# Introduction

Time series forecasting is one part of machine learning that is often neglected. When we think about machine learning, it is usually about the machine learning from our past experiences or facts we know through data in the computer that is then interpreted by the machine learning algorithm, then the machine learning model is able to predict the outcome given certain features to a certain extent. What happens if our problem has a feature that involves a time component? Time series forecasting is used to solve this kind of problem (Brownlee, 2020).

There are actually many prediction problems that involve a time component, however these problems were mostly neglected because it is this time component that makes time series problems more difficult to handle compared to normal classification and regression problems (Brownlee, 2020).

## Time series

A normal machine learning dataset is actually a collection of observations. For example, a machine learning model that predicts the survivability of Titanic passengers requires a dataset that includes information on each passenger, such as their age, which cabin they lived in, which class they sat in the ship and so on. This information is an observation on the passengers at that point of time, 1912.

Predictions are made for new data when the actual outcome may not be known until some future date. The future is being predicted, but all prior observations are almost always treated equally. This isn’t usually the case in real life as things change as time goes on. Usually in normal machine learning we solve this “concept drift” by fixing the observations at one point of time instead of continuously following the flow of time.

A time series dataset is different from the normal machine learning dataset.

Time series adds an explicit order dependence or relationship between observations: a time dimension. This additional dimension is both a constraint and a structure that provides a source of additional information that makes machine learning more complex. There are two parts of time series, time series analysis and time series forecasting (Brownlee, 2020).

## Time Series Analysis

The primary concern is the analysis of time series, which means to develop models that can describe an observed time series in order to understand the underlying causes. In layman terms, we are trying to find the why behind the time series dataset. This often involves making assumptions about the form of the data and decomposing the time series into constitution components. The result is mathematical models that are able to provide plausible or reasonable descriptions from the sample data. The quality of the descriptive model can be seen by how well it describes all available data and the interpretation it provides to better inform or understand the problem domain (Brownlee, 2020).

## Time Series Forecasting

When predicting the outcome of a time series dataset, it is called time series forecasting. Forecasting involves taking models fit on historical data and using them to predict future observations. Since the future is not known the results is only an estimation of the future based on past events and observations. The quality of a time series forecasting model is determined by its performance or accuracy at predicting the future (Brownlee, 2020).

## Concerns of Forecasting

Before building a forecasting model it is necessary to understand the goal of forecasting the time series dataset. A good suggestion is to start by asking yourself certain questions to get a good idea on the specifics of your predictive modeling problem. Such as:

1. How much data do you have available and are you able to gather it all together?
   1. More data is usually more helpful, offering greater opportunity for exploratory data analysis, model testing and tuning and model fidelity.
2. What is the time horizon of predictions that is required? Short, medium and long term? Shorter time horizons are often easier to predict with higher confidence.
3. Can forecasts be updated frequently over time or must they be made once and remain static? Updating forecasts as new information becomes available results in more and more accurate predictions.
4. At what temporal frequency are forecasts required? Often forecasts can be made at lower or higher frequencies allowing for down-sampling, or up-sampling of data, which can help tune for a better model.

## Examples of Time Series Forecasting

1. Forecasting the closing price of a stock each day
2. Forecasting the birth rate at all hospitals in a city each year
3. Forecasting product sales in units sold each day for a store.
4. Forecasting the number of passengers through a train station each day.
5. Forecasting the average price of petrol each day.

In this assignment, I will be using data on the expenditure on parts from the MVSS department in Vitrox from 2016-2020 and apply time series forecasting. The data comes from an excel file which consists of the parts’ product ID, product description, the monthly expenditure of individual parts of different departments in Vitrox, transaction quantity and others. For this time series, we will be using the monthly parts’ expenditure from the MVSS department only.

A new csv datafile is created by summing up the parts’ expenditure for each month, from 2016-2020. Timestamps are included for each months’ expenditure.

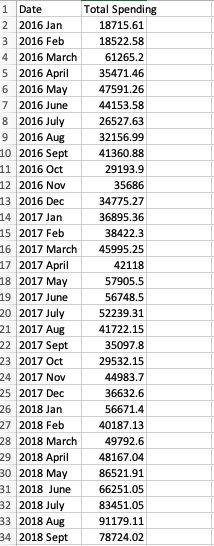


Figure: The datafile

After sorting out the datafile, we can begin programming our time series forecasting model.

