The Tale of Retail Sales

STATS 170

Takao Oba (5893), Anish Dulla (1557), Shoichiro Ueno (8290), Daniel Neufeldt (9414)

https://github.com/toba717/Time_Series_Analysis/tree/main/final_project

Our group searched for monthly or quarterly time series data that are not seasonally adjusted and have a minimum of 200 observations, either on FRED or Quandl databases. We selected 3 variables to help answer the overarching research question. We justified our choice and provided the source information in a table with codes, descriptions, time periods, and dependent or independent variables. We then accessed the time series data with the Quandl library in R and plotted them using the dygraph function. We split the dependent variable into a training set and a test set, and worked only with the training data for the dependent variable. We determined whether to do additive or multiplicative decomposition and implemented strategies if appropriate. Our group then did a seasonal box plot of the raw data and the autocorrelation features of the random term of the dependent variable. Next, we used our findings to fit an appropriate exponential smoothing model to our raw training dependent variable, and forecasted on the test set with the model. Lastly, we fit a polynomial regression model to the trend of our data and multiplied by the seasonal component to gain an additional model with forecast.

I. Introduction

The retail industry is a constantly evolving space, with consumer behavior and market trends playing a significant role in shaping the success of various retail segments. In this project, we will take a closer look at the performance of three important retail segments – 1. Electronics & Appliance Stores, 2. Sporting Goods, Hobby, Musical Instrument and Book Stores, and 3. Furniture and Home Furnishings Stores. Using data from the Federal Reserve Bank of St. Louis (FRED), we will explore the trends and patterns in Retail Sales for each of these segments, and gain valuable insights into the state of the retail industry as a whole.

FRED Code	Description	Time Period	Variables
RSEASN https://fred.stlouisfed.org/series/RSEASN	Retail sales of Electronics and Appliance Stores (monthly)	1992:01	Dependent
RSSGHBMSN https://fred.stlouisfed.org/series/RSSGHBMSN	Retail sales of Sporting Goods, Hobby, Musical Instrument, etc. (monthly)	1992:01	Independent
RSFHFSN https://fred.stlouisfed.org/series/RSFHFSN	Retail sales of Furniture and Home Furnishings Stores (monthly)	1992:01	Independent

Table 1.1: Metadata Description of Selected Datasets

Through analyzing the above datasets, we aim to generate a forecasting model that will best capture the image of the future cases of the time series dataset. After loading our variables, we examine relationships between them by using a Multiple Time Series Plot.

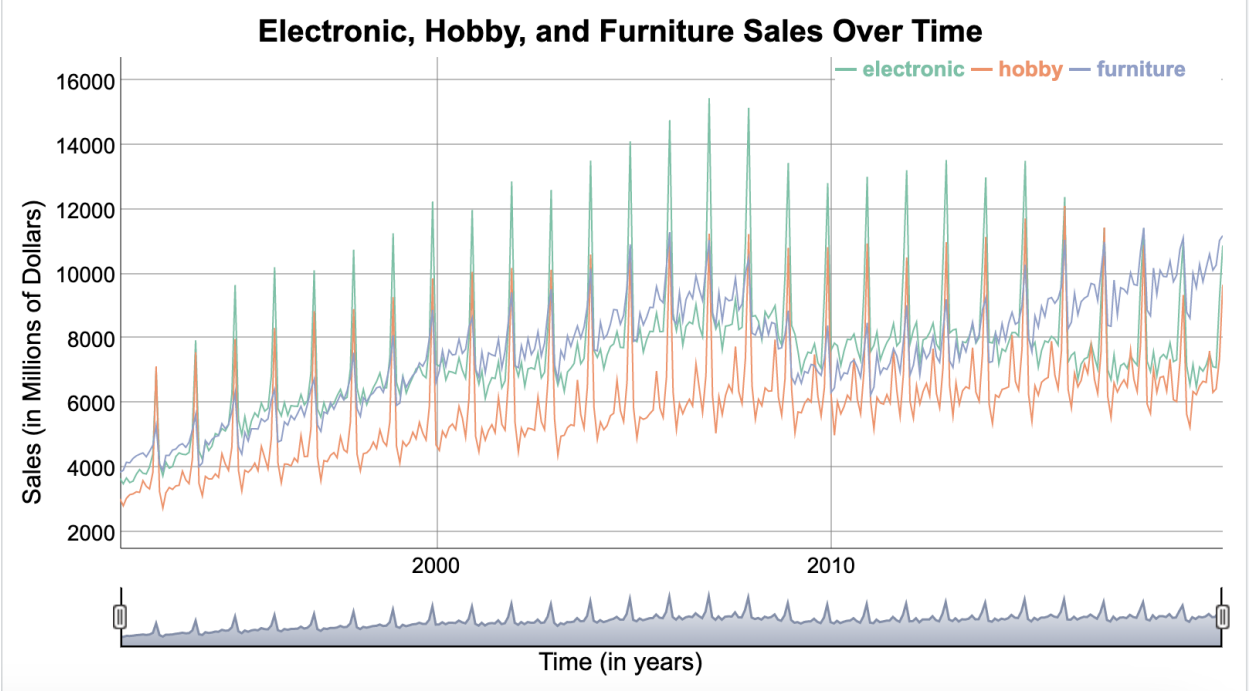

Fig 1.2: Multiple Time Series Plot (DyGraph) of Electronic, Hobby, and Furniture Sales

In Fig 1.2, we see that the electronic and furniture retail sales patterns remained relatively similar to each other in terms of having a slight upwards trend before 2008 with a sharp dip around 2008-2009, and then having another upwards trend afterwards. The hobby variable appears to have a different trend as it is more stable. Although the hobby variable does not dip around the recession era, the constant upwards trend was not as strong as when the other two variables were having their respective upwards trends. One key observation we can make is that there is very strong seasonality occurring during the end of the year holiday season for all three variables.

Ultimately, we choose retail sales of electronics to be our dependent variable since it displayed the most interesting changes over time, making a more insightful analysis. Additionally, electronic products are purchased quite frequently in the United States, so we felt that it would be a generally good indicator of spending habits in America. With our dependent variable, we choose to split our data into a training and testing set. In this process, we aimed to have training/testing sets equal sizes for all variables in case of future analysis. The beginning of the data for all three time-series was January 1992. In order to have the three variables be in the same window, we set the end to be December 2019 (This was to avoid the sudden change in our data due to Covid-19). We decided to set our training data to span over the course of January

1992 (start date) until December 2018 and set the testing data to be the last twelve months in our original time series (January 2019 to December 2019) for our retail electronics sales data.

II. Components Features of the dependent variable

Before model fitting and deeper analysis, it is critical to comprehend the fundamental composition of the data to render precise forecasts and draw pertinent conclusions. A critical approach to gaining insights into the structure of a time series is through decomposition, which involves breaking down the series into its individual components. There are two primary methods of decomposition: additive and multiplicative. In additive decomposition, the components are added together to estimate the time series and in contrast, in multiplicative decomposition, the components are multiplied together to estimate the time series.

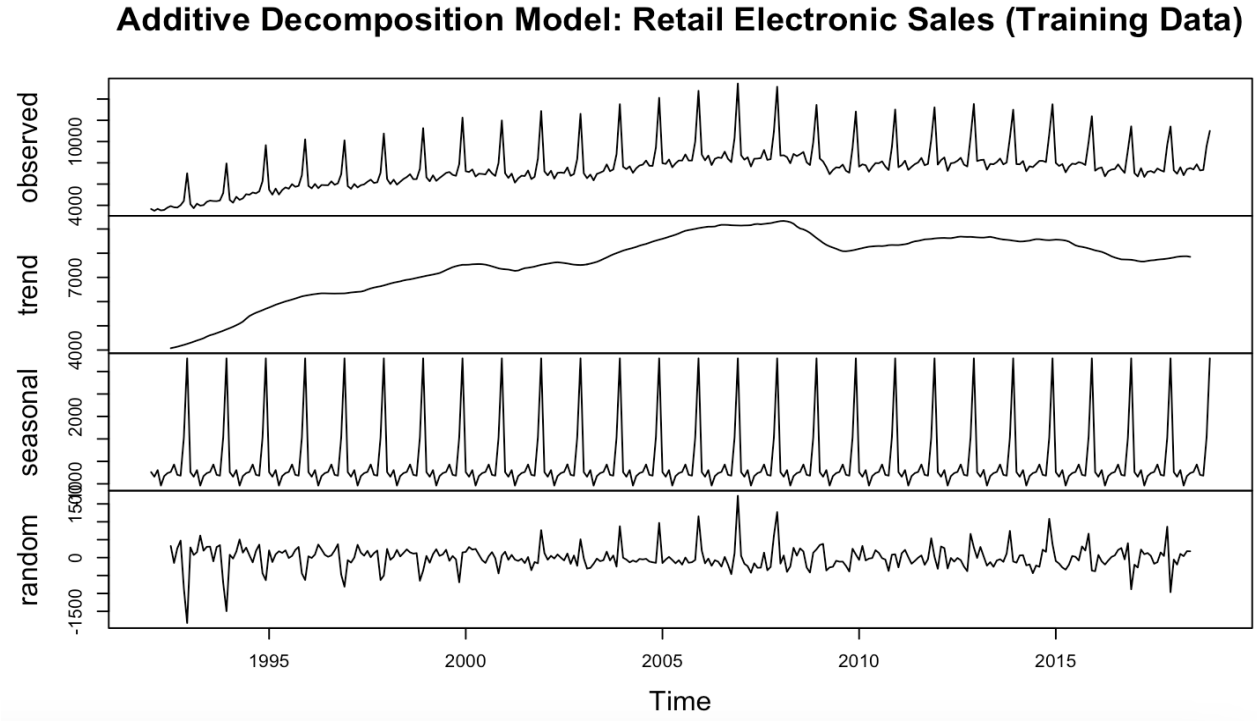

Fig 2.1: Additive Decomposition Model – Retail Electronic Sales (Training Dataset)

The additive decomposition is not appropriate to understand our time series because it fails one of its key assumptions. The seasonal effects are not constant over time. This is proven in Fig 2.1 where the plot of the random term shows a clear pattern and uneven seasonal effects. Therefore, we should look into alternative methods.

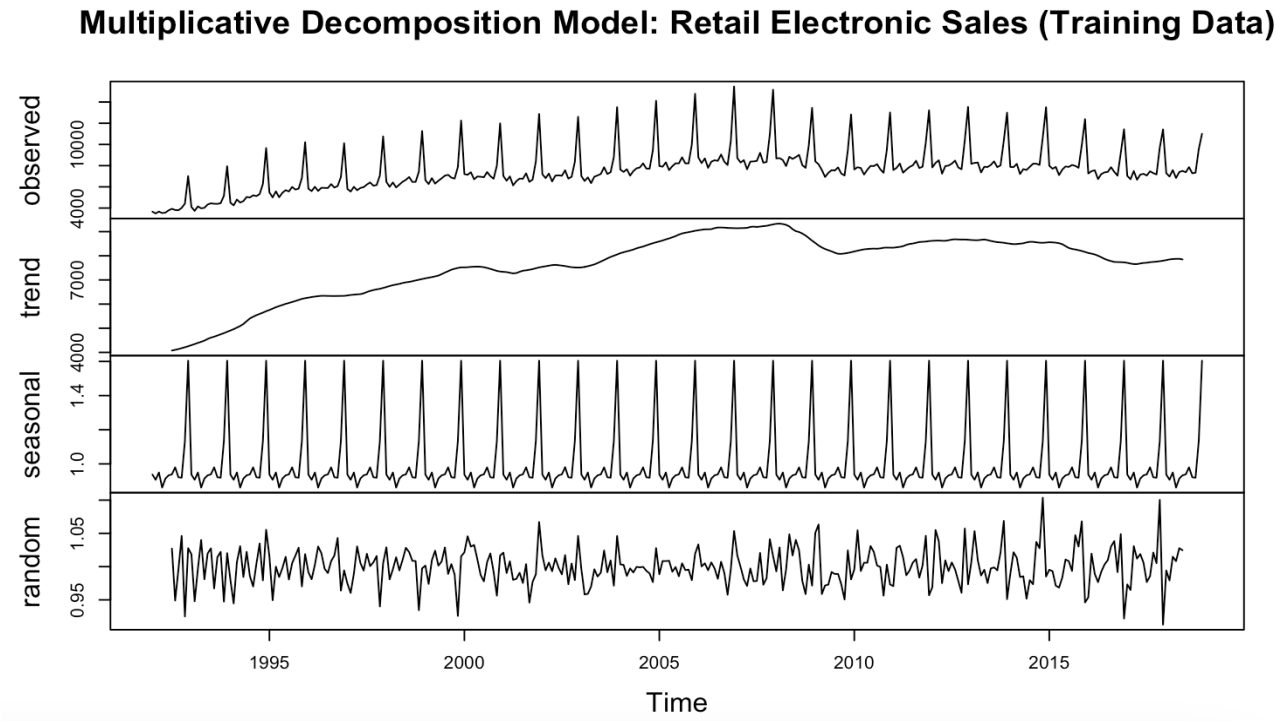

Fig 2.2: Multiplicative Decomposition Model – Retail Electronic Sales (Training Dataset)

On the other hand, multiplicative decomposition is a more appropriate model to understand our time series because we see that the random term plot seen in Fig 2.2 shows no clear pattern (indicates randomness) and thus more properly models the seasonal effects. Therefore, we can better understand relations of our time series using a multiplicative decomposition model.

The multiplicative decomposition shows that there is a clear increasing trend over time. Additionally, there is consistent seasonality with sharp spikes during certain months of the electronics yearly sale cycle. This is intuitive as the sales of electronics tend to increase in months or seasons where there are nation-wide sales or new product releases. Thus, we would expect sales to have a sharp increase around the holiday season every year and Autumn when market leaders such as Apple releases new series of iPhone products. For further analysis on seasonality, we can look at a seasonal boxplot of our data to discover insights.

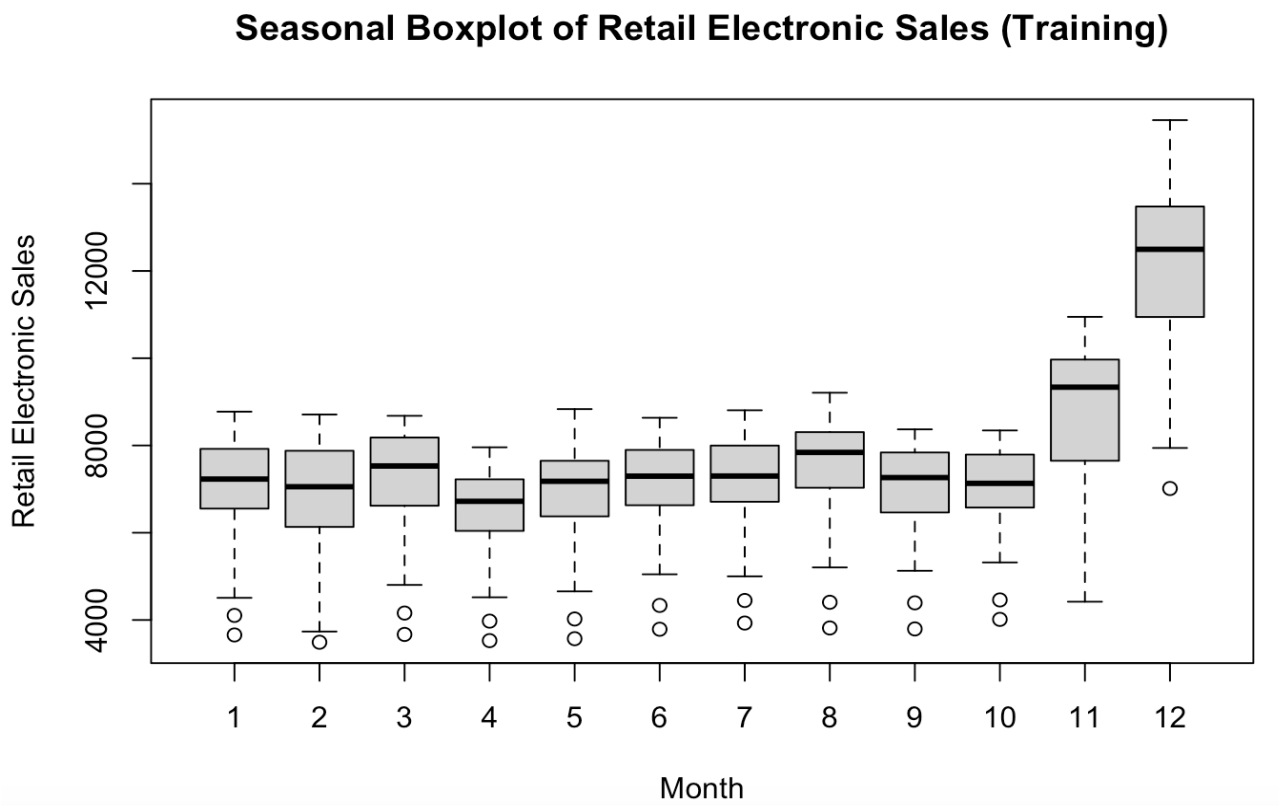

Fig 2.3: Seasonal Boxplot of Retail Electronics Sales (Train Data)

Note: In the x-axis, each number corresponds to each month (e.g., 1 = January, 2 = February, etc)

Looking at the seasonal boxplot, we can examine the distribution of the data within each season and discern any noticeable patterns or trends. Notice that in Fig 2.3, retail electronics sales spike during November and December (months 11 and 12) for the holiday season, indicating clear and recurring seasonality within our data. This further solidifies our seasonality analysis.

III. Autocorrelation features of the random term of the dependent variable in the training set

After exploring seasonality and trend in our data, we investigate autocorrelations in the time series. We look into the ACF and PACF of the random component of the retail electronic sales' multiplicative decomposition.

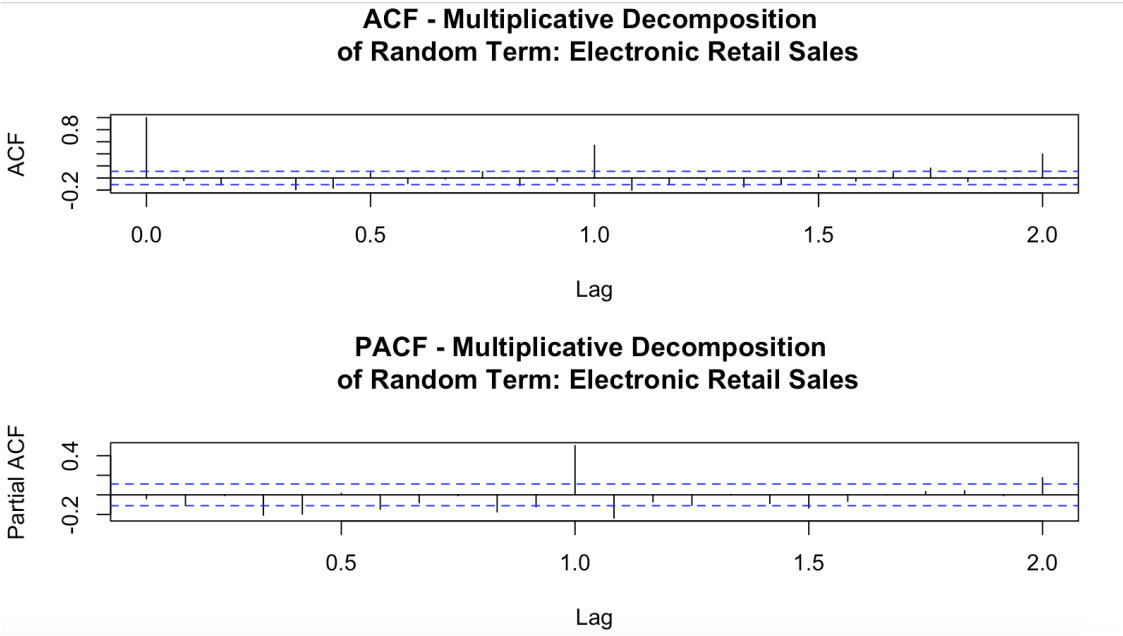

Fig 3.1: ACF and PACF of Random Term of Multiplicative Decomposition

In the ACF plot of the random term, we observe an oscillating wave pattern with negative and positive autocorrelations. Furthermore, we observe abnormally high autocorrelations at lag k = 1, 2 which signals strong seasonality in our data. Furthermore, there are also other significant autocorrelations present not at exact lags. However, this can be attributed to sampling error – roughly 5% of autocorrelations that are not significant will appear as significant. The ACF plot indicates that our data is not stationary, and there seems to be a deeper process for generating our time series data. Next, we examine the PACF plot. Similar to the ACF plot, there are high autocorrelations at lag k = 1, 2. Also, there are more significant autocorrelations than in the ACF.

