.nn$mean + 0*fc[[5]]$mean
fc.comb4 = 0.3075391*volume.ets$mean + 0.5098171*fc.nn$mean + 0.1826438*fc[[5]]$mean

accuracy(volume.naive$mean, test_data[,"Total.Volume"])
accuracy(volume.ave$mean, test_data[,"Total.Volume"])
accuracy(volume.drift$mean, test_data[,"Total.Volume"])
accuracy(volume.sma$forecast, test_data[,"Total.Volume"])
accuracy(volume.ses$mean, test_data[,"Total.Volume"])
accuracy(volume.ets$mean, test_data[,"Total.Volume"])
accuracy(volume.arma$mean, test_data[,"Total.Volume"])
```

```r
accuracy(volume.arima$mean, test_data[,"Total.Volume"])
accuracy(volume.sarima$mean, test_data[,"Total.Volume"])
accuracy(dlr.fc$mean, test_data[,"Total.Volume"])
accuracy(lm.fc, test_data[,"Total.Volume"])
accuracy(fc[[5]]$mean, test_data[,"Total.Volume"])
accuracy(fc.nn$mean, test_data[,"Total.Volume"])

accuracy(fc.comb1, test_data[,"Total.Volume"])
accuracy(fc.comb2, test_data[,"Total.Volume"])
accuracy(as.numeric(response1), test_data[,"Total.Volume"])
accuracy(as.numeric(response2), test_data[,"Total.Volume"])
accuracy(fc.comb3, test_data[,"Total.Volume"])
accuracy(fc.comb4, test_data[,"Total.Volume"])


autoplot(window(tsavocado[,"Total.Volume"], start = 3.6)) +
  autolayer(fc.comb1, series="COMB1") +
  autolayer(fc.comb2, series="COMB2") +
  autolayer(fc.comb3, series="COMB3") +
  autolayer(fc.comb4, series="COMB4")

###################################################################
################## Part 9: forecast for future ####################
###################################################################

fc_Volume = tsavocado[,"Total.Volume"]

fc_volume.ets = forecast(fc_Volume, h=4)
fc_fit_nn = nnetar(fc_Volume)
fc_nn = forecast(fc_fit_nn, h = 4)
```

```
fc.fit.arima = auto.arima(fc_Volume, seasonal=FALSE)
fc.volume.arima  = forecast(fc.fit.arima, h=4)

fc_fit5 = auto.arima(fc_Volume, xreg = fourier(fc_Volume, K=5), seasonal=FALSE)
fc_fc5  = forecast(fc_fit5, xreg = fourier(fc_Volume, K=5, h = 4))

fc.Price = tsavocado[,"AveragePrice"]
fc.SmallBag = tsavocado[,"Small.Bags"]
fc.LargeBag = tsavocado[,"Large.Bags"]
fc.extvar = cbind(fc.Price, fc.SmallBag, fc.LargeBag)
fc.dlr.fit = auto.arima(fc_Volume, xreg = fc.extvar)
fc.dlr = forecast(fc.dlr.fit, xreg = fc.extvar)

fc_avg_comb =(fc.volume.arima$mean + fc_volume.ets$mean + fc_nn$mean + fc_fc5$mean +
fc.dlr$mean)/5

fc_avg_comb

autoplot(tsavocado[,"Total.Volume"]) +
  autolayer(fc_avg_comb, series = "Average Comb")
```