# Methodology

## Import Dataset and Describe it

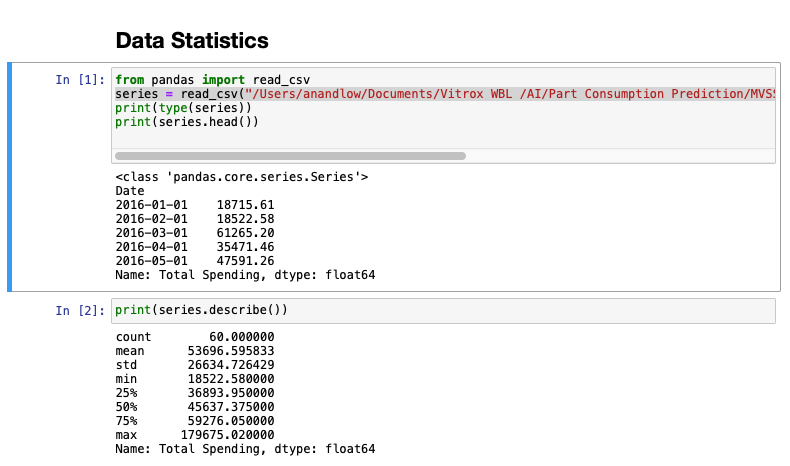


Figure: Data Statistics

The datafile is read using pandas’ read\_csv function. We can see that the program reads the data correctly, even recognising the timestamps in the datafile. There are a total of 60 data points in the dataset. Maximum value is 179675.02 and the minimum value is 18522.58, with mean = 53696.60 . Next, we will use various methods to visualize the data points to get a better understanding of them.

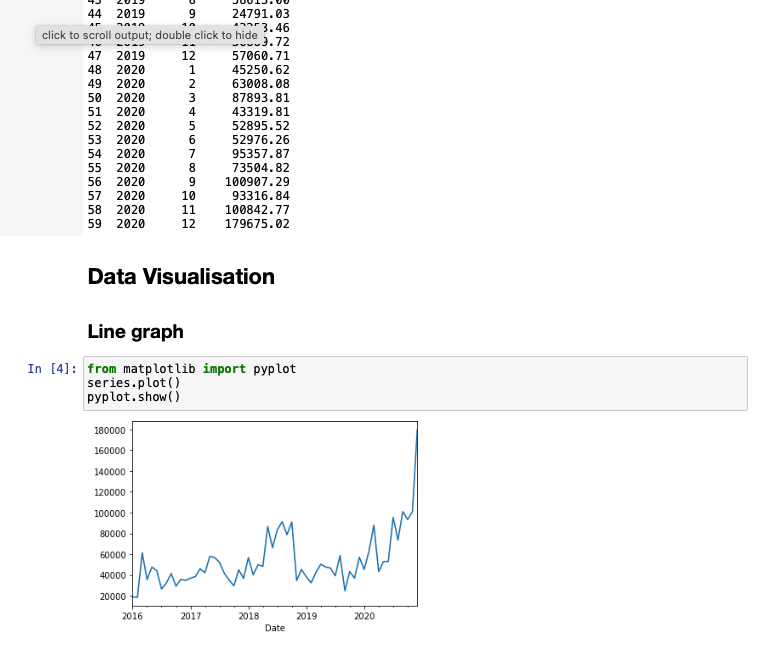
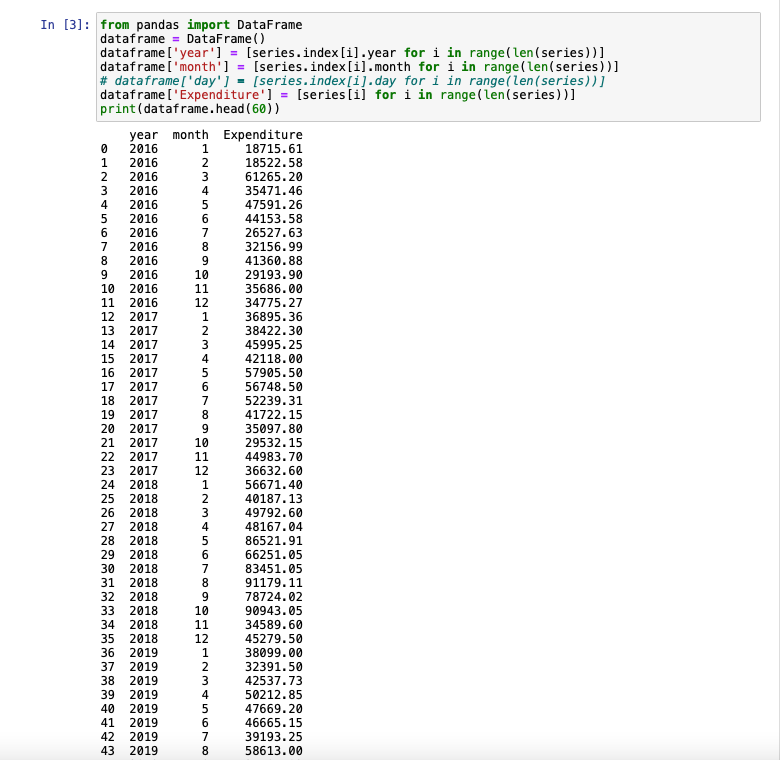


Figure: Data points

## Data Visualization

# 

Figure: Line graphs (whole dataset and year-by-year)

We can see here that for 2016, 2017 and 2019, the parts expenditure ranges from RM20, 000 to RM60, 000. For 2018 and 2020, we can see spikes up to around 90, 000. At the end of 2020 parts’ expenditure increased to a high of around 180, 000.