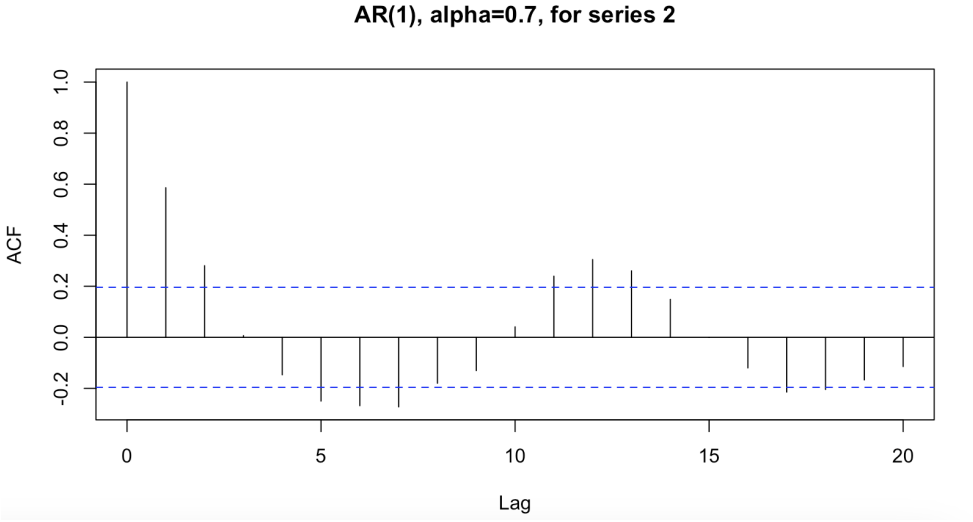

Fig 3.2: ACF of a Simulated AR(1) Process, alpha = 0.7

After observing the ACF of different simulated models, we concluded that our time series most closely resembled the AR(1) model for Series 2. This is because the ACF of the AR(1) model for Series 2 also has a wave-like pattern and many significant autocorrelations at the later lags. Looking at the PACF, while our plot doesn't perfectly match what is expected from an AR(1) model, it does show significant autocorrelations at lags and dies down later in the plot. Out of the simulated models given to us in the R script, this behavior most closely represents what is seen in the AR(1) model both in ACF and PACF, despite not being a perfect fit. Thus, we determined that our model was likely an AR(1) model; however, further analysis must be done to be certain.

IV. Exponential smoothing modeling and forecasting

Because the retail sale of electronic data has increasing seasonality and trend, we chose the seasonal exponential smoothing with multiplicative decomposition as our model.

Final Fitted Model: hw_electronic = HoltWinters(retails_electronic_train, seasonal = 'multiplicative')

First, we fit our Triple Exponential Smoothing model to our raw training electronic retail sales data. This model accounts for the training data set from the initial observation to the end of 2018.

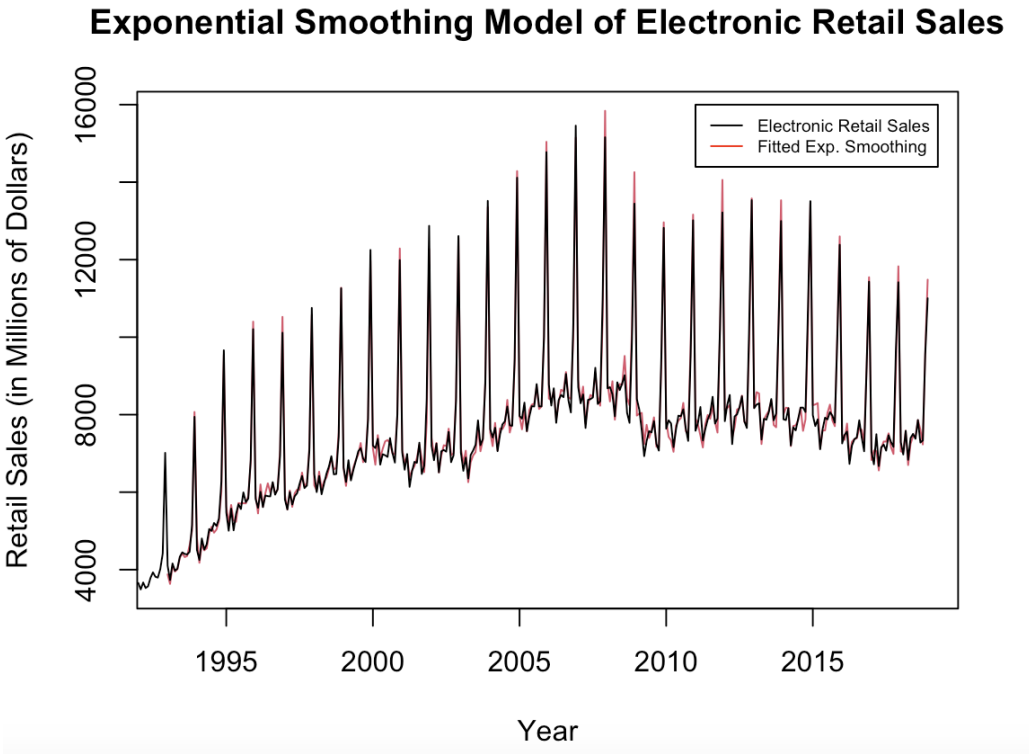

Fig 4.1: Multiple Time Series Plot of Raw Retail Electronic Sales and Fitted Holt Winters Exponential Smoothing Model (January 1992 and December 2018)

In our exponential smoothing model, the peaks are slightly overestimated from the original data; however, this is only by a small margin. Because the data looked like it was multiplicative due to the endpoints being significantly smaller than the middle points, we decided to make our model treat the seasonality as multiplicative. Ultimately, we can see that this fitted model follows the trend relatively well and accounts for the seasonality peaks. Next, we forecast with our model.

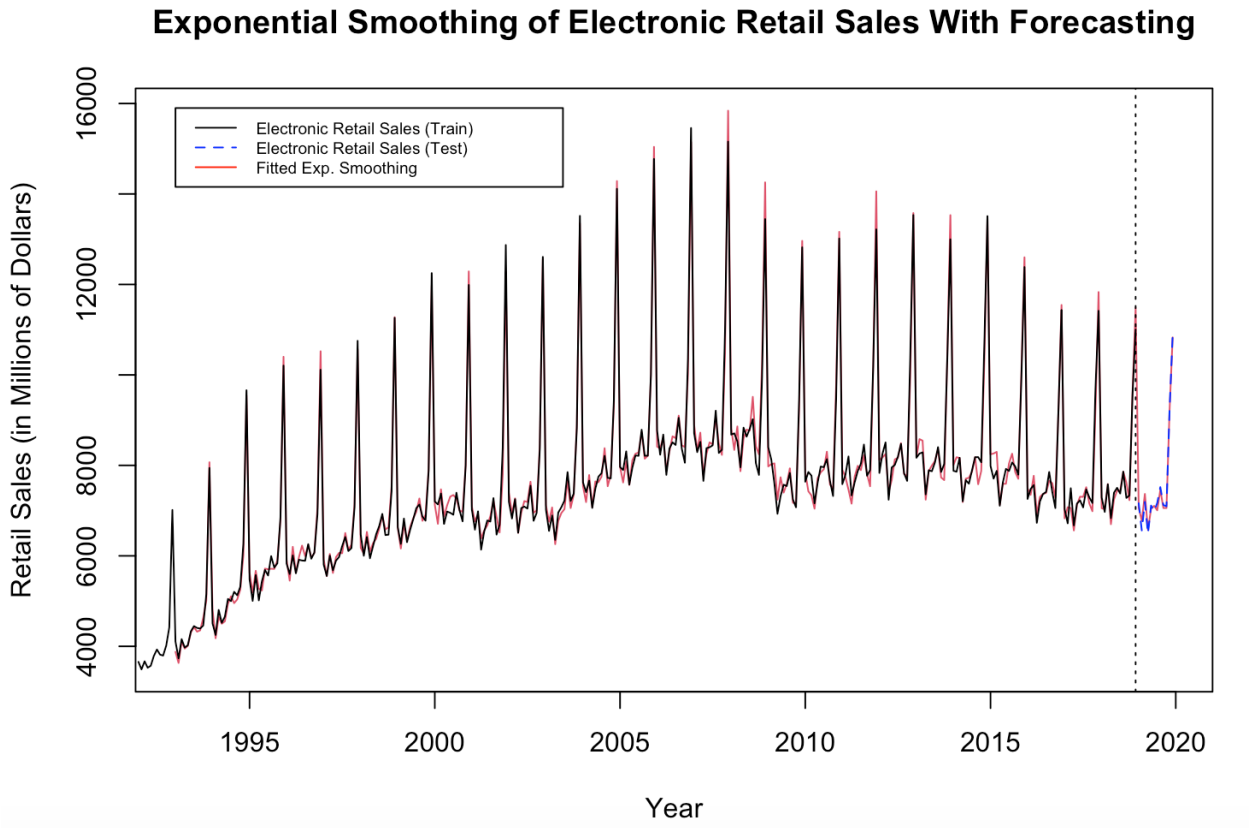

Fig 4.2: Multiple Time Series Plot of Raw Retail Electronic Sales and Fitted Forecast Exponential Smoothing Model With Forecast onto Testing Dataset

Here, we can see on the right side of the vertical dotted line, the test data and the predicted 2019 data are very close to each other. Because of this, we can determine that this model does a very good job in predicting the data; however, there could be some overfitting due to how well this model fits the training data. We must examine a close up of the forecast and residuals.

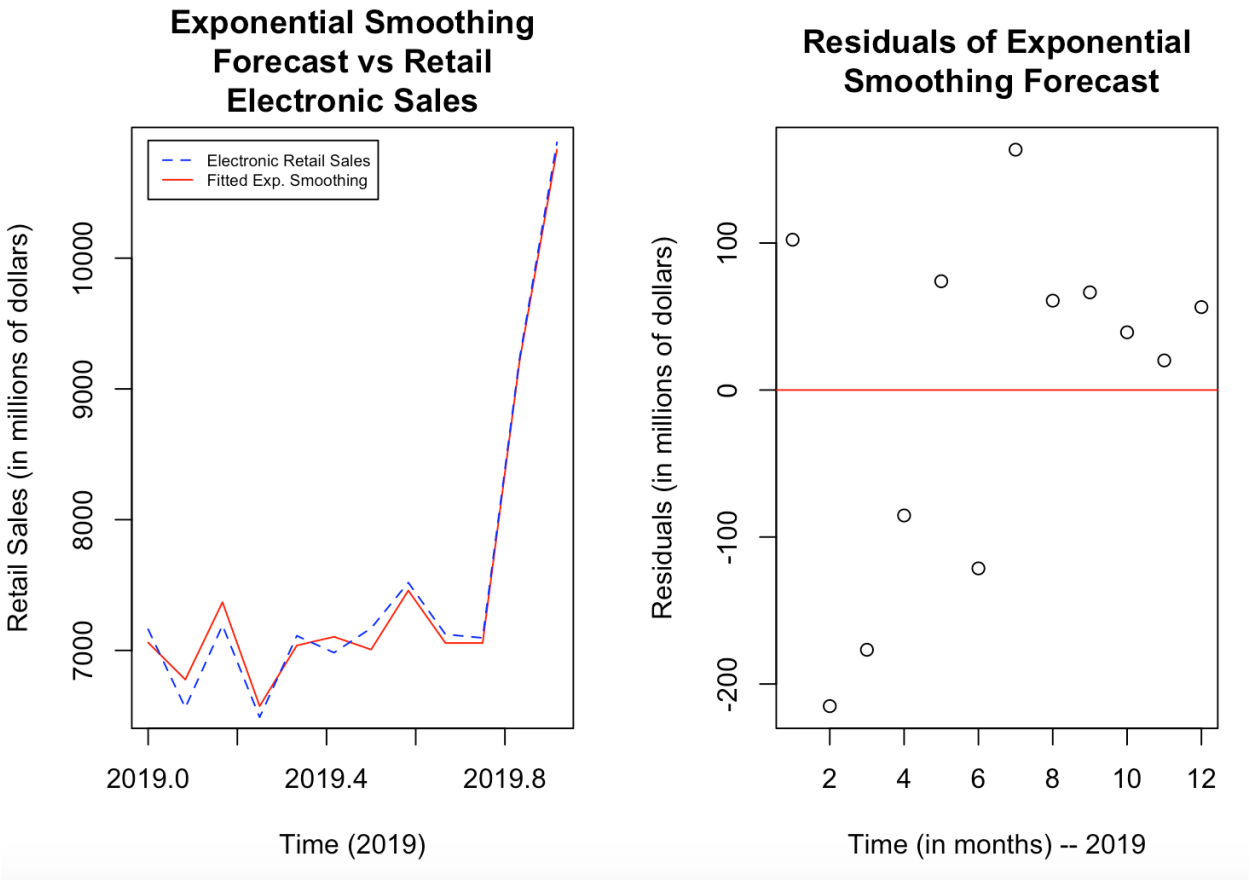

Fig 4.3: Multiple Time Series Plot of Raw Retail Electronic Sales and Forecast of Exponential Smoothing Model 2019 (depicted in left) Residuals of Forecasted Model (depicted in right)

Using our fitted model, we then forecasted the next 12 observations (the year of 2019), the same size as our testing dataset. To assess the validity of our model, we compare our forecasts to the testing dataset – conducting analysis on residuals. When we see the plot of our actual values versus the predicted values as well as our residual plot, we can see that the residual values are not very large. Furthermore, there is no clear pattern and the residuals appear random (Fig 4.3). Because of the plots available, we can see that this is a good prediction for our testing dataset.

V. Polynomial regression plus seasonal effect modeling and forecasting

In order to explore other methods of forecasting, we decided to examine polynomial regression. To begin, we fit a variety of models, including linear, quadratic, and cubic regression models. The linear model: $y = mx + b$ failed to properly fit the data since the trend of our data does not follow a linear pattern. On the other hand, the cubic model: $y = ax^3 + bx^2 + cx + d$

introduces multicollinearity and overcomplicates our model. This either leads to overfitting in the data and capturing noise as part of our time series' trend or zeroing out the coefficients that are insignificant to the model, resulting in redundancy. We observed that the quadratic model was just right, capturing the trend of our time series without fitting noise. Furthermore, of all the regression models, it had the lowest RMSE score at 427.2484. This is lower compared to the linear model's RMSE: 2016.195 and the same RMSE as the cubic model without overcomplications. The same RMSE was found due to the high multicollinearity, which resulted in the quadratic model and cubic model to have the same coefficients (coefficient for time^3 is NA). Thus, we use a simpler and more effective quadratic regression model.

Regression Model: $Y_t = -60585143 + 60297 * time - 15 * time^2$

Through the model shown above, we were able to generate forecasts onto the testing data (year 2019) after fitting the quadratic model on the training data. The model itself only forecasted the trend for the year 2019, so we had to extract the seasonal effect from the multiplicative decomposition of the training data and multiply each element of the trend by its corresponding seasonal effect. The resulting time series is the $\hat{Y}_t$ column shown below.

Month	Y_t Raw Testing Data (Actual Observations from 2019)	$\hat{Y}_t$ Predicted Data (From Quadratic Polynomial Regression) $\hat{T} * \hat{S}$	$\hat{T}_t$ Estimated Trend (From Quadratic Polynomial Regression)	$\hat{S}$ Seasonal Effect (Extracted from Multiplicative Decomposition of Training Data)	$\hat{X}$ Residual of Raw Data Minus Predicted Data (From Quadratic Polynomial)
2019-01	7162	7044.520	7513.659	0.9375618	117.47975
2019-02	6562	6793.898	7490.672	0.9069812	-231.89827
2019-03	7192	7090.710	7467.476	0.9495458	101.28958
2019-04	6489	6410.620	7444.072	0.8611711	78.38017
2019-05	7112	6782.889	7420.459	0.9140794	329.11069
2019-06	6983	6900.253	7396.639	0.9328904	82.74653
2019-07	7170	6920.184	7372.610	0.9386342	249.81625
2019-08	7519	7199.514	7348.372	0.9797427	319.48607

2019-09	7123	6751.797	7323.926	0.9218822	371.20252
2019-10	7096	6708.599	7299.272	0.9190777	387.40149
2019-11	9242	8249.709	7274.410	1.1340727	992.29068
2019-12	10886	11630.554	7249.339	1.6043606	-744.55384

Table 5.1: Raw Testing Data and Forecasted Data using Quadratic Regression, along with Trend, Seasonal Effect, and Residuals of the Forecasted Data.

Next, we aim to determine whether our model is a good fit for our data. To do so, we examine the scaled residual plot, comparing the true values in the training set to $\hat{T} * \hat{S}$ from our model.

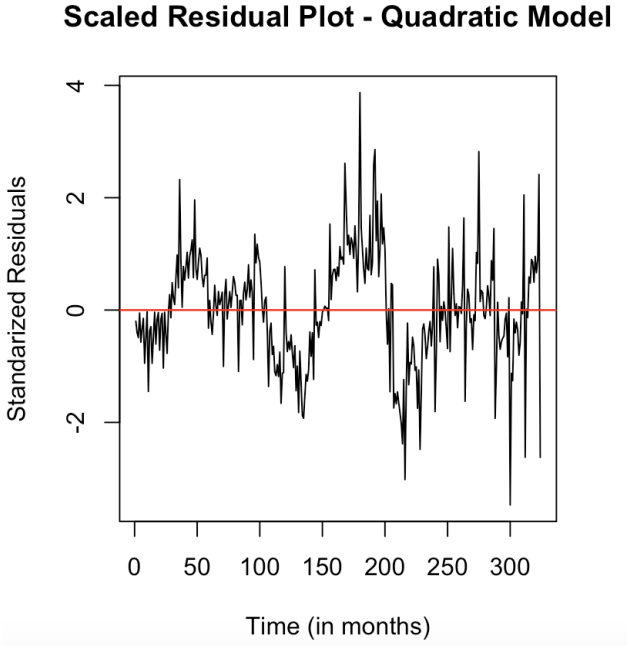

Fig 5.3: Time Plots of Scaled Residuals for Quadratic Model on Testing Data

While there does seem to be some cyclical pattern in the standardized residual plot, it does exhibit variation around 0 and it is more random than the linear and cubic residual plots. Although it's not an ideal model, it does have the lowest RMSE of all our polynomial models and does provide the best fit. Thus, we select the quadratic regression model. The final forecast for the quadratic regression model can be found in Table 5.1 in the $\hat{Y}_t$ column.

VI. Conclusion

Throughout this research, we have provided an extensive analysis on the time series data of Advance Retail Sales of Electronics and Appliance Stores to gain valuable insights on the retail industry. By analyzing invaluable components such as trends, seasonality, and making forecasts, we were able to gain a comprehensive understanding of the industry as a whole.

Through our analysis of the Electronics and Appliance retail sales data set, we can see that the variance of the time series begins to slowly increase over time, reach a peak around 2008, and then slowly begin to die down again. For the trend, it follows a similar pattern as the variance where it gradually increases until 2008; however, there is a very clear dip in the time series after that point in time. We saw that there was a gradual increase in spending from the beginning of our time series until around 2008 when the recession happened. After the 2008 recession, the amount of spending stayed relatively constant. The first portion of our time series plot where the spending gradually increased can possibly be due to inflation in the market as well as more economic prosperity; however, after the recession we can see that a large part of the consumer base may still have been impacted by this event.

Another thing that we saw with the data was a large amount of seasonality that occurred very consistently throughout the window of this time series. This can be seen as the peaks during the holiday season exponentially increase and return during the months after. This spike is happening obviously because electronics and appliances are commodities that are very popular during the Holidays, so consumers are more likely to purchase these items in a higher abundance. Lastly, we were also able to make forecasts on our time series, using different models.

Date	Raw Data Values	Exponential Smoothing Forecast	Quadratic Regression Forecast	Average Forecast
2019-01	7162	7059.742	7044.520	7052.131
2019-02	6562	6777.029	6793.898	6785.463
2019-03	7192	7368.711	7090.710	7229.710
2019-04	6489	6574.351	6410.620	6492.486
2019-05	7112	7037.990	6782.889	6910.440
2019-06	6983	7104.341	6900.253	7002.297
2019-07	7170	7006.499	6920.184	6963.341
2019-08	7519	7458.211	7199.514	7328.863

2019-09	7123	7056.559	6751.797	6904.178
2019-10	7096	7056.725	6708.599	6882.662
2019-11	9242	9221.868	8249.709	8735.789
2019-12	10886	10829.590	11630.554	11230.072
	RMSE:	**113.7903**	**427.2484**	**270.5194**

Best: Exponential Smoothing
Worst: Quadratic Regression

Table 6.1: Comparative Results of Forecast Methods

We fit two distinct models: Exponential Smoothing and Polynomial Regression. By fitting models using the raw training data of retail electronic sales, we can then forecast twelve months into the future, and compare it with our testing data set. We achieve the following results as shown in Table 6.1. Ultimately, we see Exponential Smoothing outperform Polynomial Regression due to a lower RMSE. So far, we would prefer to use Exponential Smoothing.

As for future steps, this project can be expanded to include more segments of the retail industry, or to compare the results with other economic indicators, such as consumer confidence, employment, and housing prices, to gain a more complete overview of the retail industry and its interactions with other economic variables. Furthermore, incorporating more advanced time series techniques, such as ARIMA, VAR, and other machine learning models, could provide even deeper insights into the relationships among the different retail segments.

—---------- End of Section 1 —----------

VII. ARIMA modeling and forecasting

Exploring other modeling and forecasting methods for our dependent variable – electronic retail sales – we look to one of the most widely used approaches: ARIMA. While ARIMA is a proven and flexible model, it requires pre-processing steps to fulfill its assumptions.

1. Address the need for pre-transformations

One of the most critical assumptions in ARIMA is that it requires a stationary time series. Thus, we must remove non-stationary features in the variance and mean of our time series. While

non-stationary mean is handled by differencing in ARIMA, we must handle non-stationary variance by pre-transforming our time series to ensure seasonality is constant throughout.

Possible pre-transformations include square root, quartic, and log transformations. The goal of these transformations is to ensure seasonal spikes are constant throughout the time series.

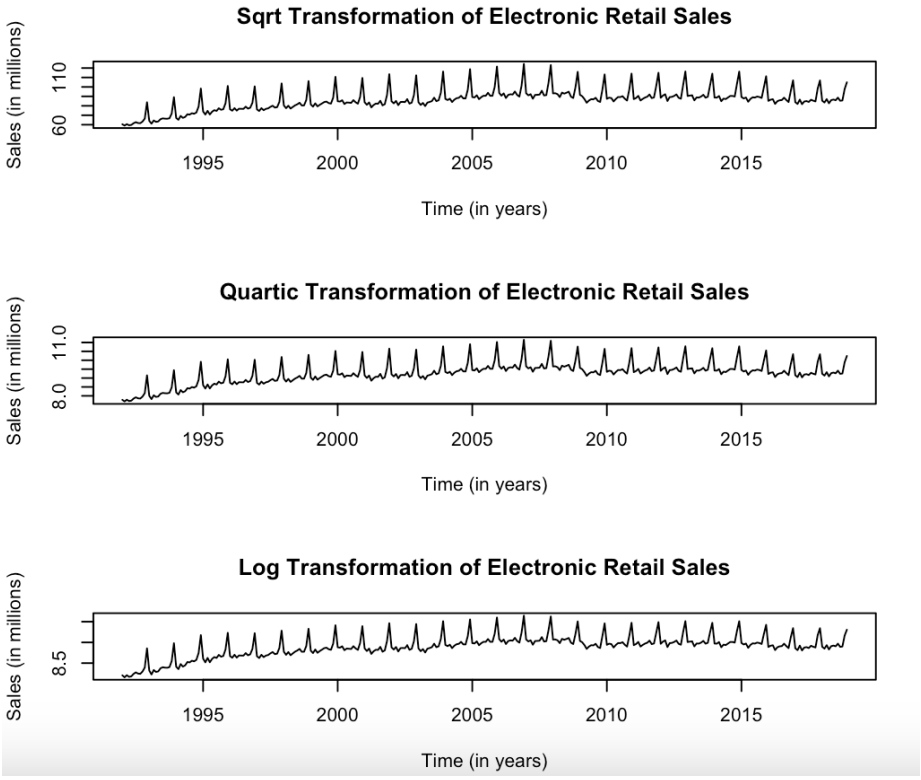

Fig 7.1: Pre-Transformations of Electronic Retail Sales to Assess Stationary Variance

To decide which pre-transformation is best, we look at which plot has the most constant seasonal spikes – this indicates constant variance. The square root transformation indicates higher seasonal spikes around 2005-2010 compared to earlier or before in the plot. Similarly while not as drastic, the same pattern can be seen in the quartic transformation. On the other hand, in the log transformation the seasonal spikes seem to be of the same magnitude all throughout the plot. Therefore, we use the log transformation to remove non-stationary variance.

2. Assessment of mean stationarity

While we now have variance stationarity, we must also have mean stationary to fit an appropriate ARIMA model to our time series. Thus, we must handle non-stationary mean if it exists by differencing our time series until our ACF plot returns a stationary model. First, we examine our current pre-transformed time series to determine if differencing is needed to create stationarity.

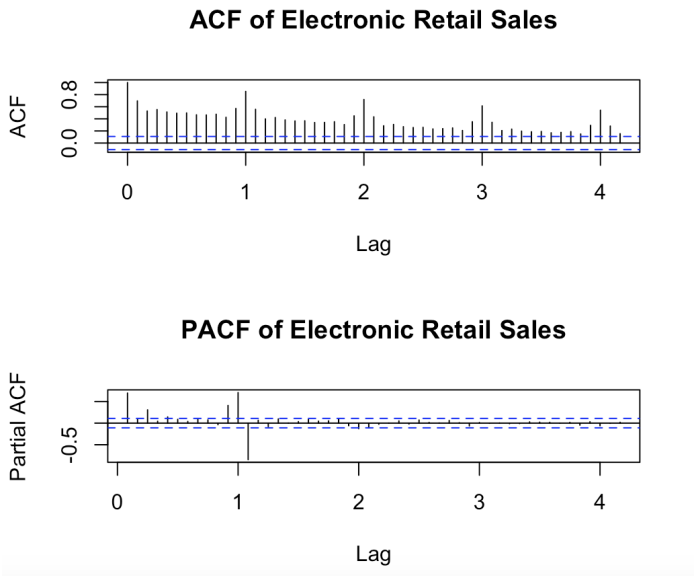

Fig 7.2: ACF and PACF Plot of Pre-Transformed Electronic Retail Sales Data

As we examine in Fig 7.2, our ACF plot is clearly not stationary with many significant autocorrelations, thus differencing is needed to make our time series stationary. Thus, we attempt three different differencing techniques: regular differencing, seasonal differencing, and both regular and seasonal differencing. We select the technique showing the most stationary in mean.

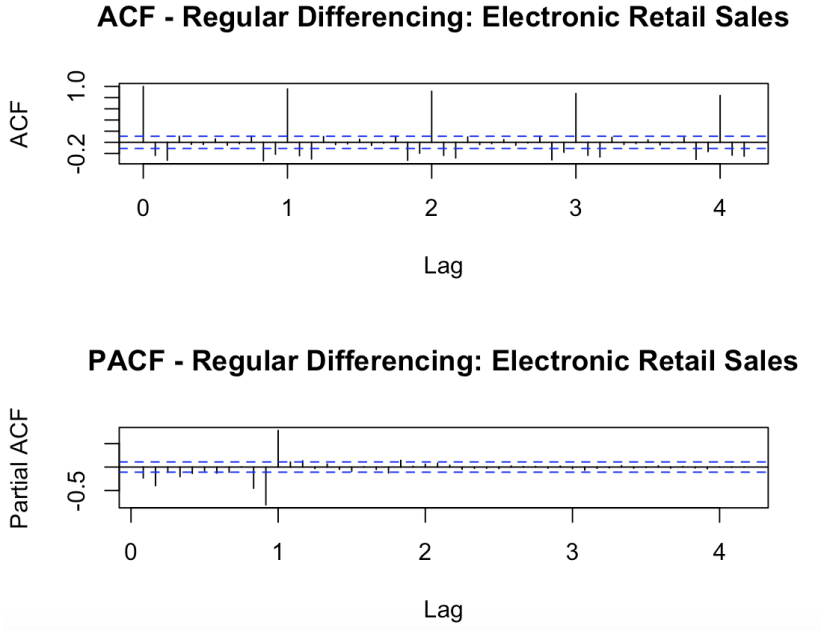

Fig 7.3: ACF and PACF Plot of Regularly Differenced Electronic Retail Sales Data

As we examine in Fig 7.3, this differencing technique definitely did not remove non-stationarity in mean. The ACF plot clearly shows many significant autocorrelations near seasonal lags.

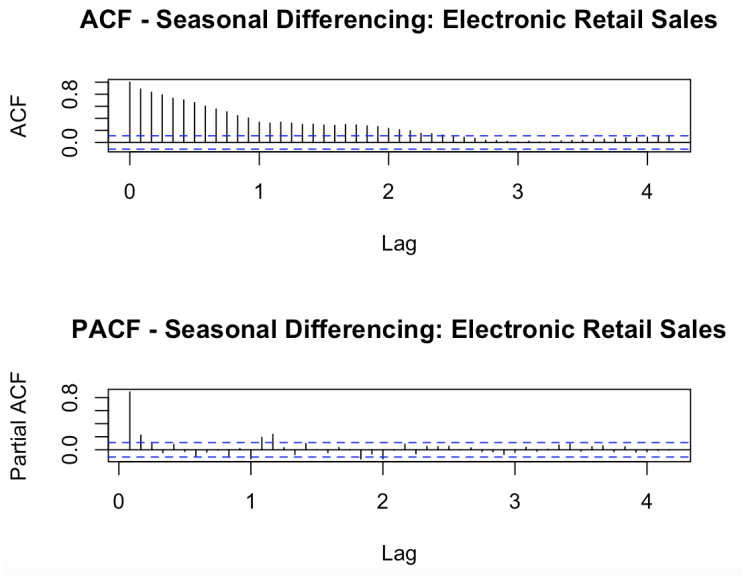

Fig 7.4: ACF and PACF Plot of Seasonally Differenced Electronic Retail Sales Data

As we examine in Fig 7.4, this differencing technique definitely did not remove non-stationarity in mean. The ACF plot clearly shows many significant autocorrelations in the first half of the plot and then slowly dying down. It appears that regular or seasonal differencing is not enough.

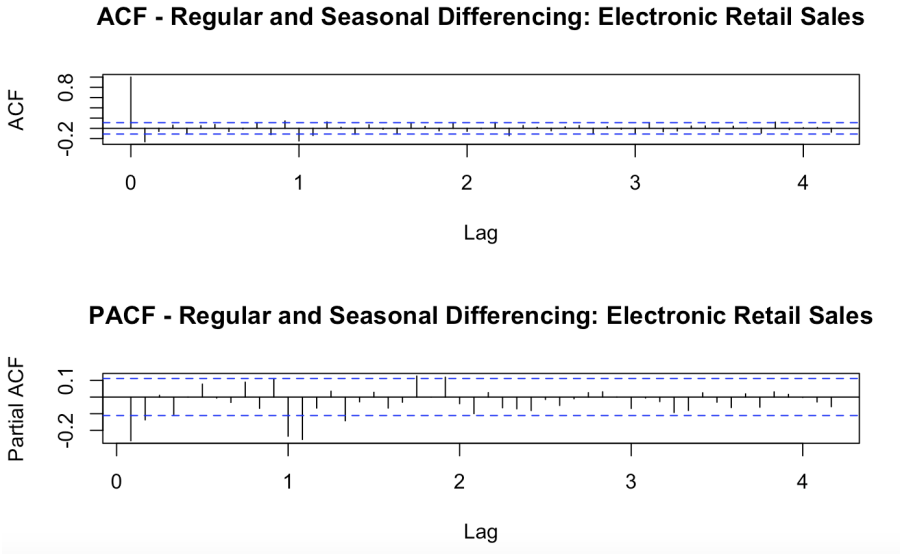

Fig 7.5: ACF & PACF Plot of Regularly and Seasonally Differenced Electronic Retail Sales

Both regularly and seasonally differencing our time series most effectively removes non-stationary features in the mean. The ACF plot best resembles a stationary time series with very few significant autocorrelations, except one at the start and a few at seasonal lags. The PACF plot also reveals key interpretations for ARIMA model fitting. Ultimately, due to the ACF plot resembling a stationary model, we use regular and seasonal differencing.

3. Identification

Now that we have produced a stationary time series by removing non-stationarity in mean and variance, we aim to fit an appropriate ARIMA(p,d,q)(P,D,Q)_F model to our time series. To do so, we analyze the ACF and PACF plots of our now stationary data and look for patterns.

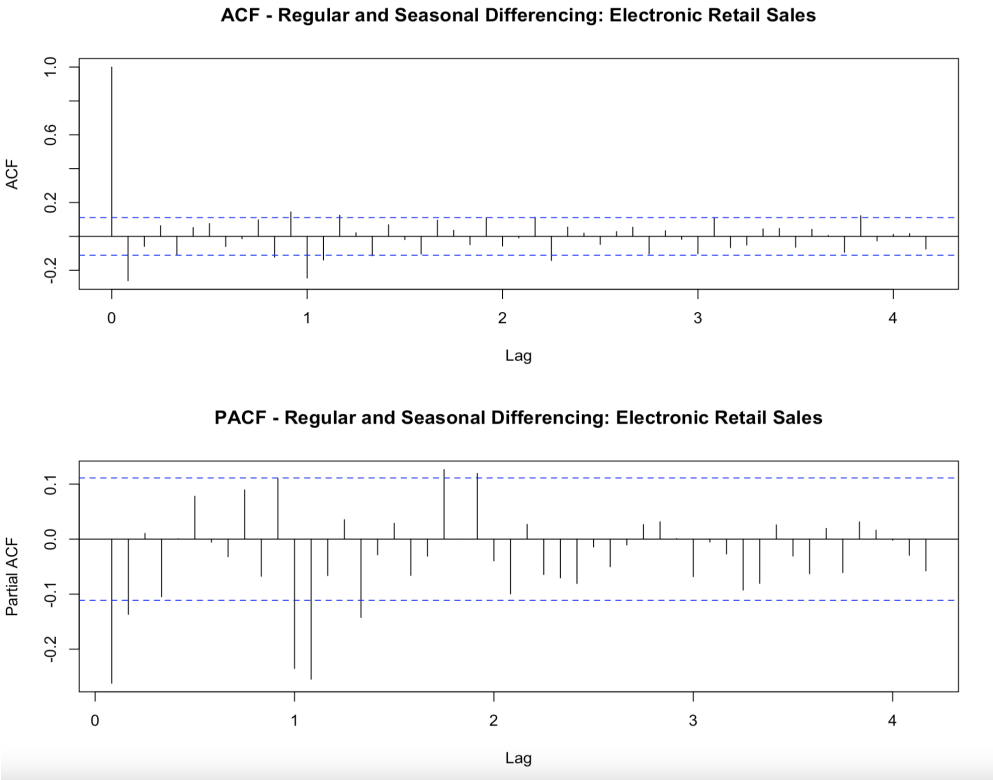

Fig 7.6: ACF and PACF Plot of Differenced Electronic Retail Sales Data (Zoomed In)

Let's first look at the regular portion of the ACF and PACF (autocorrelations in the plots before k = 1). In the ACF, we see that there are two significant autocorrelations at lag = 1, 11, excluding lag = 0 from our analysis. While lag = 1 is significant, because the points before lag = 11 are not significant, we can determine the lag = 11 is part of the 5% sampling error we allow. Additionally, when looking at the PACF we can see that the autocorrelations gradually decrease in this section in contrast to the ACF immediately dying off after lag = 1. Therefore, we can conclude that the regular part of our ARIMA model is most likely MA(1).

Now let's examine the seasonal portion of the ACF and PACF plots (autocorrelations beyond k = 1). In both plots, we see that there is a wave-like pattern that is gradually decreasing. Because we see no immediate drop in either plot, we can deduce that the seasonal component is likely to follow an ARMA(1, 1) model. If the residuals of our final model from these indications are not white noise, we can adjust our model at this point and compare the results later on.

Lastly, because we both regularly and seasonally differenced our data to make our time series stationary, we can easily tell the differencing component is 1 in both the regular and seasonal component of our ARIMA model. Thus, we find our model: **ARIMA(0,1,1)(1,1,1)_12**.

4. Fit and diagnose

While we have found a first tentative model, we still have to verify our model and determine whether it's the optimal ARIMA model for our time series. To do so, we must verify our model based on important selection criteria. First, we examine the ACF plot of our model residuals.

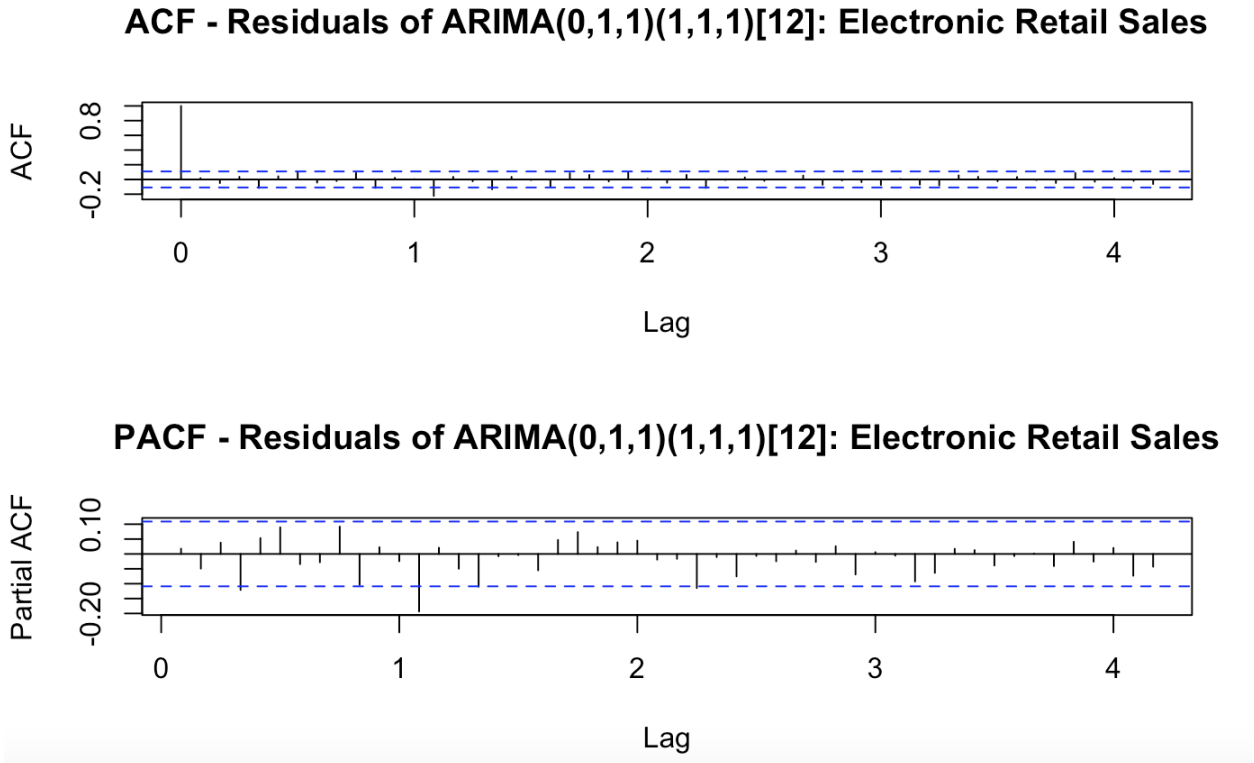

Fig 7.7: ACF and PACF Plot of Residuals from ARIMA(0,1,1)(1,1,1)_12

Our ARIMA(0,1,1)(1,1,1)_12 is a good fit if our ACF plot of residuals represents white noise. Although our ACF plot isn't an exact representation of white noise – it's nearly perfect. Outside of the single significant autocorrelation past lag k = 1, all of the autocorrelations are not significant, staying within the blue bands in our ACF plot.

We can also verify our models' residual being white noise by using the Ljung Box test. The Ljung Box test uses a Chi-Squared Test with null hypothesis that our model does not show lack of fit (time series is not serially correlated). Due to the nature of our model (containing seasonality every year), we set the number of lags in our Ljung Box test to 12. This resulted in the Ljung Box test of our model returning a p-value = 0.1164. We fail to reject the null

hypothesis and conclude our model is a good fit. Furthermore, we find our model's AIC score is -1361.687 – quite an effective model.

Yet, we still cannot be sure our ARIMA(0,1,1)(1,1,1)_12 is optimal without testing other ARIMA model variations. For this purpose, another possible effective ARIMA model we have identified by looking at Figure 7.6 is ARIMA(0,1,1)(0,1,2)_12. We follow the same steps.

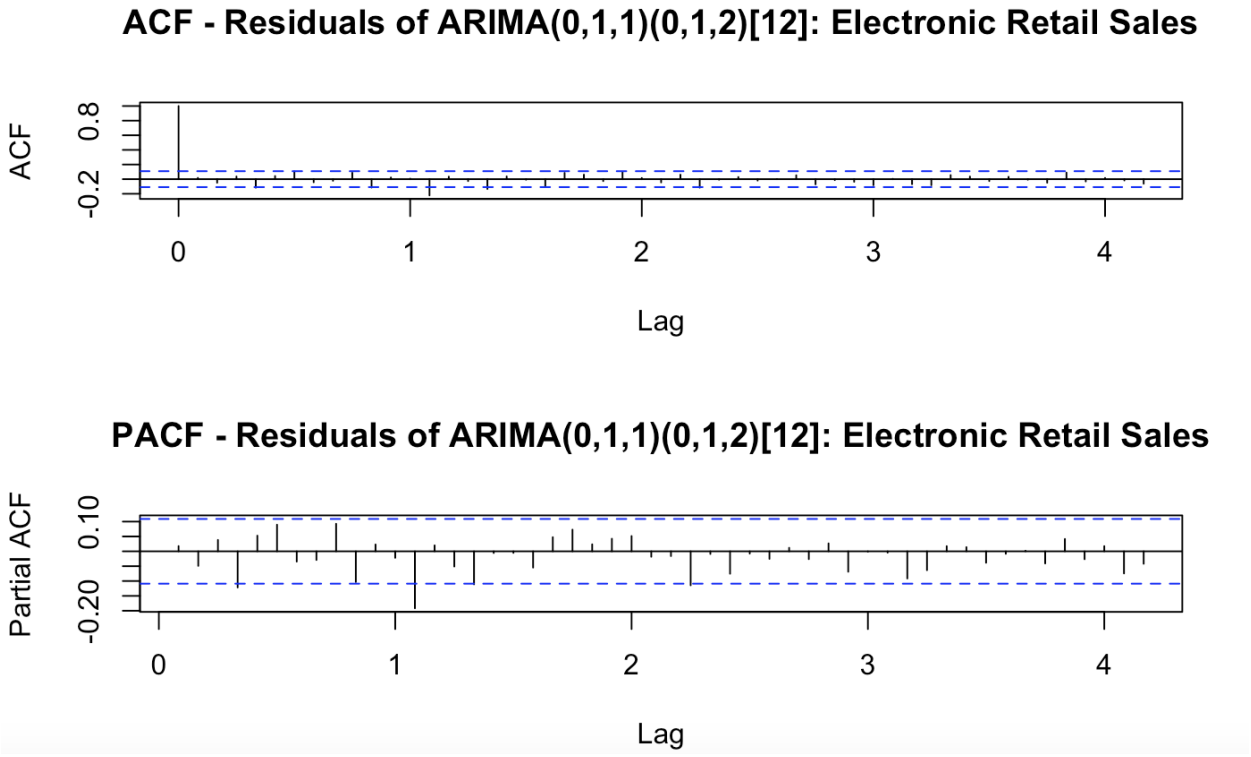

Fig 7.8: ACF and PACF Plot of Residuals from ARIMA(0,1,1)(0,1,2)_12

The ACF and PACF plot of the ARIMA(0,1,1)(0,1,2)_12 appear nearly identical to our previous ARIMA(0,1,1)(1,1,1)_12 model, except with slightly stronger autocorrelations present in both the ACF and PACF plot. Therefore, we must also compare the Ljung Box test and AIC scores. The Ljung Box test for this model's residuals returns a p-value = 0.1131, similarly we conclude this model is also a good fit. Lastly, the AIC score is -1361.519, a slightly higher AIC score than our previous ARIMA(0,1,1)(1,1,1)_12 model. Ultimately, since our AIC score is lower for our first model, we conclude that **ARIMA(0,1,1)(1,1,1)_12** is still our optimal model.