By comparing the line graphs for each year, we can deduce that for 2018 and 2020 the parts’ expenditure is high, while for 2016, 2017 and 2019 the parts expenditure is normal. This means that in 2018, 2020 Vitrox spent more on parts to build their vision inspection machine, which means more machines are built to be sold. We can deduce that in these two years the sales for MVSS vision inspection machines are high.

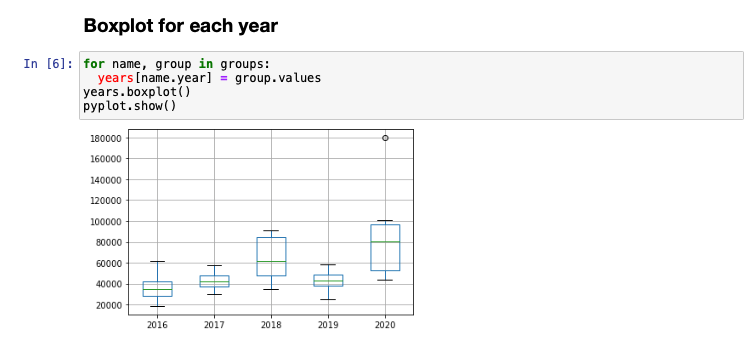


Figure: Boxplot for each year

By using boxplots, we are able to see the range of the data points, median and outliers in parts’ expenditure each year. As we suspected by seeing the line graph, 2018 and 2020 averages higher parts’ expenditure compared to the other 3 years.

## Data Analysis

### Time Series Decomposition

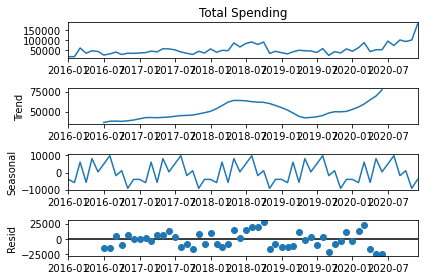


Figure: Time Series Features

Here, the time series dataset is decomposed into three important time components which are features, trend, seasonal and residual error.

* Trend: The increasing or decreasing value in the series.
* Seasonality: The repeating short-term cycle in the series.
* Noise: The random variation in the series.

From the trend, we can see that from 2016 to 2018, the parts’ expenditure increases until it reaches a peak at around June of 2018. From there, the parts’ expenditure decreases until it hits a bottom of around 30, 000 around May of 2019. Then, it increases up to a peak of about 180, 000 at the end of 2020.

From seasonal, we are able to see that the first few months and the last few months have lower expenditure compared to the middle months. There are still some ups and downs but overall the middle months have higher parts’ expenditure.

The residual shows high variability during 2018 and 2020.

### Lag Scatter Plot

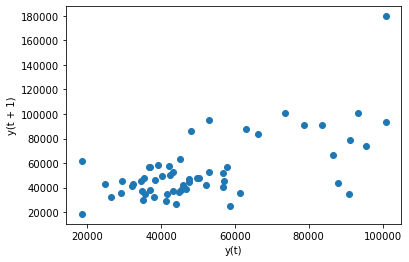


Figure: lag1 Lag Scatter Plot

The previous observations in a time series are called lags. The observation at the previous time step is called lag1. Lag scatter plots explore the relationship between each observation and the lag of that observation. It plots the observation at time t on the x-axis and the observation at the next time step (t + 1) on the y- axis. if the points cluster along a diagonal line it means it has a high correlation relationship between lag and observation. As we can see above the spending variance between months is evident, thus the lag and the observation have a low correlation relationship. We will try different lags in step time to see whether there is a change in correlation between the lag and the observation.

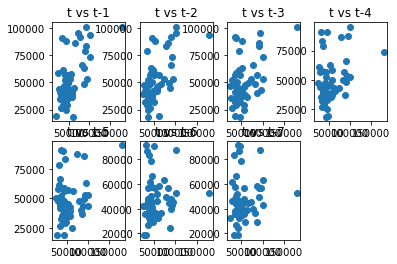


Figure: Lag Scatter Plot for lag1, lag2, lag3, lag4, lag5, lag6 and lag7

We can see that the correlation between lag and observation is not that high, and that correlation decreases going from lag1 to lag7. This means the ideal lag feature to create is lag1, lag2. To be more precise, we can use auto-correlation plots to quantify the correlation between lags and the observations.