When looking at our final model, we are given coefficients: ma1 = -0.3745176, sar1 = 0.1843340, and sma1 = -0.5782330. Because we also both regularly and seasonally differenced our data, this modifies our polynomial. The resulting polynomial is:

$$(1 - 0.1843340B^{12})(1 - B)(1 - B^{12})Y_t = (1 - 0.3745176B)(1 - 0.5782330B^{12})W_t$$

In the ARIMA model, the AR component is always invertible, but we need to determine if that component is stationary. To do this, we check if the absolute root of the polynomial on the AR side is greater than 1. Using R, we check for the root of $(1 - 0.1843340B^{12})$, which gives us 5.425936. This means that the AR component of our model is stationary.

When looking at the MA component of the model, we know that the MA component is always stationary; however, we don't know if it is invertible, so we must check for that. To do this, we need to determine if the absolute roots of the polynomial on the MA side is greater than 1. Using the same methodology as above, we find the roots of polynomials $(1 - 0.3745176B)$ and $(1 - 0.5782330B^{12})$, obtaining the values 2.670227 and 1.729505 which corresponds to the root of each polynomial respectively.

To determine if these coefficients are significantly different from zero, we must utilize the t-test in order to determine the significance. We must conduct a hypothesis test where we have:

$$H_0: \hat{\alpha}_1 = 0 \text{ and } H_A: \hat{\alpha}_1 \neq 0$$

We then find the t-test statistic by doing $\frac{\hat{\alpha}_1 - 0}{se_{\hat{\alpha}_1}}$. If the absolute value of this equation is greater than 2, that means that our $\hat{\alpha}_1$ is more than two standard errors away from zero and thus proving that the coefficient is significant. We do this process for all three coefficients and get that the t-test for our ma1 coefficient is -6.733550, the t-test for our sar1 coefficient is 1.648812, and the t-test for our sma1 coefficient is -6.346381. We can see that the absolute values of our ma1 and sma1 t-tests are greater than 2, meaning significantly different than 0, but the sar1 t-test is not.

Based on this, we looked back at our model and removed the AR(1) component from our final model and ran the same analysis we performed (ACF of residuals, Ljung-Box test, AIC test); however, the results of this new model were simply worse than when we left the sar1 coefficient untouched. The modified model had an AIC score that was worse than our original model and had a p-value of 0.03403 in the Ljung-Box test, meaning that the residuals are not white noise. Because of this, we determined that our original model is better than our modified model, so we will continue using our unmodified model.

Using our final ARIMA model, we can isolate our Y_t value to create our forecasting equation. After rearranging the polynomial form ARIMA model above, we obtain the forecasting equation:

$$Y_t = Y_{t-1} + 1.1843340Y_{t-12} - 1.1843340Y_{t-13} - 0.1843340Y_{t-24} + 0.1843340Y_{t-25} + W_t - 0.5782330W_{t-12} - 0.3745175W_{t-1} + 0.2165584W_{t-13}$$

5. Forecasting

With our final ARIMA model, we can now forecast our model onto the testing set.

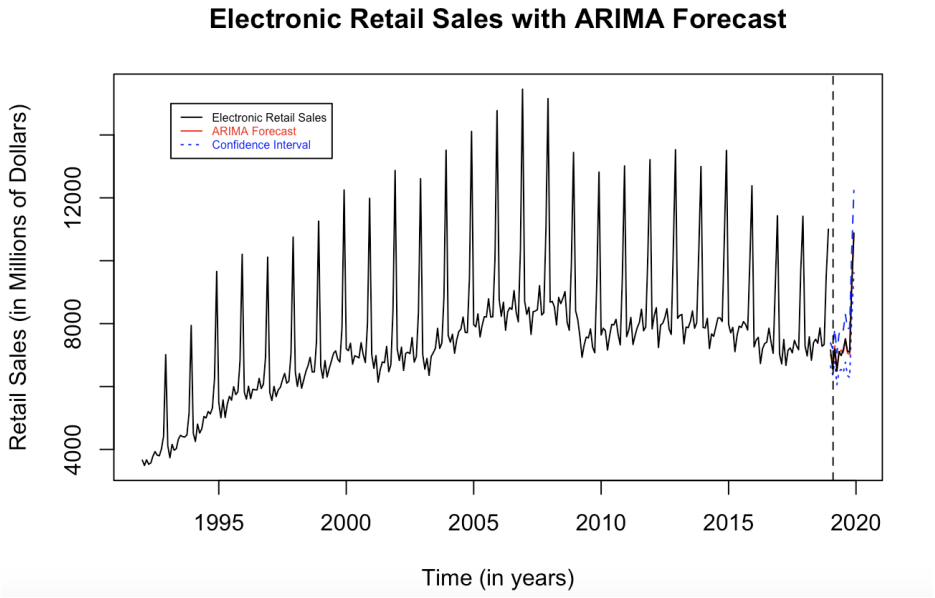

Fig 7.9: Time Series Plot of Raw Retail Electronic Sales With ARIMA Forecast

We can also examine a more in-depth look into our ARIMA Forecast with prediction intervals.

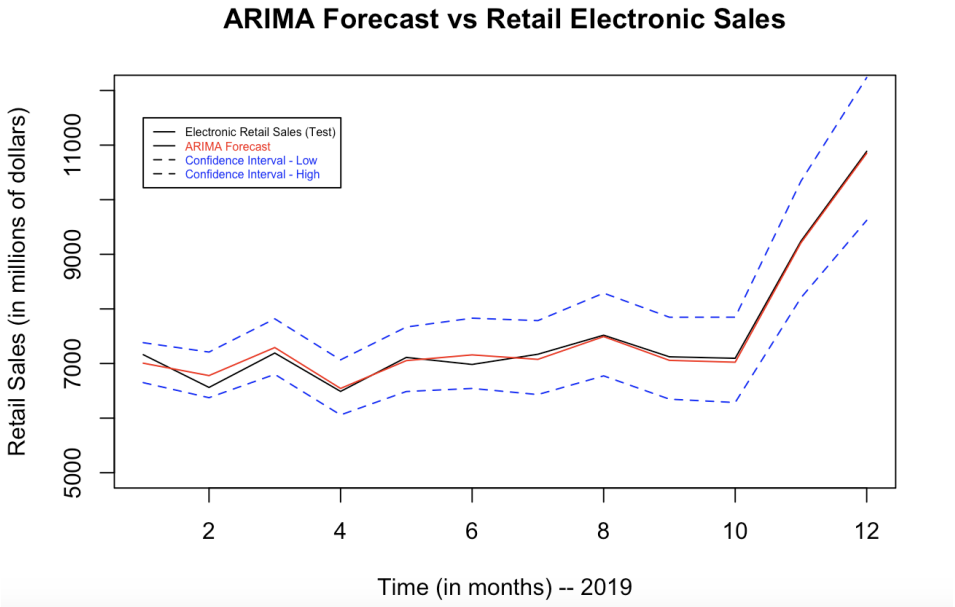

Fig 7.10: Multiple Time Series Plot of Raw Retail Electronic Sales Testing Data and ARIMA Model Forecast With 95% Prediction Intervals

Our ARIMA model performs quite well as our electronic retail sales testing data fits very comfortably within our prediction intervals as seen in figure 7.10. Furthermore, the RMSE score is 107.9269, indicating that our ARIMA model is currently our best performing model.

VIII. Multiple regression with ARMA residuals

Throughout our analysis of our electronic retail sales time series so far, we have mostly been using models that forecast the future based on past values, however, we can also explore regression-based causal models which create forecasts based on other similar variables.

We build a multiple regression model to predict electronic retail sales based on our independent variables, hobby retail sales and furniture retail sales. However, a raw multiple regression model fails to produce white noise residuals, indicating a poorly fit model. In order to build an appropriate fit causal model whose residuals produce white noise, we must transform our dependent and independent variables with a log transformation. Our model is given below:

Multiple Regression Model:

$$Y_t = e^{0.88246 + 0.74006 \times X1.star_t + 0.18189 \times X2.star_t}$$

In this equation, $X1.star_t$ represents the logged value of the retail hobby sales time-series at time t and $X2.star_t$ represents the logged value of the retail furniture sales time-series at time t. The median residual value is 0.01538 and the residual standard error is 0.09111.

As we examine the standardized residual plot of our pre-transformed multiple regression model, we find there is trend and seasonality present. Thus, we apply regular and seasonal differencing.

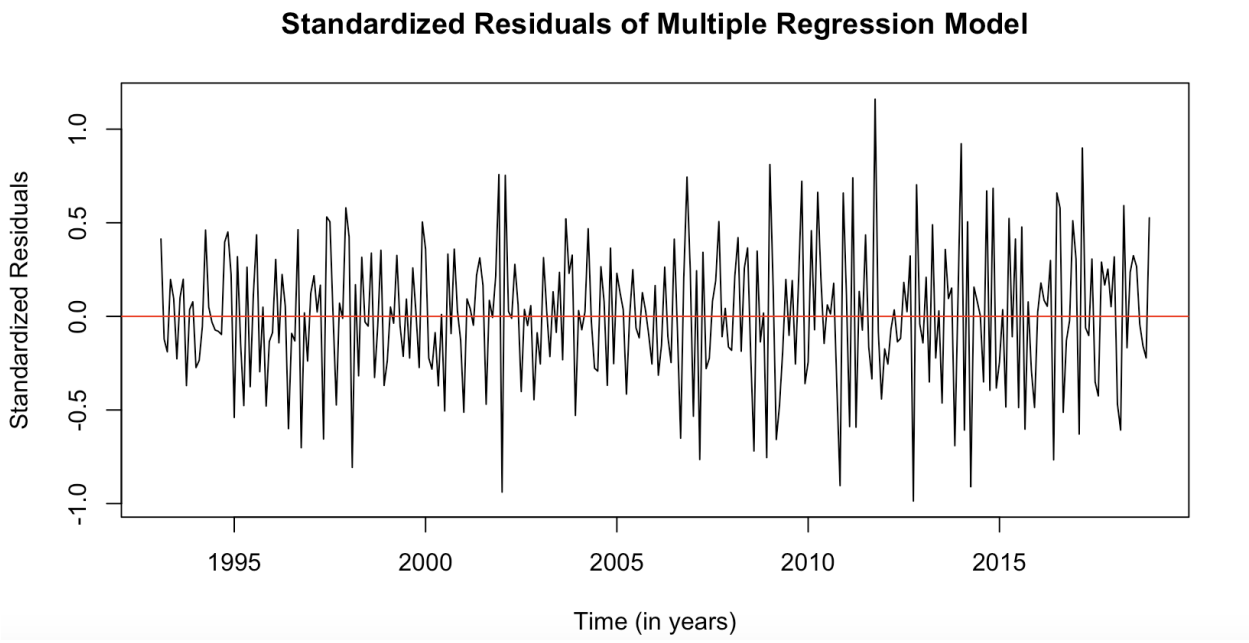

Fig 8.1: Standardized Residuals of Pre-Transformed Multiple Regression Model After Differencing. Models Electronic Retail Sales From Hobby and Furniture Retail Sales

Our results seem quite stationary. However, we also must examine the ACF and PACF plots.

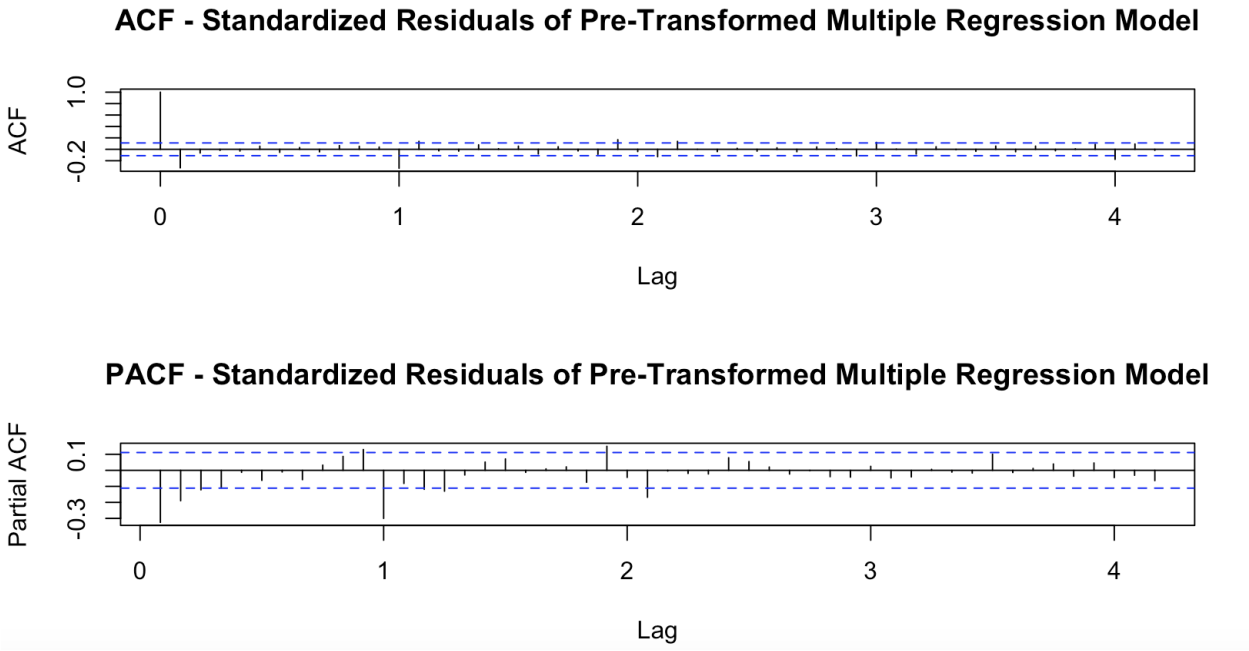

Fig 8.2: ACF and PACF Plot of Standardized Residuals From Multiple Regression Model

Looking at the ACF and PACF plots of our standardized residuals, we can now fit an ARMA model to our stationary residuals and incorporate that into our overall model for best results.

Now, let's fit an ARMA model. Let's first look at the regular portion of the ACF and PACF (autocorrelations in the plots before k = 1). In the ACF, we see that although there is only 1 significant autocorrelation, indicating an MA(1) model, we also see that the ACF is not gradually dying down, implying the possibility of being an ARMA(1, 1) model. Further expanding upon this, we look at the PACF. In the PACF, we can see that there is a geometric wave-like pattern; however, similarly to the ACF, the pattern is not dying down, which solidifies our notion that the regular portion of the data can be fit by an ARMA(1, 1) model.

Now let's examine the seasonal portion of ACF and PACF plots (autocorrelations beyond k = 1). We notice a strikingly similar pattern as with the regular portion of these plots. In the ACF, we see 1 very significant autocorrelation present at seasonal lags. We also see a few weaker significant autocorrelations, but this can be attributed to sampling error – roughly 5% of autocorrelations that are not significant will appear as significant. With that being said, the wave-like pattern shows no sign of dying away and thus, shows a signal that the seasonal component can be fit by an ARMA(1, 1) model. In the PACF, we see a constant wave-like pattern that's amplitude remains at a constant height, strongly reinforcing the previous statement made about the fit of the model.

Thus, we fit an ARMA(1, 1) model to both the regular and seasonal components. Ultimately, we fit the following model to the residuals of our multiple regression model:
ARIMA(1,1,1)(1,1,1)_12.

Next, we must verify if our ARIMA model is an appropriate fit to our residuals.

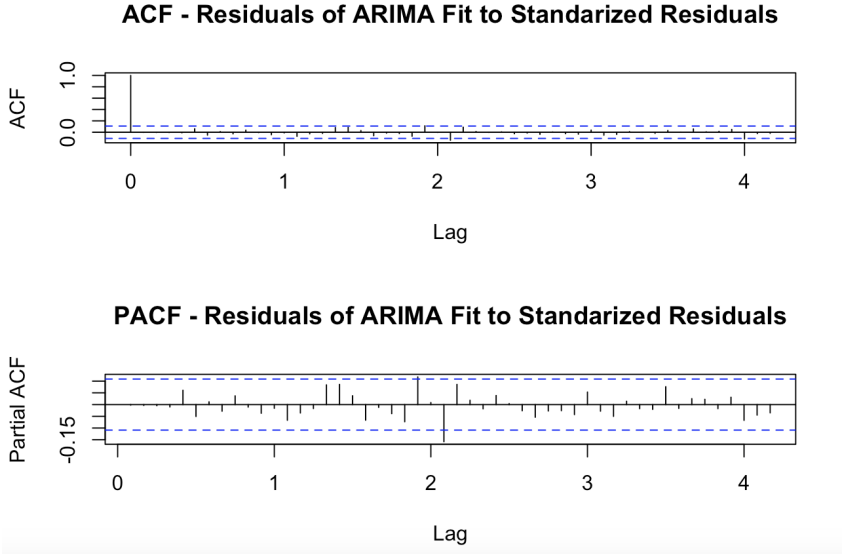

Fig 8.3: ACF and PACF Plot of Residuals from ARIMA Fit to Standardized Residuals.

We believe our ARIMA model is an excellent fit to the standardized residuals as its residuals represent white noise, as seen in Figure 8.3. The ACF plot shows no significant autocorrelations. Furthermore, we can also verify this using the Ljung Box test. Because our Ljung Box test returns p-value = 0.99, we fail to reject the null hypothesis, concluding our model is a good fit.

Now that we have an ARIMA model fit to our residuals, we can now incorporate that into our modeling for best results by using a Generalized Least Squares Model. Thus, we extract MA(1) and AR(1) coefficients from the regular part of our ARIMA model and use it as a correlation value in our GLS model. Our MA(1) coefficient is -0.6293. Our AR(1) coefficient is 0.1547. Thus, we fit the following model:

Generalized Least Squares Model:

$$Y_t = e^{-0.1585686 + 0.5916488 \times X1.star_t + 0.4427037 \times X2.star_t}$$

For context, in this equation $X1.star_t$ represents the logged value of the retail hobby sales time-series at time t and $X2.star_t$ represents the logged value of the retail furniture sales time-series at time t. Additionally, the median of the standardized residuals are 0.1257252 and the residual standard error is 0.1000981.

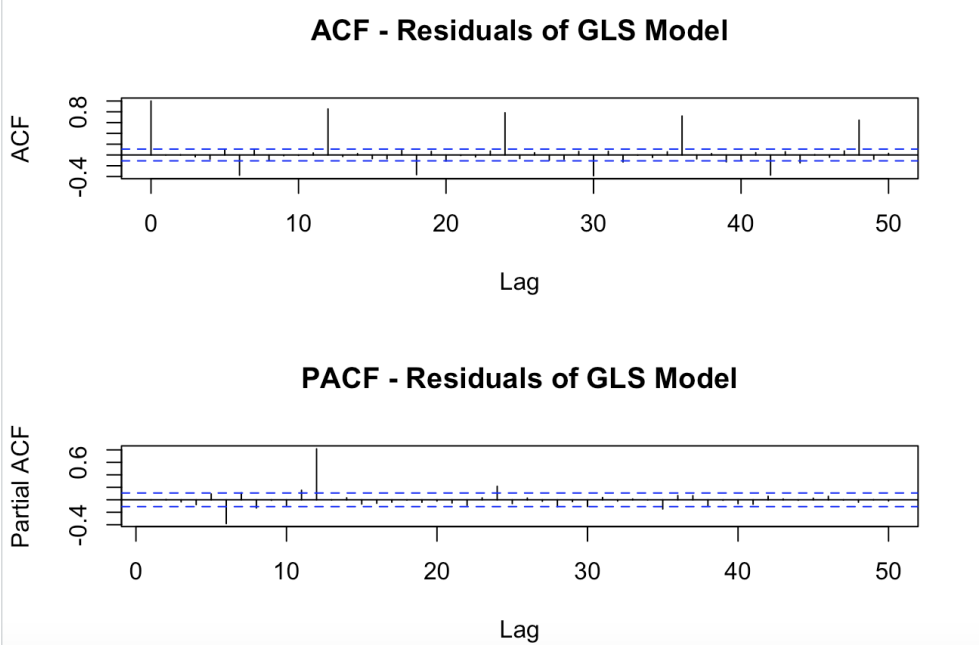

Fig 8.4: ACF and PACF Plot of Residuals from GLS Model

Although the residuals of our GLS model do not produce white noise, this is the best fitting GLS model we could fit. We experimented with multiple pre-transformations and different ARMA residuals to feed into our GLS model, yet this combination produces the lowest RMSE among GLS models that we tried and its residuals are no less stationary than other models. It appears that multiple regression with ARMA residuals is not a proper forecasting method for our data, due to the inability to account for seasonal components in the correlation feature of GLS.

Next, we forecast onto our testing set of Electronic Retail Sales data using our GLS model.

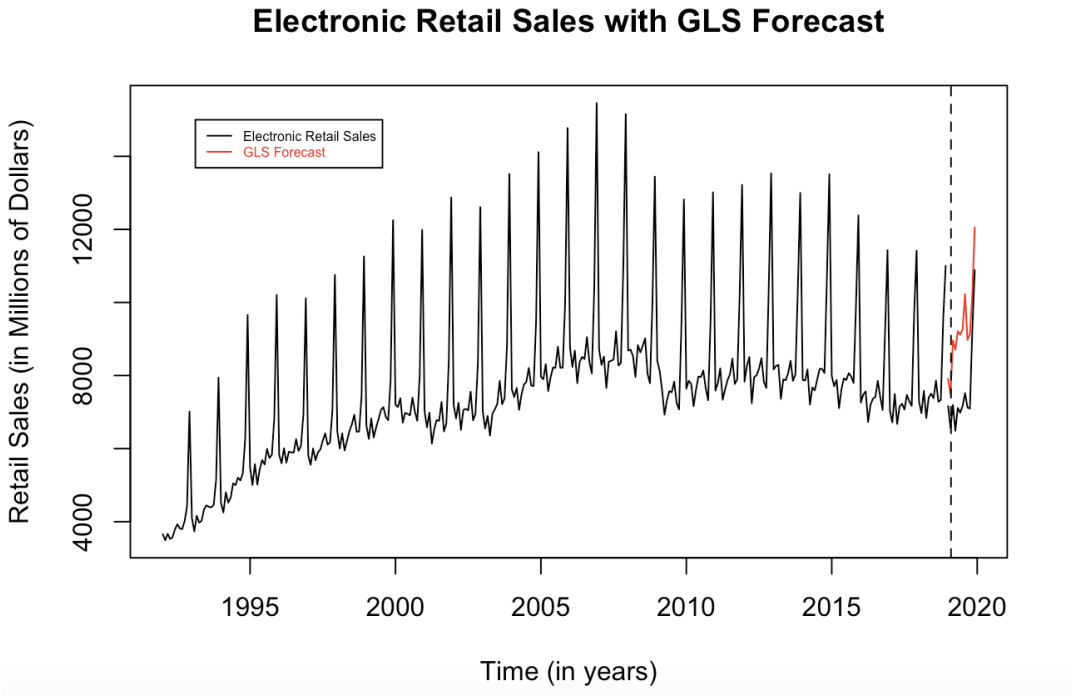

Fig 8.5: Time Series Plot of Raw Retail Electronic Sales With GLS Forecast

We can also examine a more in-depth look into our GLS Forecast with prediction intervals.

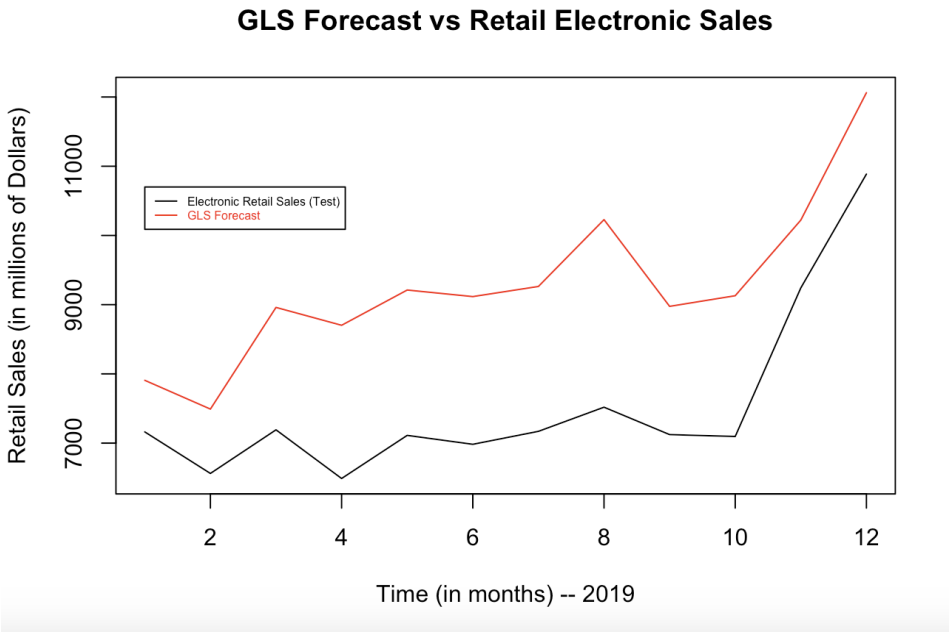

Fig 8.6: Multiple Time Series Plot of Raw Retail Electronic Sales Testing Data and VAR Model Forecast With Prediction Intervals

Our GLS model performs quite poorly as the electronic retail sales testing always sits below our forecasts. We attribute this to causal models not being an appropriate fit to our data. Furthermore, the inability of the GLS to model the seasonal components of our residuals lead to overestimating seasonal spikes, leading to a poorly performing model. Furthermore, the RMSE score is 1826.47, indicating that our GLS model is currently our worst performing model.

IX. Vector autoregression

In this section of the research paper, we will explore the Vector Autoregression (VAR) model. The VAR model is a multivariate time series model that allows us to analyze the relationships between multiple variables over time. Specifically, each variable in a system is regressed on its own past values as well as the past values of all the other variables in the system. The VAR model captures both the contemporaneous and lagged relationships between the variables, allowing for a more comprehensive understanding of their interdependence and the impact of one variable on the others. In our analysis of the retail industry, we will use the VAR model to examine the interactions between our dependent variable, retail sales of electronics and independent variables, retail sales of hobbies, and retail sales of furniture.

Previously, we illustrated the autocorrelation function (ACF) which measures the autocorrelation of a single time series with itself. To get a more dynamic picture of the VAR model, we introduce

the cross-correlation function (CCF) which measures the correlation between two different time series. We start by exploring the correlation between variables and finding significant lags.

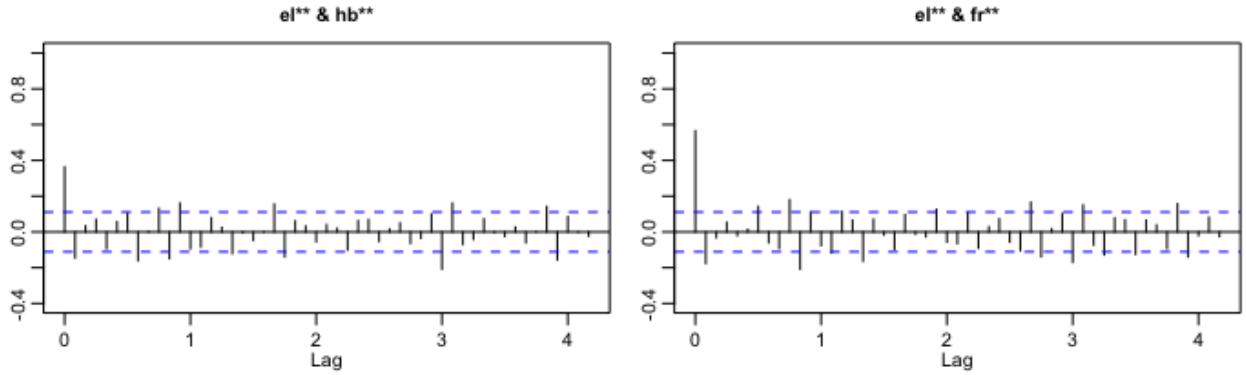

Fig 9.1: CCF Plot: Effect of Retail Hobby Sales on Retail Electronic Sales (depicted on the left) CCF Plot: Effect of Retail Furniture Sales on Retail Electronic Sales (depicted on the right)

We aim to examine the two CCF plots to identify the lags at which the correlation is significant. More specifically, we look for the significant peaks and troughs in the plot. We see that the CCF plot for Retail Hobby Sales on Retail Electronic Sale (Fig 9.1 Left) shows a significant peak at lag 11. Furthermore, the CCF plot for Retail Future Sales on Retail Electronic Sale (Fig 9.1 Right) shows a significant peak at lag 10. Thus, the tentative order of the VAR model will be 11.

As we determined the order of the model, we will next construct the model considering the parameter "p" to be 11. As this will be a very large model, certain coefficients can be insignificant. Ultimately, we only want to construct a model with significant coefficients as we want to make our model effective, but we also want to keep it as simple as possible. Let "e" represent electronics, "h" represent hobbies, and "f" represent furniture.
Thus, the three models will be:

Equation 9.1: Electronics

$$e_t = 0.821092 * e_{t-1} + 1.000956 * h_{t-1} - 1.018519 * f_{t-1} + 0.654822 * f_{t-3} - 0.674638 * h_{t-4} + 0.544872 * f_{t-4} + 0.762893 * h_{t-5} + 0.373897 * e_{t-6} + 0.599314 * h_{t-7} - 0.863093 * h_{t-8} + 1.834374 * h_{t-9} - 0.602400 * f_{t-9} + 0.943366 * e_{t-10} - 1.580715 * h_{t-10} - 0.458425 * f_{t-10} + 0.480967 * f_{t-11} + w_t$$

Equation 9.2: Hobby

$$h_t = 0.10541 * e_{t-1} + 0.70516 * h_{t-1} - 0.41361 * f_{t-1} + 0.21312 * h_{t-2} - 0.17140 * f_{t-2} + 0.25321 * h_{t-3} + 0.14612 * f_{t-3} - 0.20188 * e_{t-4} + 0.13153 * f_{t-4} - 0.44817 * h_{t-5}$$

$$+ 0.12198 * e_{t-6} - 0.22678 * h_{t-6} - 0.34819 * h_{t-8} + 0.60660 * h_{t-9} + 0.36117 * e_{t-10} - 0.71265 * h_{t-10} - 0.10237 * e_{t-11} + 0.24781 * f_{t-11} + 279.22517 + w_t$$

Equation 9.3: Furniture

$$f_t = 0.51168 * e_{t-1} + 1.00894 * h_{t-1} - 0.65622 * f_{t-1} - 0.34296 * e_{t-3} - 0.61966 * f_{t-3} - 0.65531 * e_{t-4} + 0.66630 * f_{t-4} + 0.81732 * h_{t-5} + 0.41822 * e_{t-6} - 0.98736 * h_{t-6} + 0.77700 * h_{t-7} - 0.86167 * h_{t-8} + 1.67396 * h_{t-9} - 0.50475 * f_{t-9} + 0.81070 * e_{t-10} - 1.50191 * h_{t-10} - 0.34971 * f_{t-10} + 0.44669 * f_{t-11} + w_t$$

To determine which variable leads and what variable lags, we must consider which variable affects the other variable at an early lag. Note that in equation 9.1, 9.2, and 9.3, all of the variables are significant at lag of 1, which is the earliest lag. In other words, the retail sales for electronics, hobby, and furniture last month affects retail sales for electronics, hobby, and furniture this month. This can be said for all other existing lags (For example, retail sales for furniture three months ago affects retail sales for electronics this month). The significant coefficients indicate that each variable is influenced by the others at an earlier time period, suggesting that the variables are interdependent and influence each other in a dynamic way. This follows our intuition as all three variables record retail sales and thus should show close dependence. In this case, it may be more appropriate to consider the direction of causality based on theoretical considerations or prior knowledge of the underlying economic relationships between the variables. Alternatively, we can consider the amount of times each of the variables is significant in the three equations to gain an understanding of leading indicators. In all of the equations above, hobbies show the greatest presence for various lags. This is a signal that retail sales of hobbies may be the leading indicator for retail sales of electronics, hobby, and furniture.

Next, we aim to look at the impulse response function which shows the dynamic response of each variable in the system to a one-unit shock to a particular variable while holding all other variables constant. It can provide insights into the direction, size, and duration of the response of each variable to a shock.

The horizontal axis of the IRF plot represents the amount of periods after the shock occurred, and the vertical axis represents the magnitude of the response in terms of standard deviation.

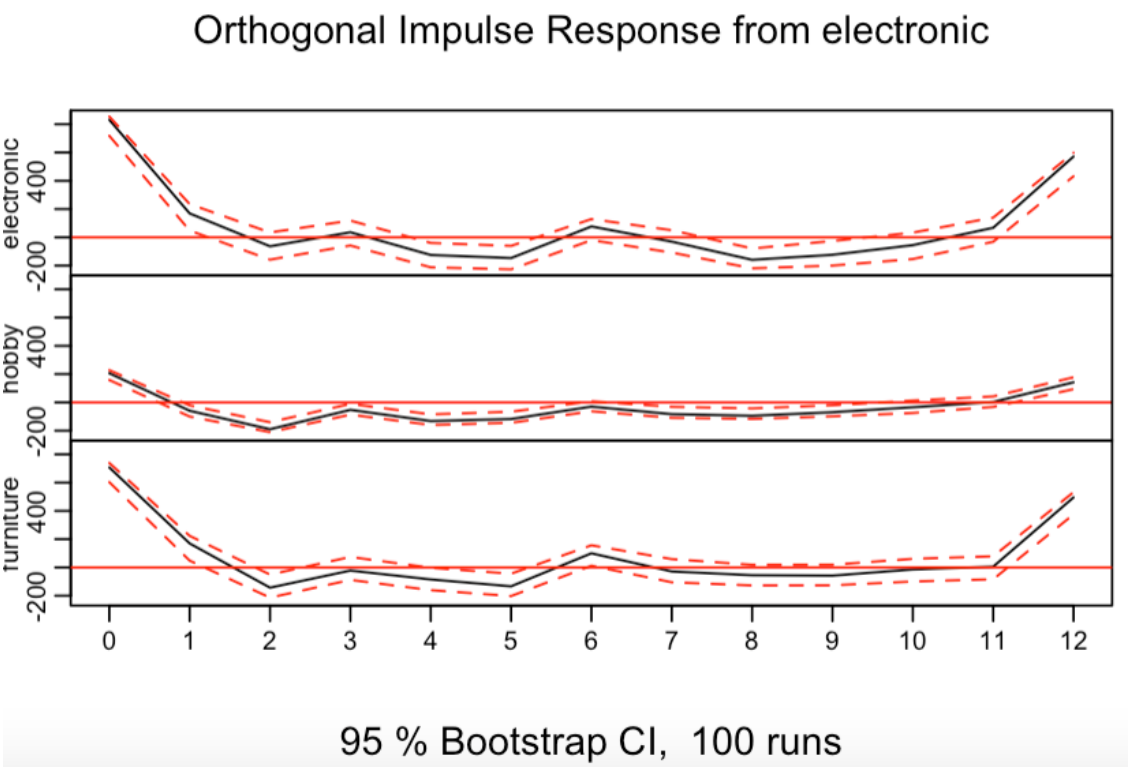

Fig 9.2: Effect and Length of the Effect of a Shock from Electronic to the System

The length of the effect can be inferred from the length of time it takes for the response to decay to zero, or the point at which the line crosses the x-axis. The length of the effect may vary for different variables and different lags.

From fig 9.2, we notice that 2 months after the shock occurred from the electronic variable, electronics and furniture go back to equilibrium. However, just 1 month after the shock occurred, the hobby variable goes back to equilibrium.

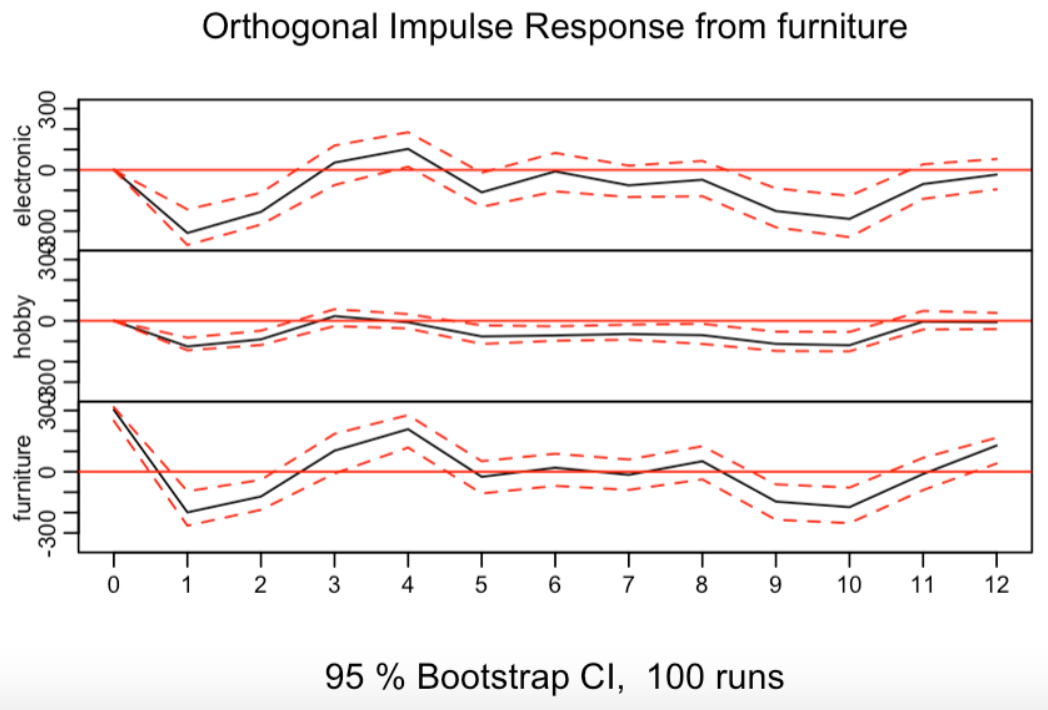

Fig 9.3: Effect and Length of the Effect of a Shock from Furniture to the System

Looking at fig 9.3, it takes 12 months for electronics and furniture to get back to equilibrium and 3 months for hobby to get back to equilibrium.

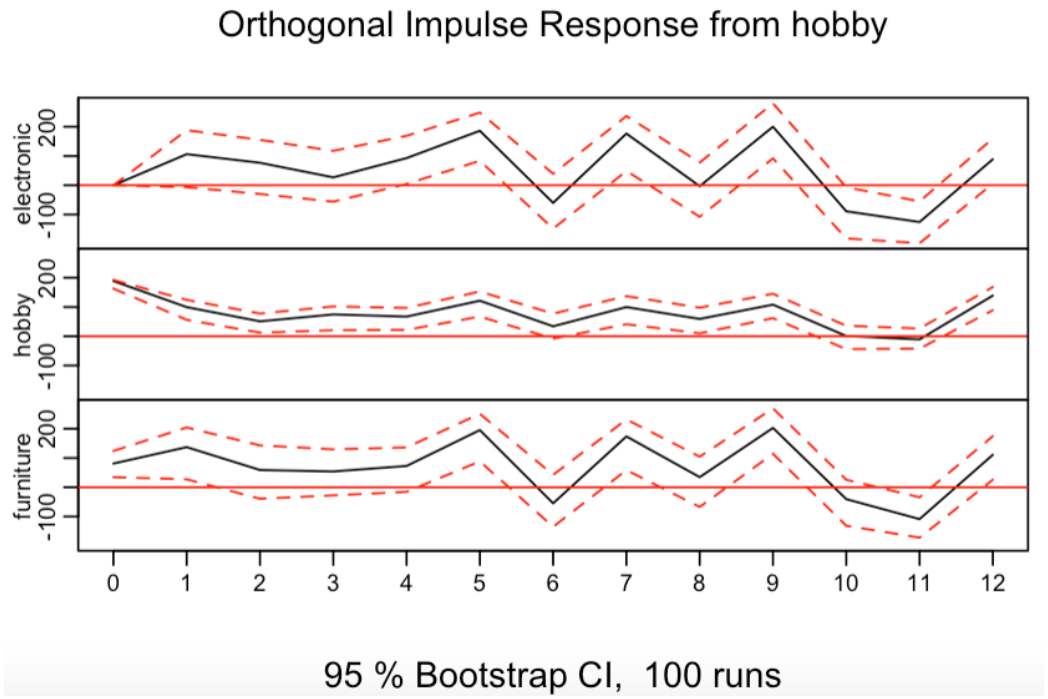

Fig 9.4: Effect and Length of the Effect of a Shock from Hobby to the System

Looking at fig 9.4, we notice that if shock occurs in a hobby, it takes 10 months for hobby and furniture to get back to equilibrium, and 5 months for electronics.

Overall, the impulse response function provides us a similar conclusion as what we have achieved through constructing the equations. We see that the variables are interdependent, but hobby has slightly shown greater significance to the three variables as seen through the most instances where the entire 95% confidence interval does not include the value 0. However, this is a very meager difference compared to the other two variables, as all three variables are generally around the red horizontal line at y = 0. In conclusion, we notice that there are no great spikes or long time periods, proving the previous comment about interdependence between variables.

With our final VAR model, we can now forecast our model onto the testing set.