### Auto-correlation plots

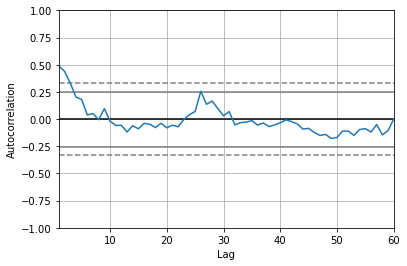


Figure: Auto- Correlation Plot ( Lag vs correlation)

A correlation value calculated between two groups of numbers, such as observations and their lag=1 values, result in a number between -1 and 1. The sign of the number indicates a negative or positive correlation respectively. A value close to zero suggests a weak correlation, whereas a value closer to -1 or 1 indicates a strong correlation. The correlation values, usually called correlation coefficients, can be calculated for each observation and different lag values, then formed into a line plot to better understand how the relationship changes when lag increases. This plot is called an autocorrelation plot. As we can see above, decent positive correlation between lag and observation can be found when lag is around 1. When the lag increases, the correlation between observation and lag gets weakened, until there is no correlation between the two values. So based on this, the lag feature that we will create will use lag1 because it has the best positive correlation with the observations.

## Baseline Forecast Model

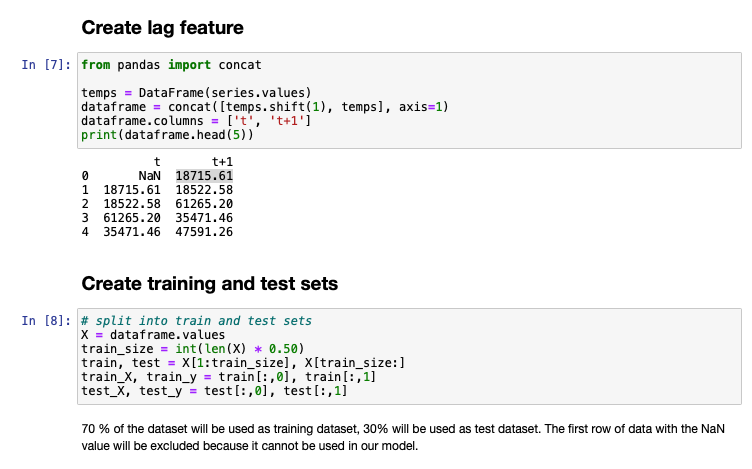


Figure: Data Preparation

First, we have to create a lag feature so that it can be fitted into different kinds of forecasting models. The lag feature we created is lag1, which creates two columns, t and t + 1. Column t consists of the previous time step (lag1), while column t + 1 contains the observation at that point of time. There is a total index of 60. The first row of data with the NaN value will be excluded because it cannot be used in our model.

Next, we will split this dataset into training and test sets. We will split the dataset into half, so the training dataset has 30 data points, and the test dataset has 30 data points. In time series forecasting, the data points cannot be randomized in order, as they are bound to timestamps which have to be followed strictly, unlike machine learning. In layman terms, our training dataset consists of the upper half of the dataset, expenditure for months in 2016, 2017 and 2018, while our test dataset consists of the bottom half of the dataset, expenditure for months in 2018, 2019 and 2020.

For our baseline model, we will have to use a model that is easy to configure and is able to produce fast results to provide a reference for future modifications and improvements. For this reason, we will use the most common baseline method for time series forecasting, which is the persistence algorithm. The persistence algorithm **uses the value at the current time step (t) to predict the expected outcome at the next time step (t + 1)**. The persistence algorithm is naive. It is often called the naive forecast. It assumes nothing about the specifics of the time series problem to which it is applied. This is what makes it so easy to understand and so quick to implement and evaluate.

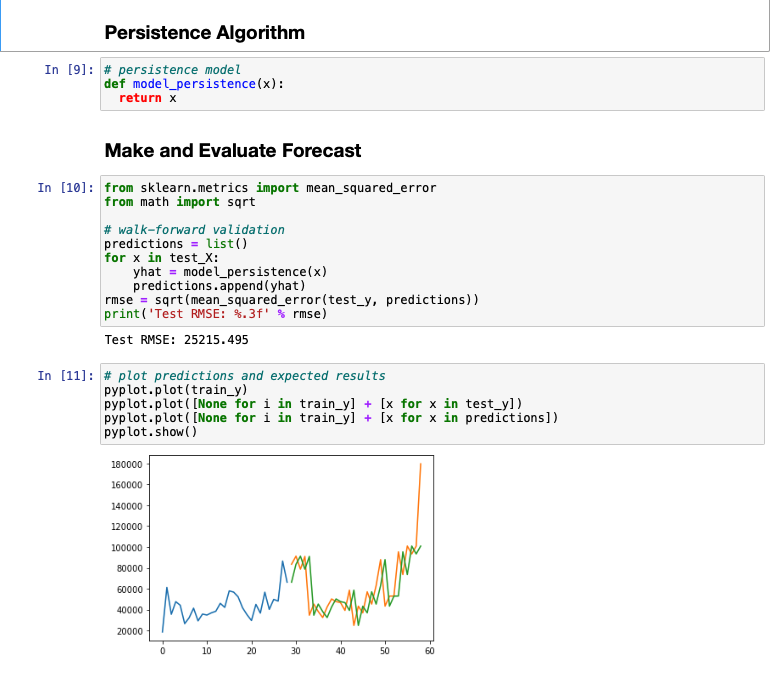


Figure: Persistence Algorithm ( Training Set - Blue, Test Set - Orange, Prediction - Green)