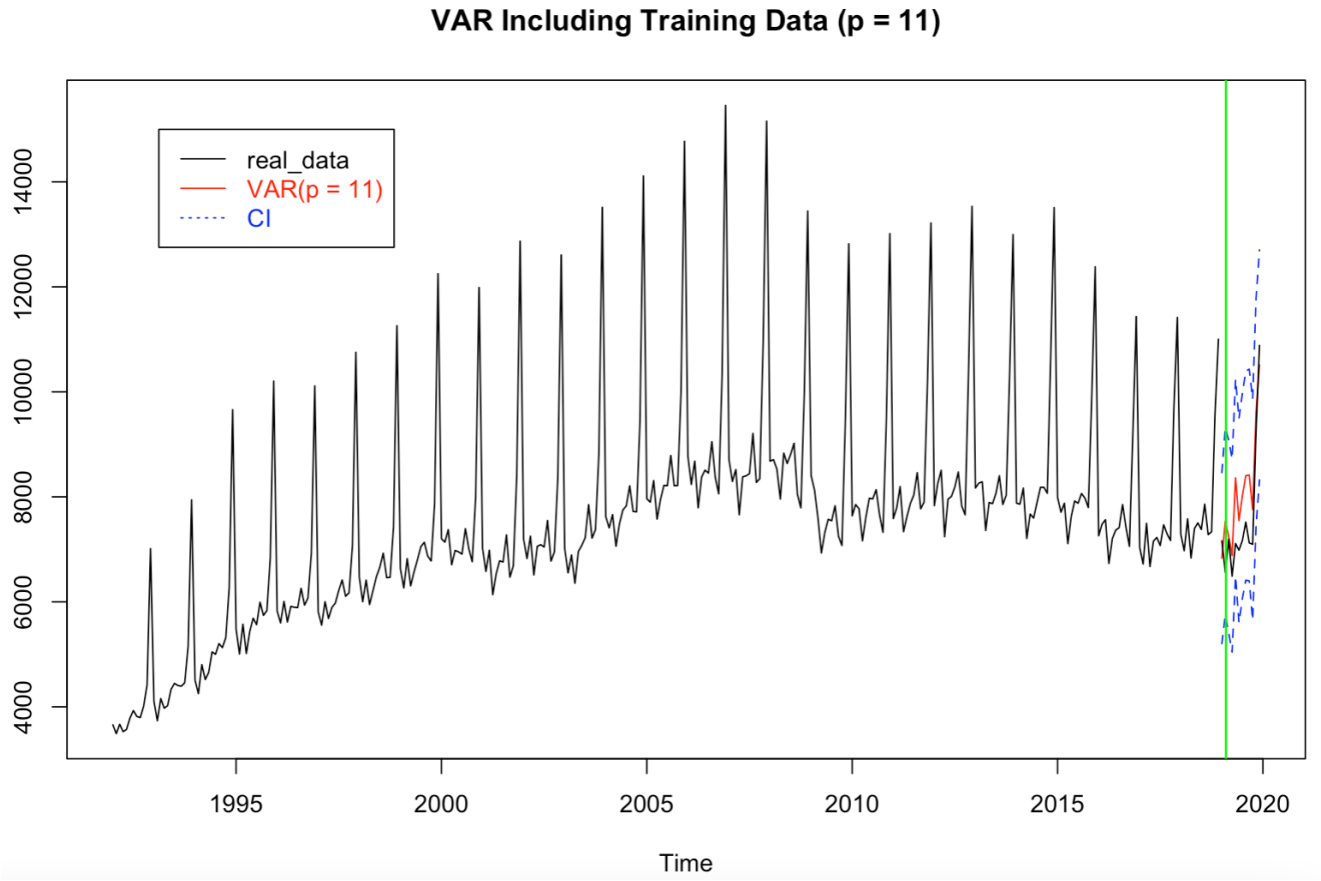

Fig 9.5: Time Series Plot of Raw Retail Electronic Sales With VAR Forecast

We can also examine a more in-depth look into our VAR Forecast with prediction intervals.

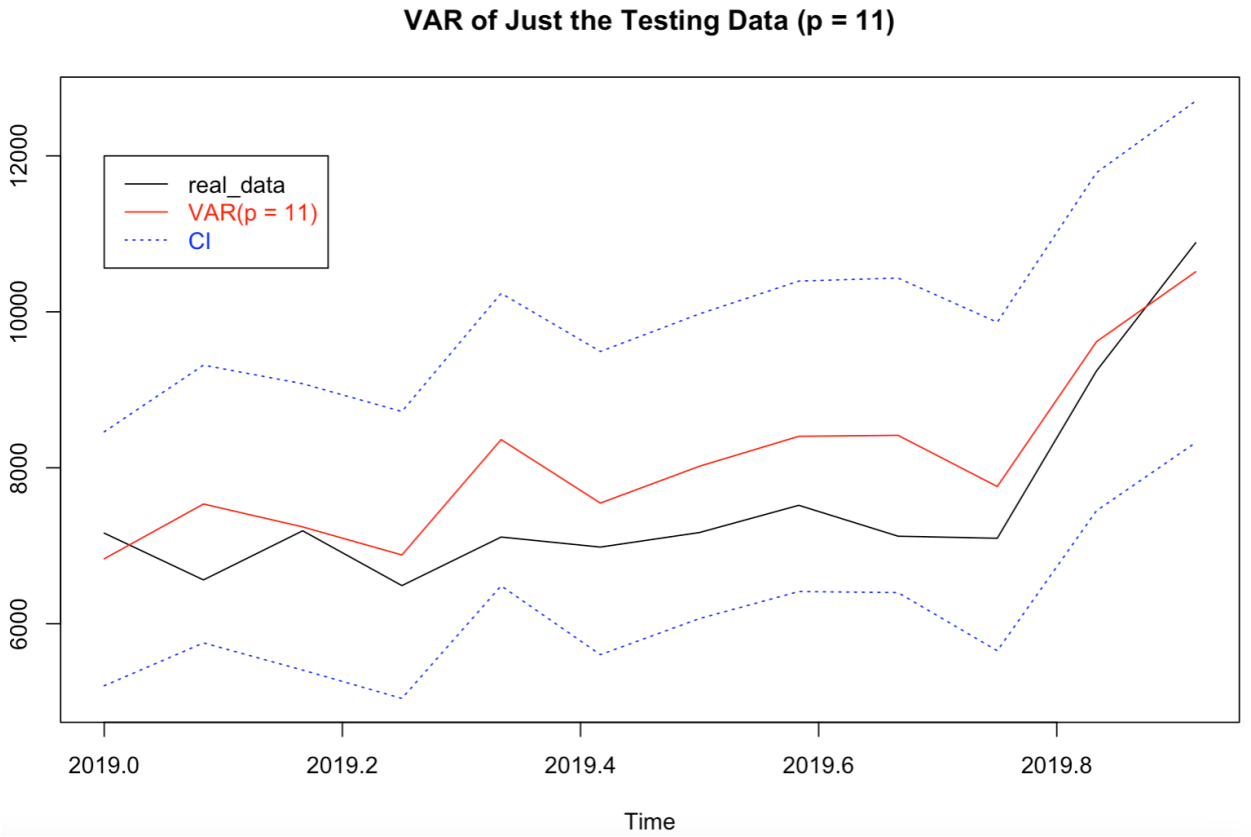

Fig 9.6: Multiple Time Series Plot of Raw Retail Electronic Sales Testing Data and VAR Model Forecast With 95% Prediction Intervals

Our VAR model performs fairly well, however, the model frequently overpredicts the actual value. The 95% confidence interval of our forecast is indicated with the blue, dotted line and it follows the overarching movement of the actual data well. Furthermore, the model achieved an RMSE value of 763.2758. Although not the best result, it is definitely a compelling approach.

X. Forecast Comparison and Final Conclusion

Date	Raw Data Values	ARIMA Modeling Forecast	VAR Forecast	Time Series Regression	Exponential Smoothing	Average Forecast
2019-01	7162	7006.071	6833.065	7907.207	7059.742	7201.521
2019-02	6562	6779.033	7535.368	7491.655	6777.029	7145.771
2019-03	7192	7292.659	7242.002	8960.311	7368.711	7715.921

2019-04	6489	6543.341	6881.852	8702.239	6574.351	7175.446
2019-05	7112	7053.258	8361.517	9211.528	7037.990	7916.073
2019-06	6983	7158.227	7547.278	9116.689	7104.341	7731.634
2019-07	7170	7076.086	8020.477	9263.109	7006.499	7841.543
2019-08	7519	7493.295	8403.767	10228.018	7458.211	8395.823
2019-09	7123	7057.755	8416.277	8975.524	7056.559	7876.529
2019-10	7096	7023.909	7759.948	9129.192	7056.725	7742.443
2019-11	9242	9210.224	9617.172	10221.634	9221.868	9567.725
2019-12	10886	10851.702	10513.665	12061.170	10829.590	11064.032
	RMSE:	**107.9269**	**763.2758**	**1826.47**	**113.7903**	**621.7122**

Best: ARIMA
Worst: Time Series Regression

Table 10.1: Comparative Results of Forecast Methods 2

We can also visualize our different forecasting methods compared to our testing data.

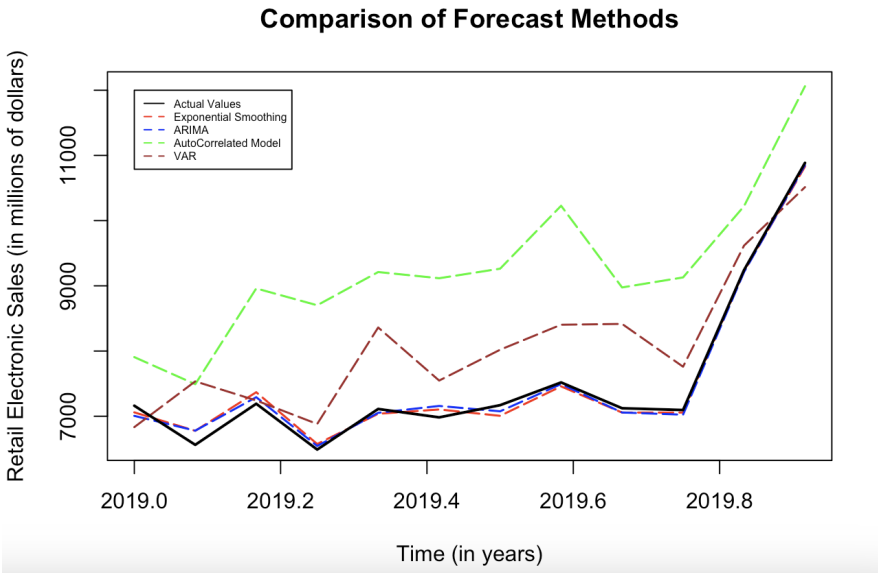

Figure 10.2: Multiple Time Series Plot of Forecasting Methods vs Testing Data

In this portion of the analysis, we introduced three new methods of forecasting the retail electronic sales for 2019. The first method introduced is ARIMA modeling forecasting. For this process, we made the variance of the retail electronic sales data stationary by pre-transforming it through logging the data , made the mean stationary through differencing, and then analyzed the ACF and PACF plots to determine that ARIMA(0,1,1)(1,1,1)_12 is the appropriate model to create a forecast from. Through this process, we created a forecast with RMSE of 107.9269.

For the next method of analysis, we introduced a time series regression method of forecasting. In this process, we wanted to forecast the retail electronic sales through the retail hobby and furniture sales. We started by pre-transforming the data and then creating a multiple regression model. We found the residuals of this model and then standardized the values. The objective is to make the residuals white noise, so if they aren't white noise at this point we must make the residuals stationary. Because the residuals were not white noise, our next step was differencing the data and then plotting the ACF and PACF of this modified data. The residuals were still not white noise at this point even after making the residuals stationary, we need to fit the residuals to an ARIMA model and extract the coefficients of the regular component. After putting these coefficients into the gls() function, we took the normalized ACF of the residuals from our gls() model. We observed that the residuals were not white noise, even though we experimented with multiple different ARIMA models. We concluded that the time series regression method was simply not a good method for our data due to the high amount of shared seasonality between all three of the time series. This showed as this methodology produced the worst RMSE out of all of the forecasting methods with an RMSE of 1826.47.

The final model that we have introduced is the Vector Autoregression model (VAR model). This is an unique approach as it analyzes the dynamic relationship between multiple time series variables. Each of the variables is regressed on its own lagged values and the lagged values of all other variables in the system. We first find the order of the model through the cross correlation function. We achieve this through looking at the significant vertical points and counting the amount of lags. Through the process, we generate the impulse response function for the three variables. Next, we aim to find the equation of each variable. When finding the coefficients for each term, there are certain terms that are significant and some that are insignificant. We look at where the earliest lag occurs and measure which predictor is the lead indicator and further assess if there is interdependence with each of the variables. We finally plotted our prediction that we generated through the model and compared it with the actual values of the dependent variable. Through this process, we created a forecast with RMSE of 763.2758.

In conclusion, when attempting to forecast the retail electronic sales in 2019, we found that the best forecasting method was the ARIMA modeling method with exponential smoothing being a close second. We determined this because the RMSE of the ARIMA forecast was better than all of the other forecasts. The worst modeling method was the Time Series Regression method. This

was because that method did not adequately represent the extreme seasonality that was present in the dependent variable, as well as the two independent variables. Due to this, the RMSE was clearly the worst in comparison to all of the other forecasts. Finally, the VAR model had an RMSE value that was between the Exponential Smoothing model and the Time Series Regression model. This means that this forecasting method performed relatively well; however, it is still outperformed by other forecasting methods for our time series data and thus, we should stick to using the ARIMA model for future forecasting.

—---------- End of Section 2 —----------

XI. Forecast Package Auto Arima

Previously, we have introduced and implemented the ARIMA forecasting. In this section, we will dive into forecasting involving auto.arima. Auto.arima is a widely used function in the 'forecast' package of R programming language, which automates the process of selecting the optimal ARIMA (AutoRegressive Integrated Moving Average) model for a given time series data. ARIMA models are widely used for time series forecasting, where the objective is to predict future values of a variable based on its past values.
The traditional method of selecting an ARIMA model involves manual inspection of various models, testing for stationarity, differencing, and model identification through ACF and PACF plots. However, auto.arima function in R automates this process by automatically selecting the best ARIMA model based on a set of criteria, such as AIC (Akaike Information Criteria) and BIC (Bayesian Information Criteria). This saves considerable time and effort in selecting the appropriate model, especially for large datasets.

One feature that can provide us with invaluable information is trends. The trend feature captures the overall direction and magnitude of changes in the data over time. Including trend as a feature will allow the model to account for any upward or downward movement in the electronics and appliance store sales over time, which can be caused by factors such as changes in consumer preferences, economic conditions, and technological advancements. Furthermore, the electronics data had a slight upwards trend before 2008 with a sharp dip around 2008-2009 and then having another upwards trend afterwards. We aim to capture this through the “trend” feature.

Another may be the seasonal or lag = 12 as this is a monthly data. Seasonality is the presence of regular, repeated patterns of variation in the data over time. Inclusion of seasonality features in the model will allow us to capture any systematic patterns of variation that occur within a year or a month. As electronics and appliance stores often have peak sales during certain times of the year, seasonality can be an important feature to include in the model. Furthermore, the electronic data has strong seasonality occurring during the end of the year holiday season. We aim to

capture this notion of seasonality through the features “month” that lists the corresponding month as well as “lag12” which is the corresponding 12 lagged value.
Incorporating the “month” feature as a dummy variable and “trend” and “lag12” as a numeric value, we construct the model using the Auto.ARIMA forecast. The model yields an ARIMA of p = 1 and q = 1. This p and q value are significantly different from what we achieved when we manually calculated the coefficients for AR and MA because we are considering other features as well.

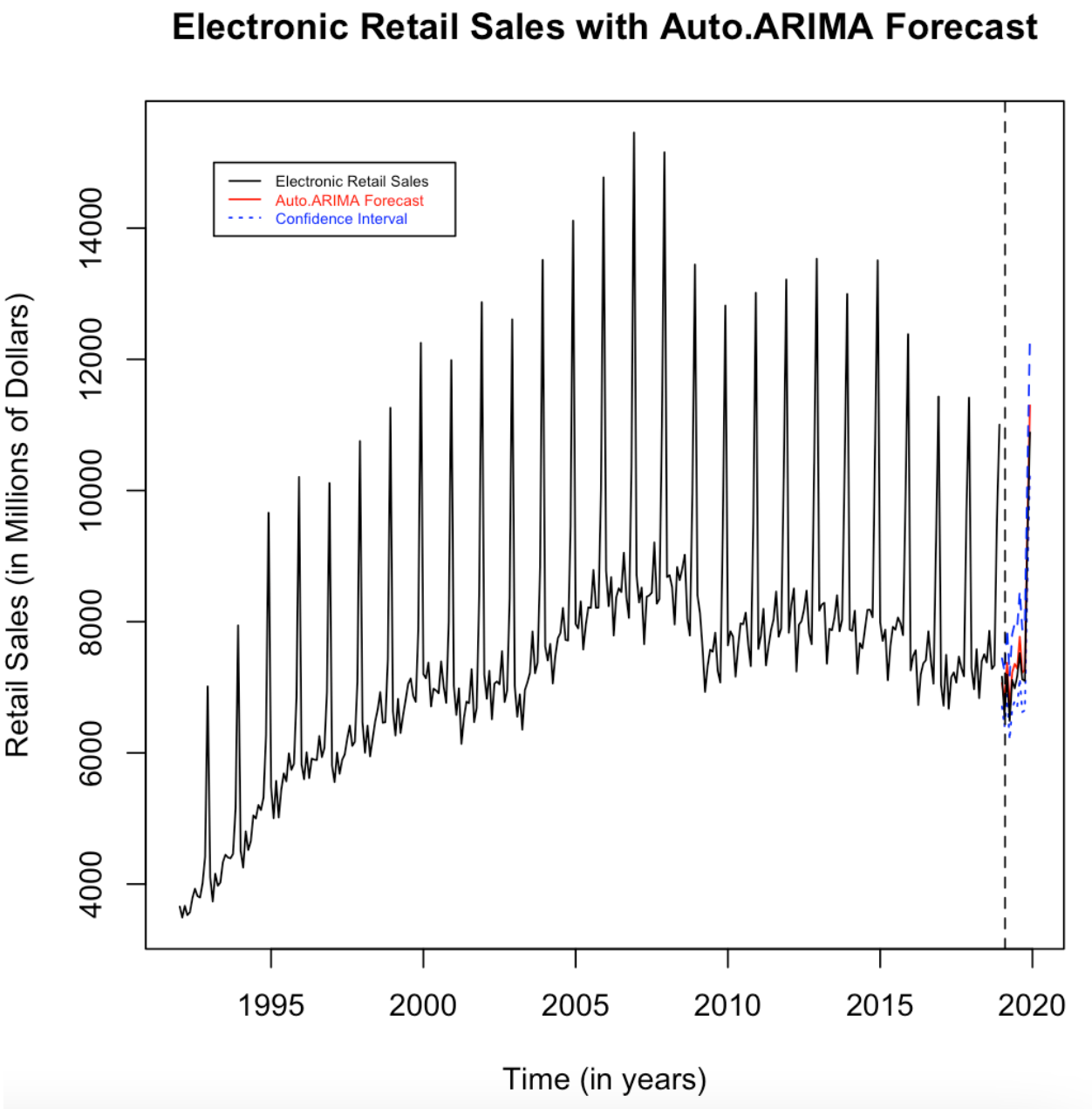

Fig 11.1: Time Series Plot of Raw Retail Electronic Sales With Auto.Arima Forecast

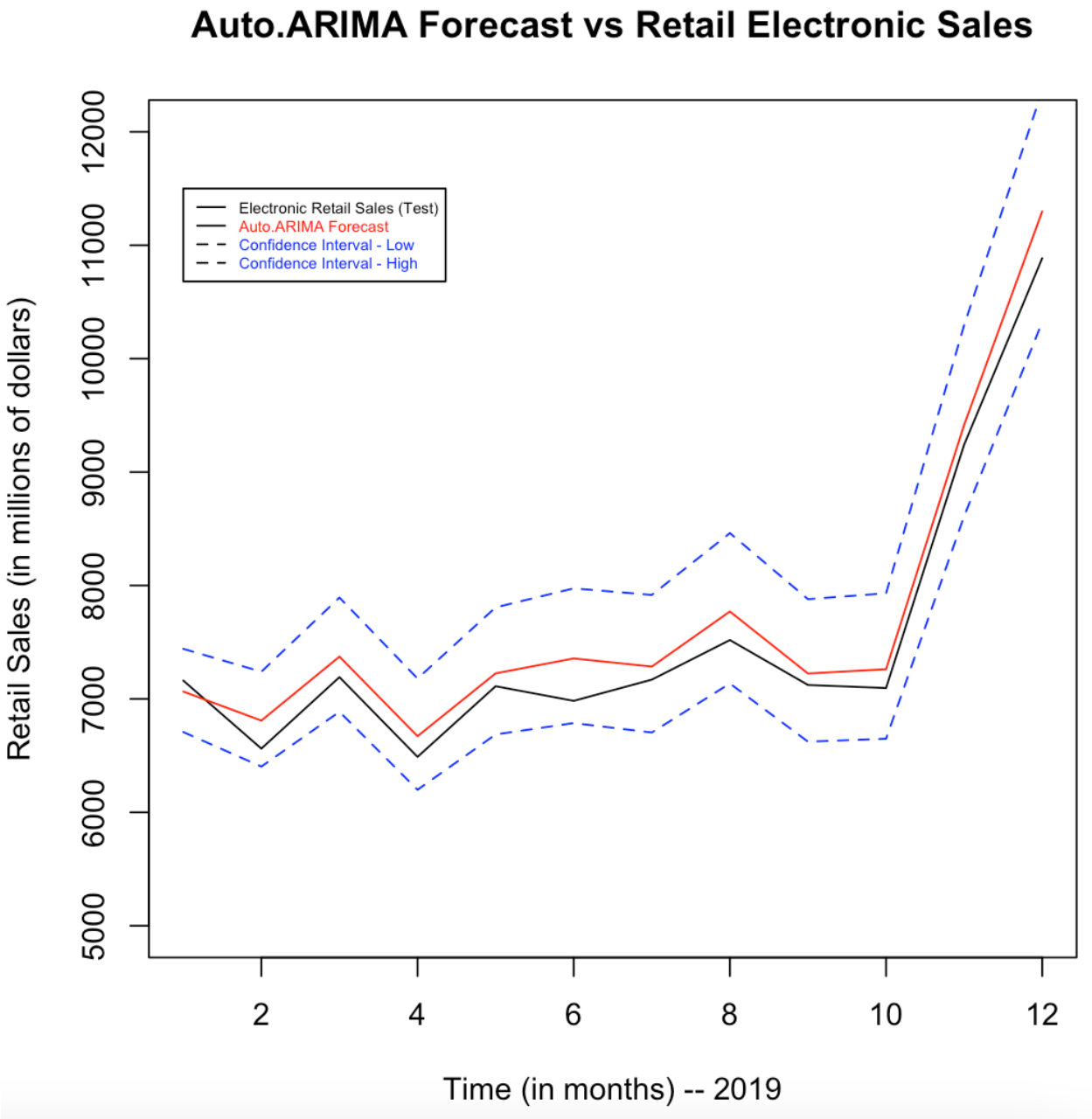

Fig 11.2: Close Up Time Series Plot with Auto.Arima Forecast

As seen in fig 11.1 and fig 11.2, the forecast tends to follow the actual value pretty well. We yield an RMSE score of 0.0284439, however notice that the forecast almost always overpredicts the actual value.

In conclusion, the Auto.ARIMA function is a powerful tool that saves time and effort in selecting an optimal ARIMA model for time series forecasting. By automating the model selection process, it allows researchers to focus on analyzing and interpreting the results of the forecasting. In this study, we applied the Auto.ARIMA function to predict retail electronic sales

using trend and seasonality features. The model performed well in capturing the overall trend and seasonality patterns, with a relatively low RMSE score. However, it is important to note that the forecast tended to overpredict the actual value. Overall, the Auto.ARIMA function proved to be a valuable tool for time series forecasting in this study.

—---------- End of Section 3 —----------

XII. Machine Learning Models

Traditional time series forecasting typically involves statistical methods such as exponential smoothing or ARIMA models, which are based on analyzing the historical data and modeling the patterns and relationships between the observations to make future predictions. These methods are typically parametric and require specific assumptions about the underlying data distribution. By leveraging advanced algorithms and statistical models, machine learning methods can extract meaningful patterns and relationships from time-based data, making it possible to accurately predict future values and trends.

We utilized the dependent variable of Retail Electronic Sales. Ultimately, we look to predict the monthly sales for the year 2019 and will compare various models to see which model's prediction is closest to the actual values. We assess the time plot of the data to see if there is any trend or seasonality.

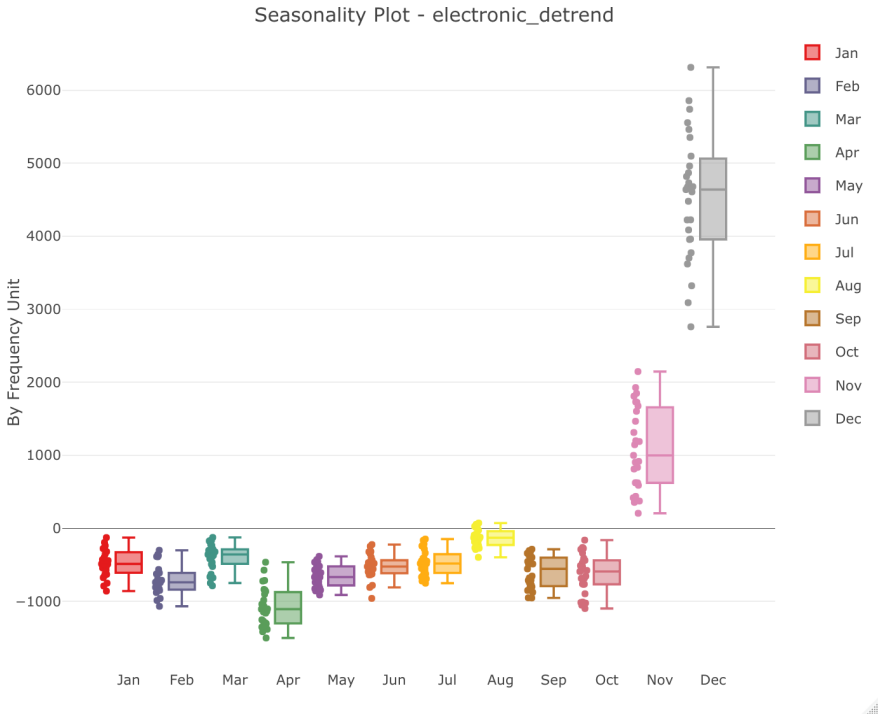

Fig 12.1: Seasonal Boxplot of Retail Electronic Sales

Looking at fig 12.1, we see that there is a clear seasonality. The retail sales drastically increase towards the end of the year, which can be attributed to the holiday seasons.

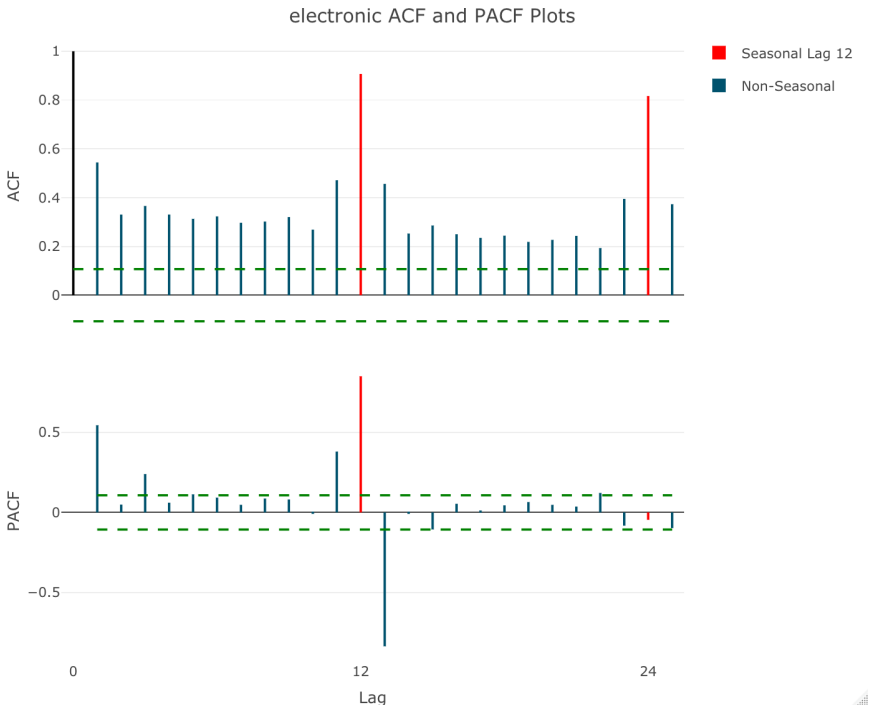

Fig 12.2: ACF and PACF Plot for Retail Electronic Sales

Furthermore, in fig 12.2, we see that this also shows that there is clear seasonality as well. Looking at the red vertical lines, there is a great spike at lag = 12, 24, and more in the ACF plot.

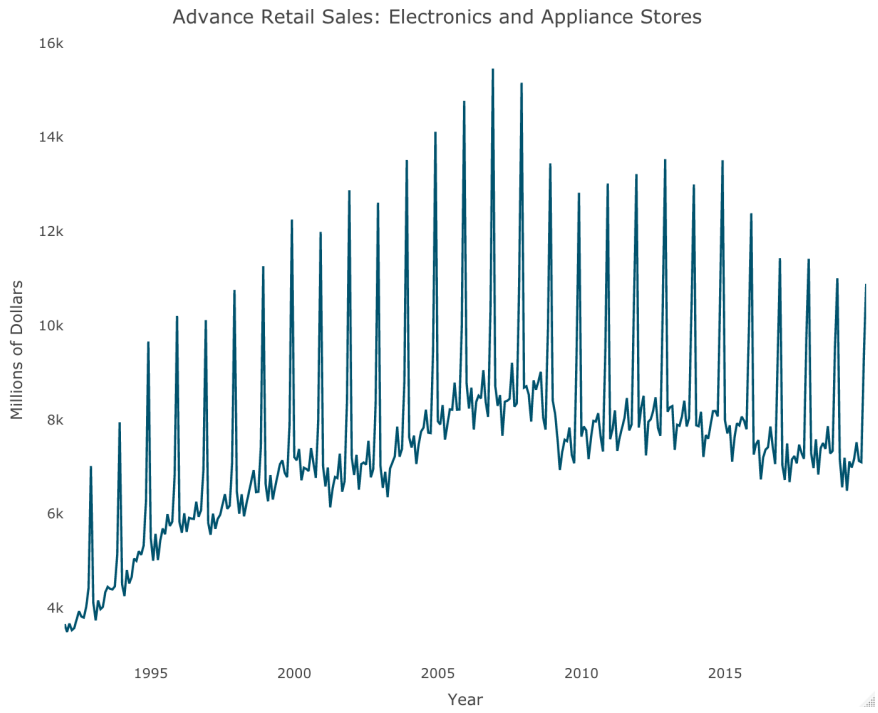

Fig 12.3: Time Plot for Retail Electronic Sales

Looking at the time plot from fig 12.3, we see that there is an upward trend until the end of 2008. However, the trend changed from 2009 and beyond. Thus, we will subset the data from 2009 to 2018 to be fed to the machine learning models and use the 2019 data as the testing data. We will then compare the predicted and actual value for the retail sales during 2019.

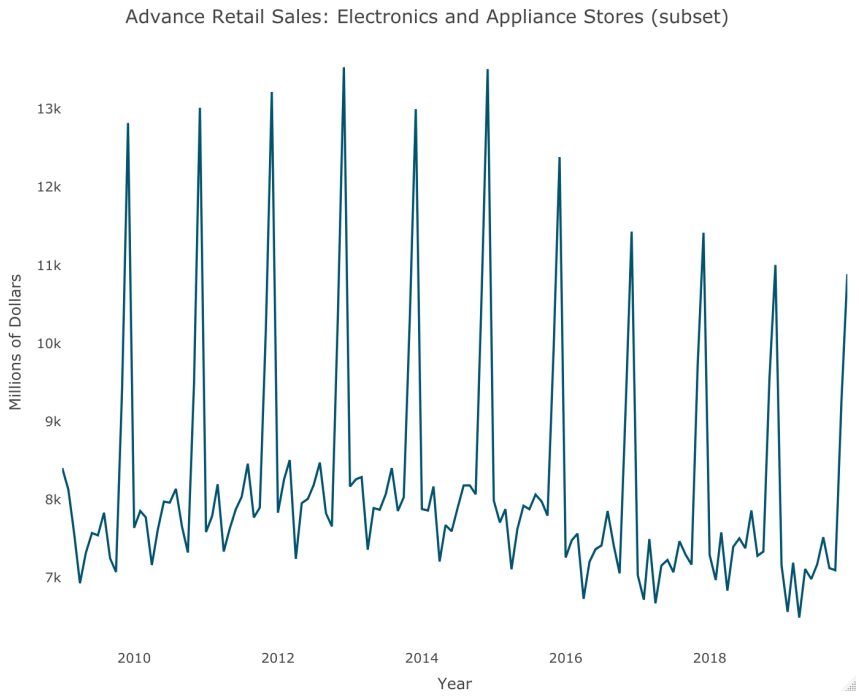

Fig 12.4: Subset of the Time Plot for Retail Electronic Sales

Subsetting our data accordingly, we will obtain data whose time plot looks like fig 12.4. Notice that there is a very meager, relatively constant trend which will later be utilized when constructing the machine learning models.

Next, we aim to find key features that would provide invaluable information to the machine learning models. I introduced four features, month, lag12, lag1, and trend.
First, I incorporated lag12 and month as predictors because we see a strong seasonality (seen through fig 3.1.1, 3.1.2, and 3.1.3) towards the end of the year, and I believe that it will be most appropriate to take this into account when forecasting future values. Lag12 indicates that I am lagging the values for 12 values and the 12 is appropriate in this context as this is a monthly data. The month predictor has integer values from 1 to 12 where this represents the month values correspondingly. Furthermore, I added lag1 as a feature because I recalled that in order to make the data mean stationary, we had to regular difference the seasonal differenced data. I aim to capture this through this predictor. Similar to Lag12, Lag1 indicates that the values are lagged for 1 value. Finally, I added the trend component to take into account the slight trend that exists in our data. As seen in fig 12.4, there is a very small, downwards trend and I want to gain as much information from the data as possible. In conclusion, I am looking into features: month, lag12, lag1, and trend.

As I select the appropriate features and manipulate the data, I will begin to construct models. First, I generate a linear regression model that will be used as a benchmark for the rest of the models.

A. Linear Regression

Linear regression is a simple yet powerful algorithm that has been extensively used in the field of time series forecasting for decades. It involves fitting a linear equation to the historical data to estimate the relationship between the independent and dependent variables.

I constructed a linear model where the dependent variable is the retail electronic sales and the predictor variables are month, lag12, lag1, and trend. I yielded a Mean Absolute Percentage Error (MAPE) of 0.02419736. This will be used as a benchmark and compared against other models.

B. Random Forest

Another form of machine learning in time series is Random Forest.
Unlike a single decision tree, which can easily overfit and perform poorly on new data, the Random Forest algorithm aggregates the predictions of multiple trees to create a more robust and accurate forecast. To utilize Random Forest in time series forecasting, one must carefully decide which features to use and how to split the time series variables. The algorithm generates a training set for each tree using bootstrap sampling of the raw data, and the prediction of all trees in the forest is averaged to create the final output. We implement the random forest machine learning algorithm to predict the retail sales of electronics.

I implemented this model through utilizing 500 trees and 5 folder cross validation. Additionally, I utilized a stopping tolerance of 0.0001 and seed of 7171. The model has performed very mediocrely and can be showcased in the figure below.

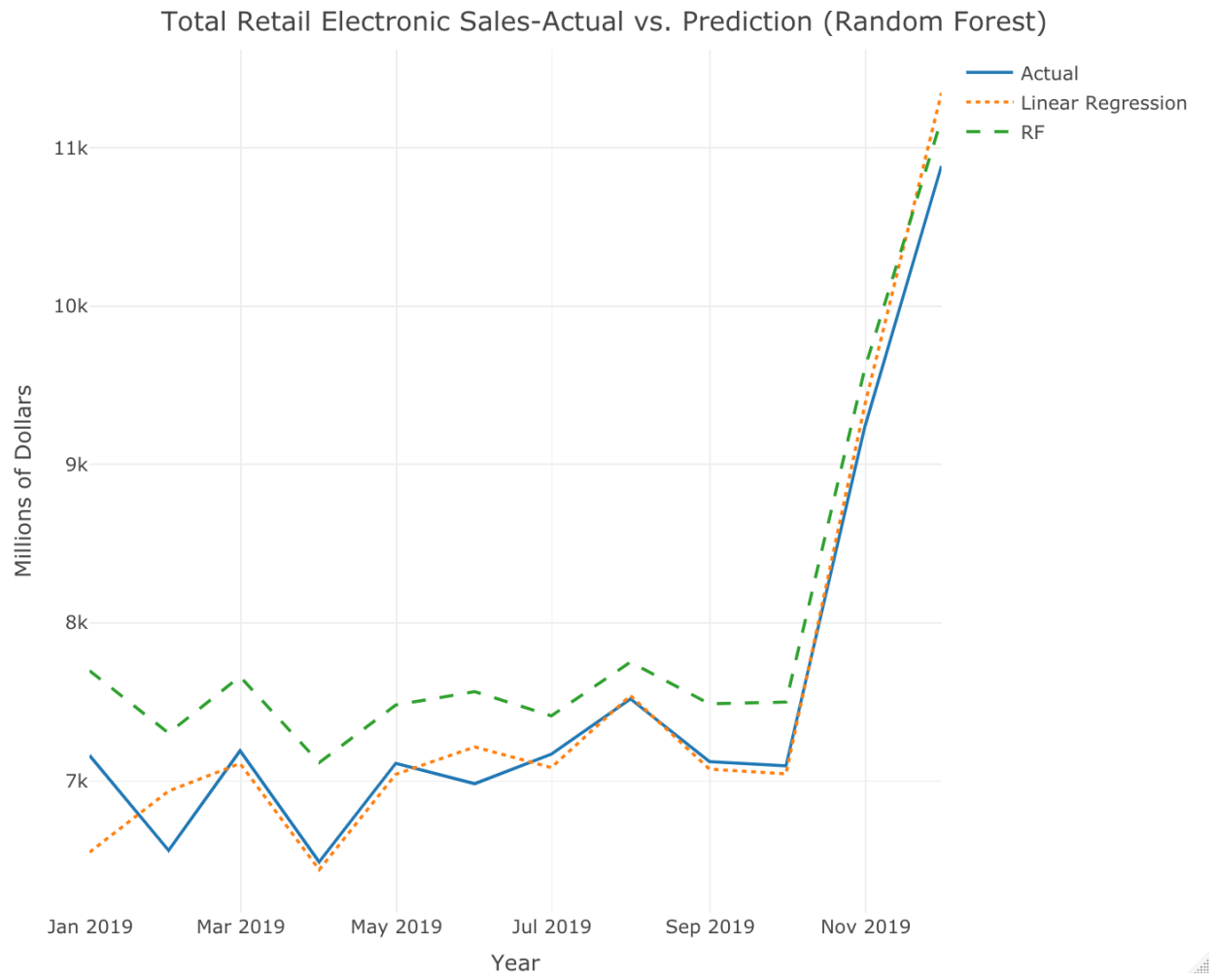

Fig 12.5: Plot of Actual Retail Electronic Sales vs Linear Regression Forecast vs Random Forest Forecast

From fig 12.5, we see that the random forest over-forecasts the electronic sales. The above figure also emphasizes how well the linear regression model captures the actual value. Upon calculating the MAPE, the random forest model yielded a value of 0.06039187 which is significantly greater than the benchmark.

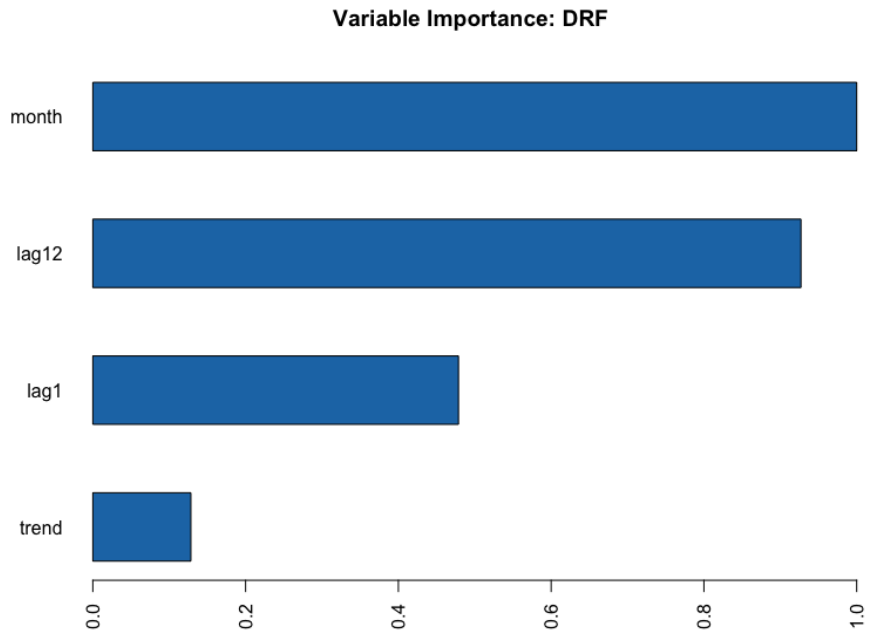

Fig 12.6: Variance Importance Plot Random Forest

To evaluate the importance of the predictors, we look at the variance importance plot. From fig 12.6, we can emphasize the fact that seasonality plays a crucial role in this data. Month and lag12 are both the leading predictors when considering the random forest model.

C. Gradient Boosting

As we delve deeper into the world of machine learning forecasting, another algorithm that has gained significant popularity in recent years is Gradient Boosting (GB). This technique involves combining many single models built sequentially into a single composite model, similar to the Random Forest algorithm. However, unlike the Random Forest algorithm, Gradient Boosting builds models sequentially, with each new model correcting the mistakes of the previous model by trying to predict the residuals of earlier ones.

I implement the gradient boosting model with the same input used in random forest. I obtained a 0.03984699 which is slightly better than the random forest model, but not quite as good as the linear regression model.