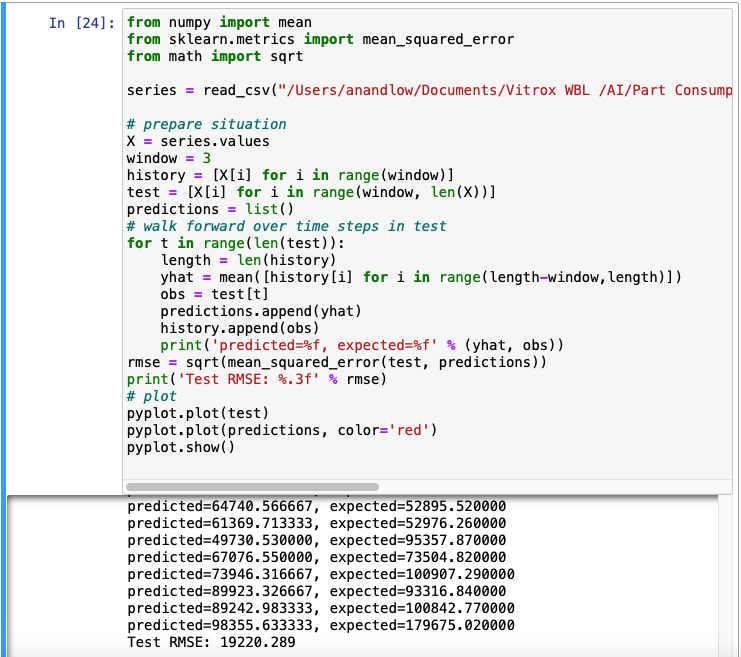
The Persistence algorithm is built, then predictions are made, then the predictions are compared to the actual expenditure. We used root mean squared error here to test the performance of our model. The lower the RMSE the better the performance of the model.

Here, we get a RMSE of 25215.495 from the persistence model, which means the average difference between the prediction and the actual expenditure is 25215.495, which is not too bad, considering our data points range from 18,000 to 180, 000.

## Alternative Models

### Moving Average Naive Model

Moving Average Naive Model calculates a moving average involves creating a new series where the values are composed of the average raw observations in the original time series. This new series of values are used as prediction. It is naive because it does not consider other aspects of the time series.



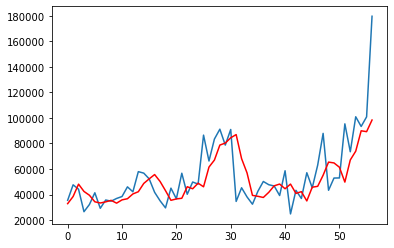


Figure: Moving Average Naive Prediction Model ( Blue- Actual expenditure, Red - Prediction)

For this, I used the average of three observations to form a new observation. Therefore, I set the window width = 3. The new average observations are used as predictions. The final test RMSE is 19220 which is better than the persistence method. However due to it being naive, it does not take into consideration the trends, seasonality and other factors into it. And it certainly does not model the underlying factors which we want for a good prediction model.

Therefore, we will try the next model, ARIMA.

### ARIMA

ARIMA model is a popular and widely used statistical method for time series forecasting. ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a class of models that captures a suite of different standard temporal structures in time series data. It provides a simple yet powerful method for making skillful time series forecasts.

* AR: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.
* I: Integrated. The use of differencing of raw observations to make the time series stationary.
* MA: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

These components are specified in the model as a parameter. Usually ARIMA is configured as ARIMA(p, d, q).

* p: The number of lag observations included in the model, also called the lag order.
* d: The number of times that the raw observations are differenced, also known as the degree of differencing.
* q: The size of the moving average window

A linear regression model is constructed using the specified number and type of terms, then the data is prepared by a degree of differencing in order to make it stationary (by removing seasonal structures and trends). If any of the components is not needed, a value of 0 can be used for that parameter. This way, the ARIMA model can be configured to perform the function of ARMA, AR, I or MA models.

AUsing an ARIMA model for a time series assumes that the underlying process that generated the observations is an ARIMA process.

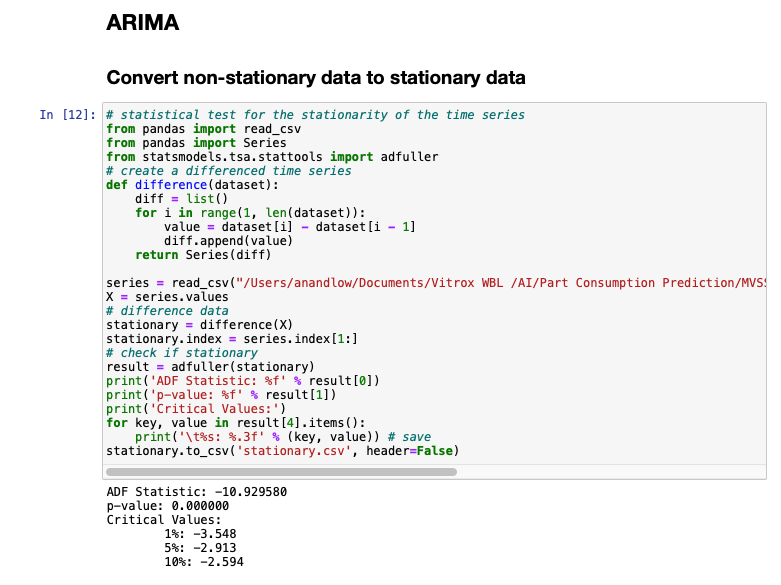


Figure: Convert non- stationary data to stationary data

The results show that the test statistic value -10.929580 is smaller than the critical value at 5% of -2.913. This suggests that we can reject the null hypothesis with a significance level of less than 5% (i.e. a low probability that the result is a statistical fluke). Rejecting the null hypothesis means that the process has no unit root, and in turn that the 1-lag differenced time series is stationary or does not have time-dependent structure ( Brownlee, 2020). By looking at our time series decomposition, our data points clearly have a trend going on which needs to be removed so that our time series is stationary.

This suggests that at least one level of differencing is required. The d parameter in our ARIMA model should at least be a value of 1.

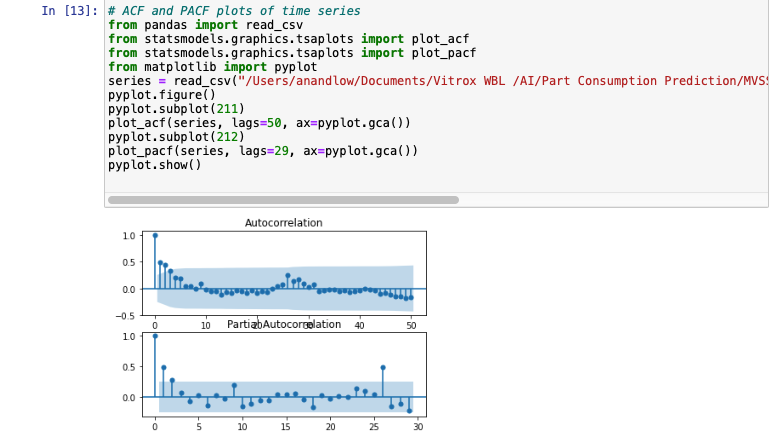


Figure: Auto-correlation function and partial auto-correlation function to determine p and q for ARIMA model

ACF is an auto-correlation function which gives us values of auto-correlation of observations with its lagged values. We plot these values along with the confidence band. To put it simply, it describes how well the present value of the series is related with its past values (lagged values). ACF also considered time series components like trend, seasonality, cyclic and residual.

PACF is a partial auto-correlation function. It finds correlation of the residual ( remains after removing the effects which are already explained by the earlier lags) with the next lag value. This models any hidden information in the relationship between residual and the next lag.

- The ACF shows a significant lag for 3 months.

- The PACF shows a significant lag for roughly 2 months.

- Both the ACF and PACF show a drop-off at the same point, perhaps suggesting a mix of AR and MA.

Therefore, a good starting point for p and q values are 3 and 2 respectively.

This quick analysis suggests an ARIMA(3,1,2) on the raw data may be a good starting point. Experimentation shows that this configuration of ARIMA does not converge and results in errors by the underlying library, as do similarly large AR values. Some experimentation shows that the model does not appear to be stable, with non-zero AR and MA orders defined at the same time. The model can be simplified to ARIMA(0,1,2).

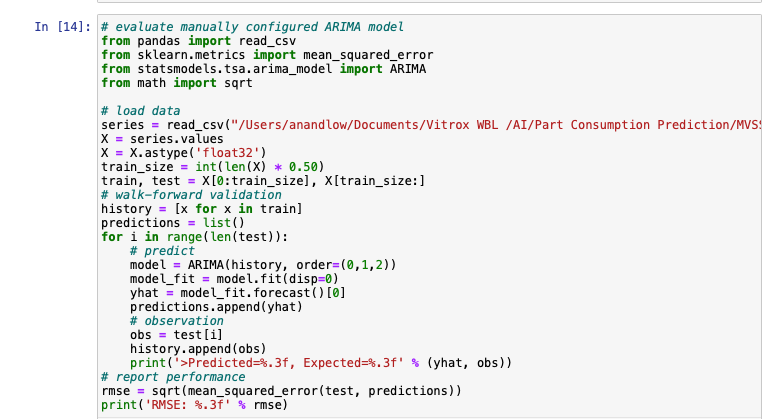
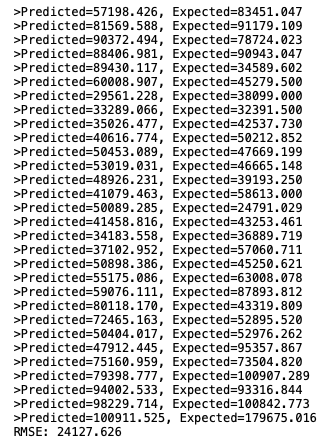