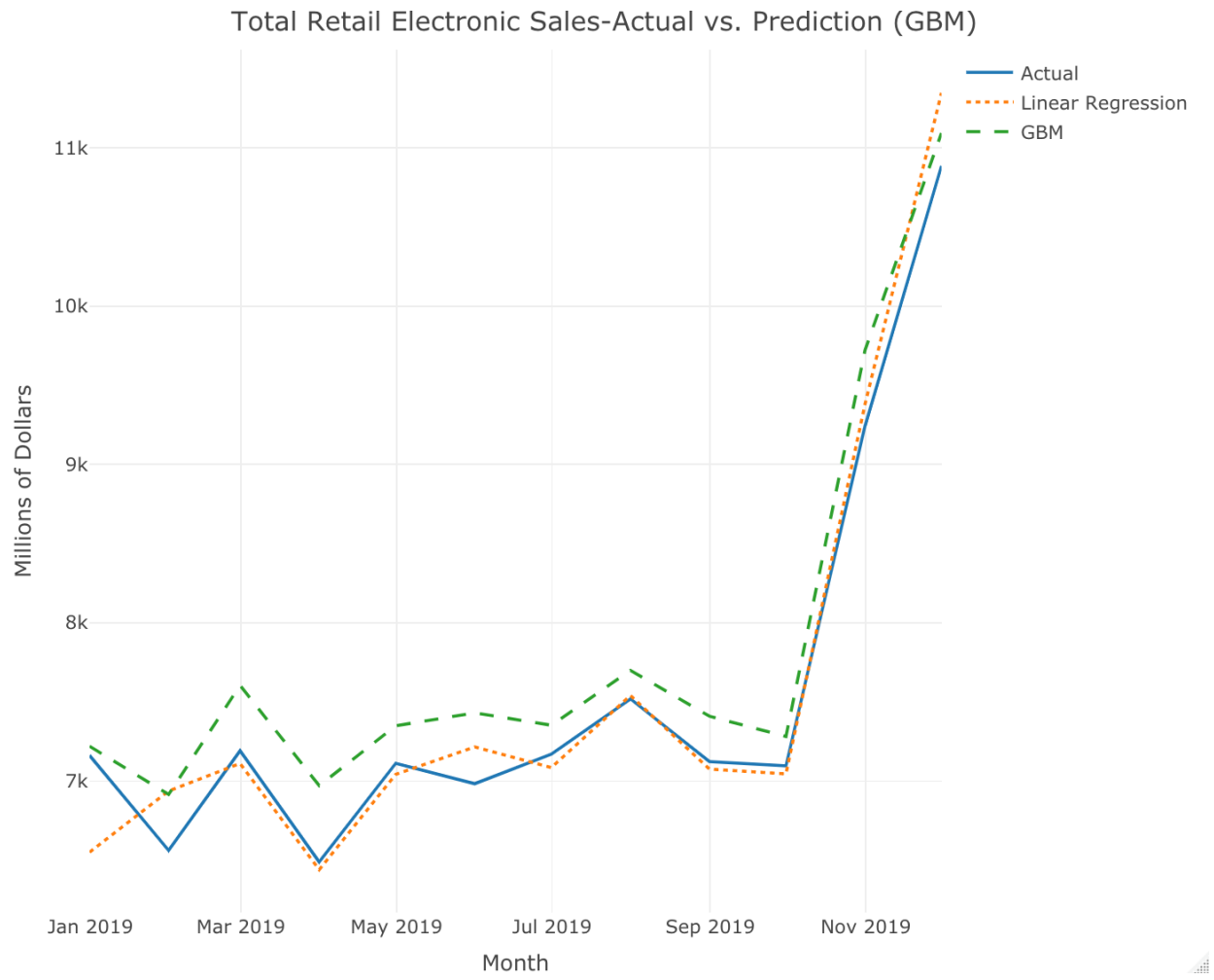

Fig 12.7: Plot of Actual Retail Electronic Sales vs Linear Regression Forecast vs Gradient Boosting

In fig 12.7, we see that again, the gradient boosting model over-forecasts the actual value. However, the magnitude of the over forecast is not as great as the random forest model.

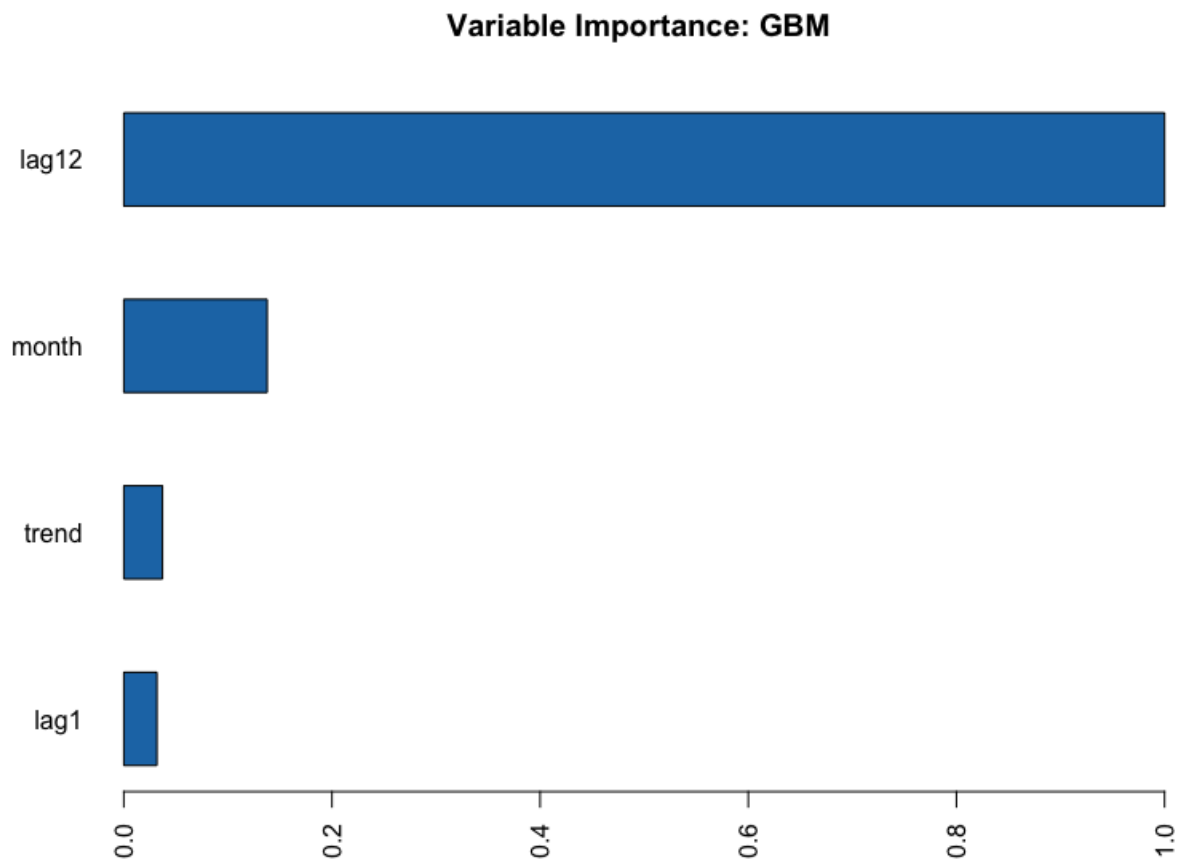

Fig 12.8: Variance Importance Plot Gradient Boosting

Additionally, in fig 12.8, I see that the bar graph significantly shifts compared to fig 12.6. This difference can be attributed to the difference in how the models are constructed. However, overall, we can still see that seasonality plays a very important role as a predictor.

D. Prophet Package

As we continue our exploration of time series forecasting methods, another promising technique that has emerged is the Prophet Package in R. While the previous sections have focused on machine learning-based forecasting methods, Prophet is a time series forecasting model developed by Facebook's Core Data Science team. Prophet is designed to be highly intuitive and easy to use, making it accessible to non-experts in the field of forecasting. In this section, we will provide an overview of the Prophet Package in R and its components. We implement the prophet package to predict the retail sales of electronics and assess the model performance against other models.

I finally implemented the model utilizing the prophet package. This is a slightly different approach compared to the machine learning models that were introduced previously. Overall, I achieved a MAPE value of 0.0285258 which is the smallest compared to the random forest model and the gradient boosting model. However, we must note that it is still greater than the linear regression model.

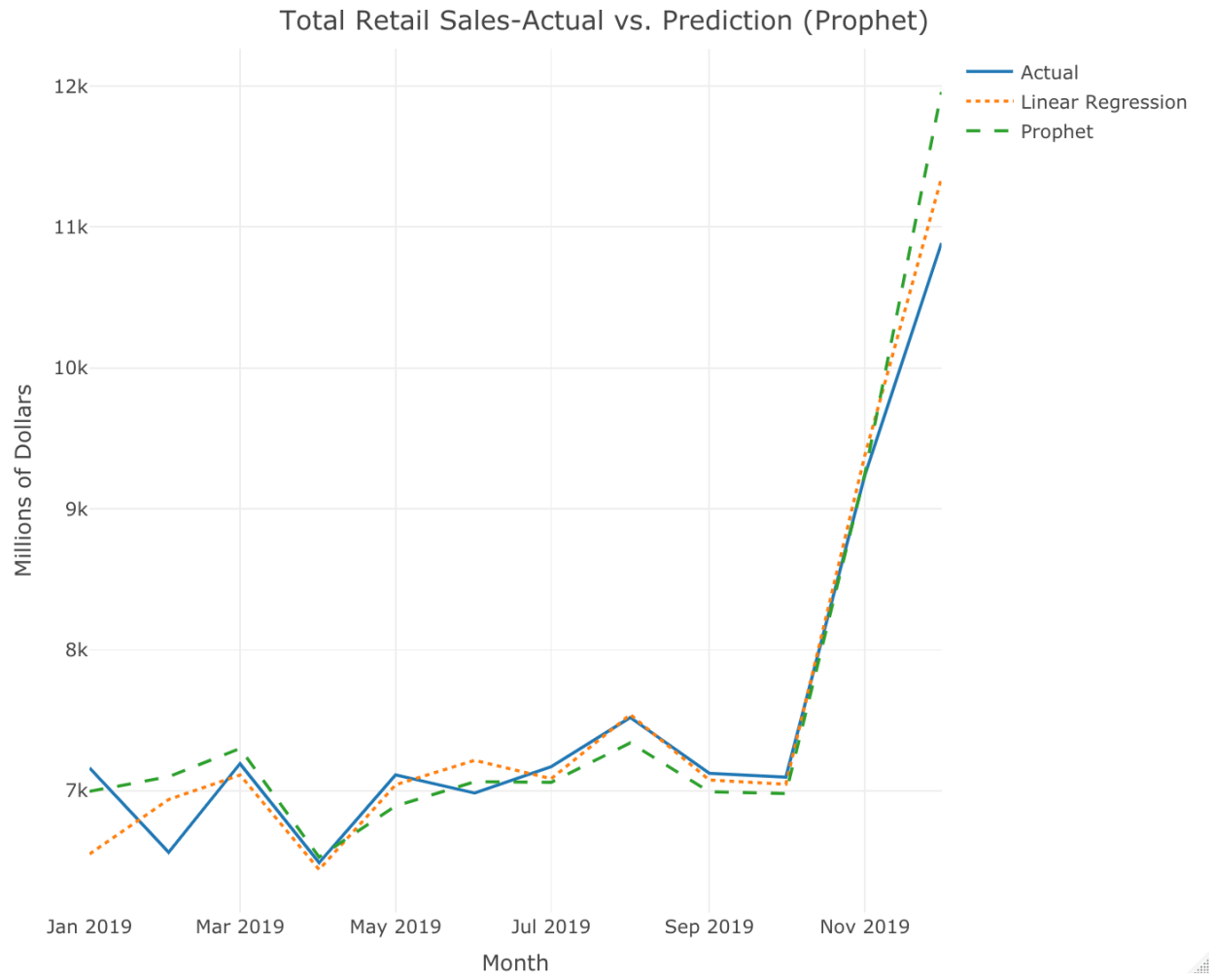

Fig 12.9: Plot of Actual Retail Electronic Sales vs Linear Regression Forecast vs Prophet Forecasting

Looking at fig 12.9, we see that the prophet model does a very good job. The green dashed line follows the actual value very closely. Further, notice that the MAPE for the prophet forecasting is very small.

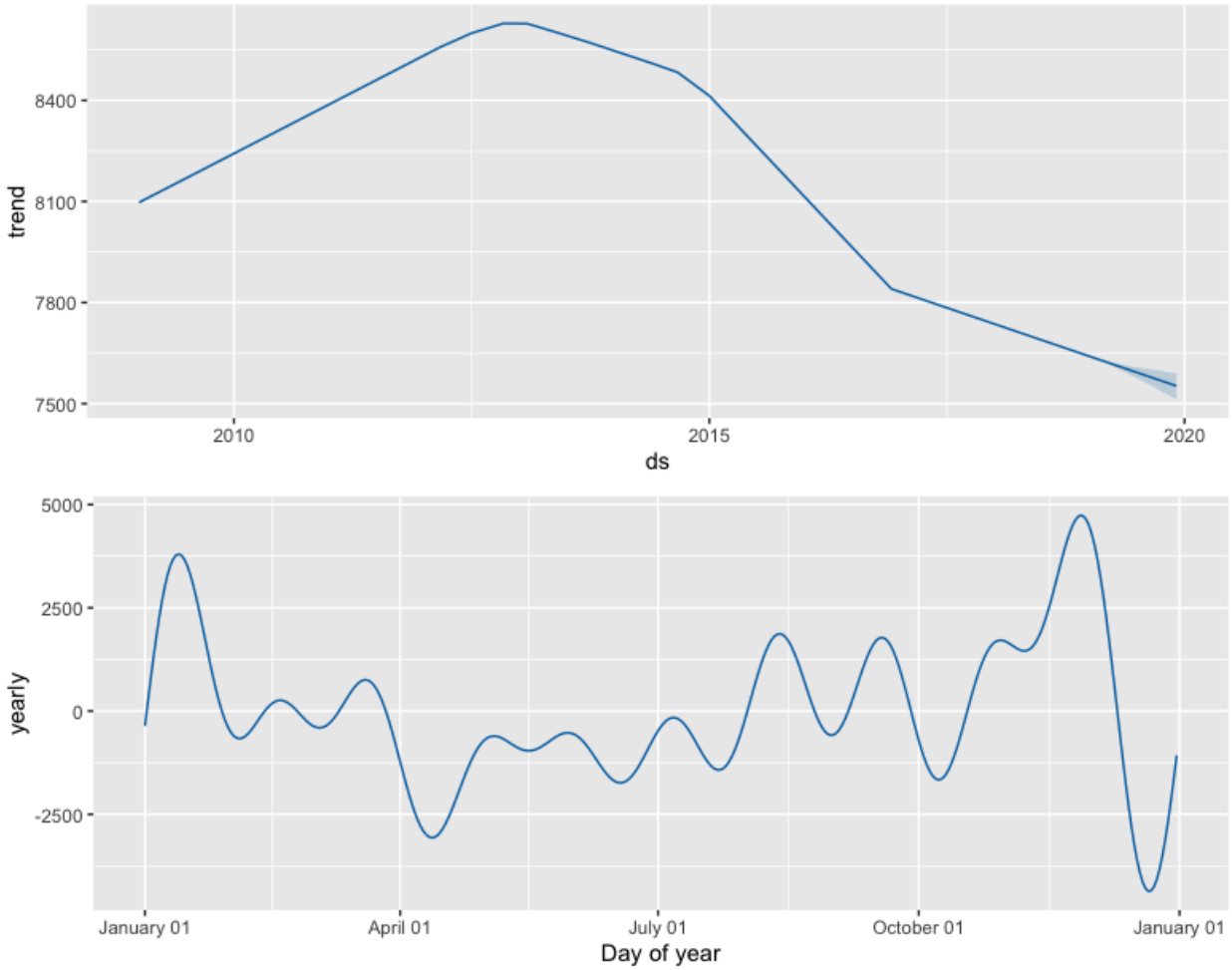

Fig 12.10: Prophet Plot Components

Furthermore, Prophet's unique package provides us with an interesting perspective on our data. The top graph showcases the overall trend as we move along in the year and the bottom graph showcases the seasonality within the year. Note that both graphs seen in fig 12.10 follow what I have previously mentioned regarding seasonality and trend.

XIII. Forecast Comparison and Final Conclusion

Now that we have constructed the models, we evaluate the performance by comparing the RMSE and MAPE values.

Date	Raw Data Values	ARIMA Modeling Forecast	VAR Forecast	Time Series Regression	Exponential Smoothing	Benchmark
2019-01	7162	7006.071	6833.065	7907.207	7059.742	6550.638
2019-02	6562	6779.033	7535.368	7491.655	6777.029	6937.180
2019-03	7192	7292.659	7242.002	8960.311	7368.711	7111.289
2019-04	6489	6543.341	6881.852	8702.239	6574.351	6440.171
2019-05	7112	7053.258	8361.517	9211.528	7037.990	7042.421
2019-06	6983	7158.227	7547.278	9116.689	7104.341	7215.591
2019-07	7170	7076.086	8020.477	9263.109	7006.499	7085.181
2019-08	7519	7493.295	8403.767	10228.018	7458.211	7540.619
2019-09	7123	7057.755	8416.277	8975.524	7056.559	7075.729
2019-10	7096	7023.909	7759.948	9129.192	7056.725	7045.548
2019-11	9242	9210.224	9617.172	10221.634	9221.868	9381.656
2019-12	10886	10851.702	10513.665	12061.170	10829.590	11347.901

	Long-Term RMSE:	**107.9269**	**763.2758**	**1826.47**	**113.7903**	**176.4025**
	Short-Term RMSE:	**188.9672**	**726.5134**	**842.4939**	**168.3659**	**236.1254**

Date	Raw Data Values	Forecast Package auto.arima	Prophet	Random Forest	Gradient Boosting	Average Forecast
2019-01	7162	7065.195	6995.156	7612.629	7220.107	7105.587
2019-02	6562	6808.860	7098.863	7367.748	6915.504	7077.325
2019-03	7192	7372.963	7299.425	7701.429	7604.762	7544.608
2019-04	6489	6671.353	6529.044	7281.929	6970.936	6954.712
2019-05	7112	7225.176	6891.194	7499.671	7349.277	7523.31
2019-06	6983	7356.988	7063.607	7622.629	7430.814	7515.023
2019-07	7170	7285.140	7059.383	7483.900	7352.616	7514.56
2019-08	7519	7770.734	7339.420	7747.250	7699.276	7970.256
2019-09	7123	7223.970	6992.267	7518.486	7409.653	7534.442
2019-10	7096	7262.010	6979.573	7476.171	7281.724	7452.668
2019-11	9242	9416.436	9250.359	9679.687	9722.762	9514.443

2019-12	10886	11298.305	11958.548	11085.145	11094.407	11235.47
	Long-Term RMSE:	**224.3123**	**362.9879**	**474.1249**	**320.7333**	**396.2397**
	Short-Term RMSE:	**187.4981**	**1627.111**	**229.8460**	**399.5248**	**513.241**

XIV. Conclusion

In this study, we explored three popular machine learning methods: random forest, gradient boosting, and the Prophet package for time series forecasting. We applied these methods to a real-world dataset to predict the demand for a specific product.

Our analysis revealed that all three methods provided relatively accurate results. However, each method had its strengths and weaknesses. Random forest is a powerful ensemble method that can handle high-dimensional data and non-linear relationships. It is also robust to outliers and missing values and can capture non-linear relationships. However, it may not perform well when there are complex dependencies between variables. Gradient boosting, on the other hand, is a powerful method that can handle complex relationships between variables. It is also robust to overfitting and can provide feature importance scores. However, it may require more tuning to achieve optimal results and can be computationally expensive. Finally, the Prophet package is a powerful tool for time series forecasting that is introduced by Facebook. It can handle seasonality, trends, and holiday effects and provides interpretable results. However, it may not perform well when there are abrupt changes in the data and may require more data preprocessing.

We can see that when we wish to perform long term forecasting - forecasting that occurs over the course of the year - we should consider models like Arima, exponential smoothing, and even linear regression forecasting. This is because all of the RMSE values are the lowest and the points are relatively close to the actual data points when we ran our tests. In contrast to this, sometimes we want to look for the short term forecasts of time series. In this case, we can look for the best model through the short-term RMSE. This is essentially found by taking the RMSE of just the first two points of the forecast prediction. In this case, Arima and exponential smoothing still dominate as the short-term RMSE are still low, but we can also look into other methods, such as gradient boosting.

—---------- End of Section 4 —----------

XV. Overall Conclusion

In conclusion, this research article provides a thorough time series analysis and forecasting of the performance of three significant retail segments in the US, namely Electronics & Appliance Stores, Sporting Goods, Hobby, Musical Instrument and Book Stores, and Furniture and Home Furnishings Stores. The study utilized time series data and various statistical models to gain insights into the trends and patterns of retail sales for each segment.

The findings of this study have several practical applications, particularly in helping stakeholders in the retail industry make informed decisions. The insights gained from this study can aid in forecasting future trends and identifying areas for improvement in retail operations.

While this study provides valuable insights, it is not without limitations. One limitation is the availability of data, as some relevant data points may not be publicly available. Another limitation is the sample size, as the study only examined three retail segments. Future research can overcome these limitations by expanding the scope of the study to include a broader range of retail segments and exploring additional data sources.

Moving forward, the next steps for researchers in this field include exploring the impact of external factors, such as economic indicators and consumer sentiment, on retail sales. Additionally, future studies can examine the effectiveness of specific retail strategies and interventions aimed at improving sales performance.

Overall, this study provides valuable insights into the performance of important retail segments in the US. The findings of this study can aid in informing decision-making processes and help stakeholders in the retail industry stay competitive in an ever-changing market.