Figure: Manually configured ARIMA model



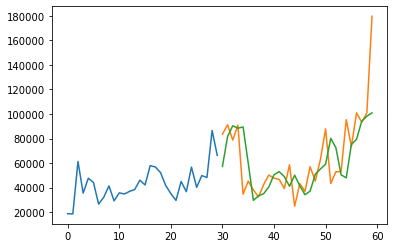


Figure: Manually configured ARIMA model results (Blue- Training, Orange - Test, Green - Prediction)

The RMSE improved considerably compared to our baseline persistence model. The root mean squared error is decreased from 25215.495 to 24127.626. We can get further improved results if we configured the ARIMA model better.

Grid Search ARIMA Hyperparameters

we will search values of p, d, and q for combinations that **do not result in error**, and find the combination that results in the best performance. We will use a grid search to explore all combinations in a subset of integer values. We will search all combinations of the following parameters:

- p: 0 to 12

- d: 0 to 3

- q: 0 to 12

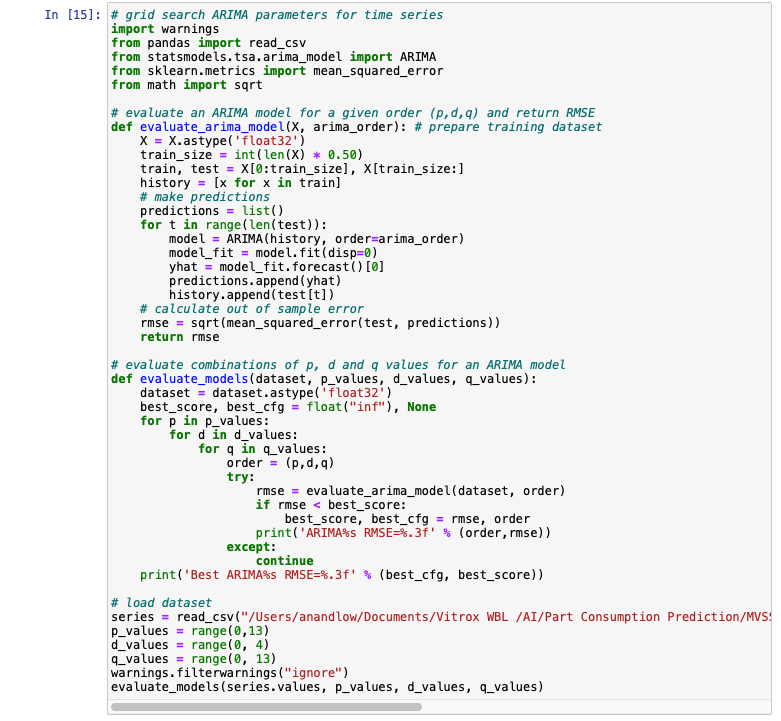


Figure: Code for Grid Search ARIMA parameters for time series

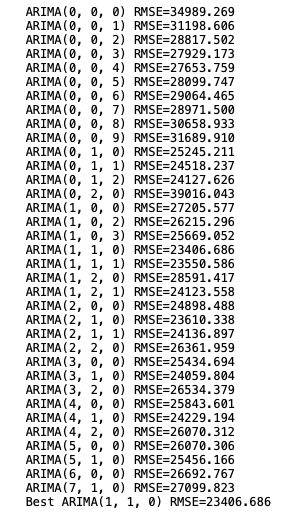


Figure: ARIMA parameter search results

From here, we can see that ARIMA(1,1,0) produces the lowest RMSE, which means that the dataset works best using the ARI model.

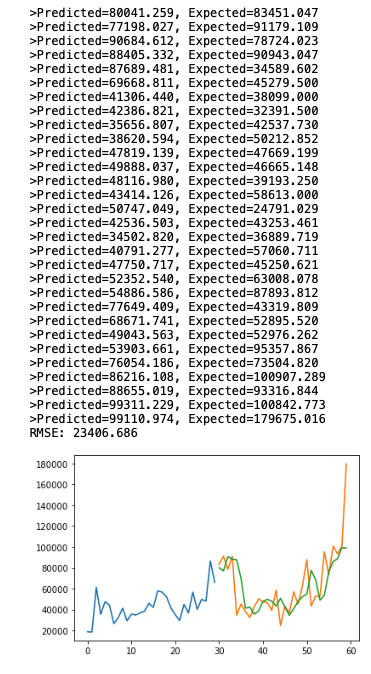


Figure: ARI model results

Outliers such as the parts’ expenditure in December 2020 is difficult to be predicted by any of the models tried so far, which increases the root mean square error by a wide margin.

# Part 2 Top 30 common parts consumption - Multi Time Series Forecasting

## Results

### Data Statistics

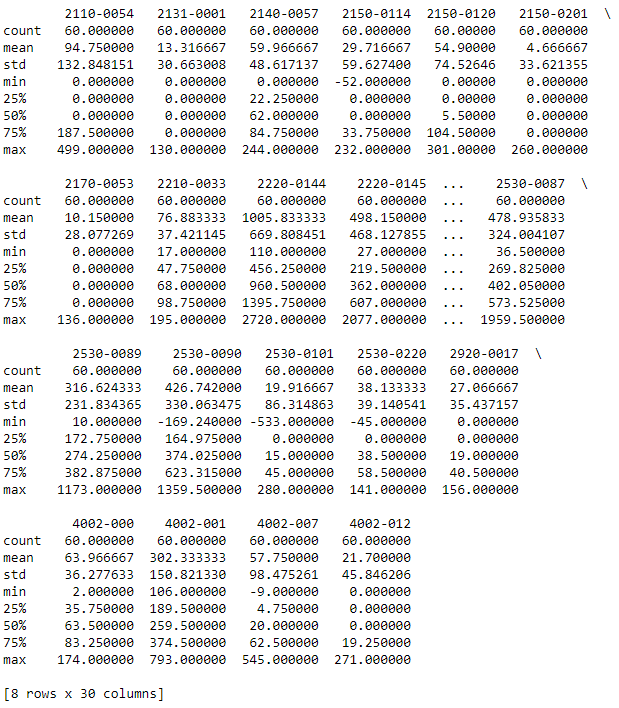
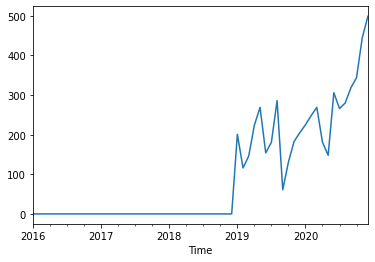


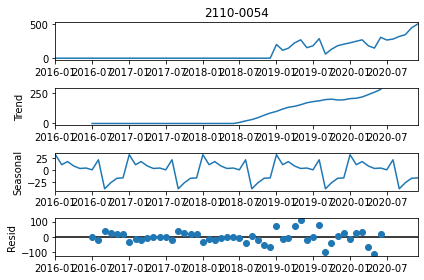
Figure: Data Statistics of some of the parts

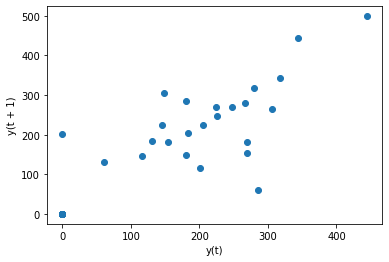
### Parts

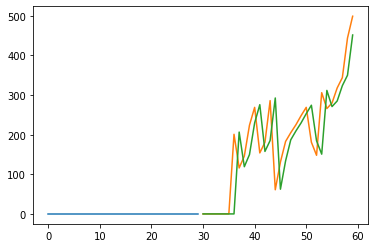
#### Part ID: 2110-0054

#### Part Desc: HDD SATA3.0 1TB 64MB Cache





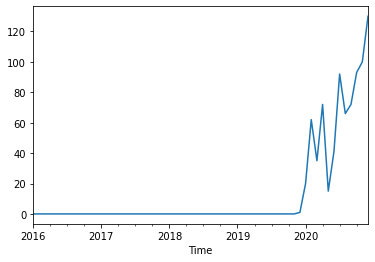


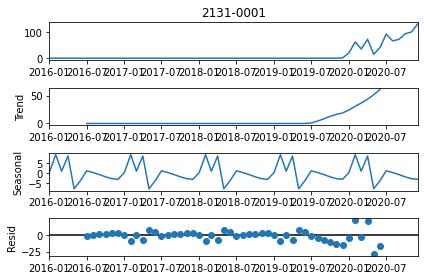


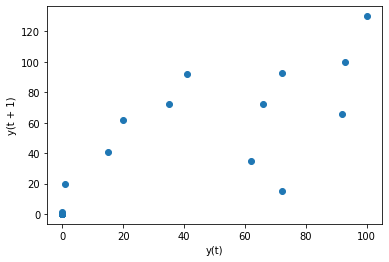
Best ARIMA(0, 1, 0) RMSE=80.141

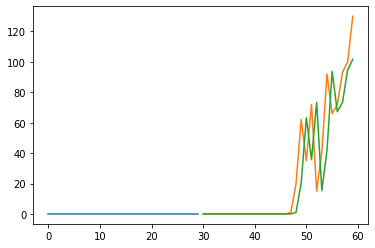
#### Part ID: 2131-0001

#### Part Desc: PC HOT SWAP CHASSIS,8 DRIVES, 4U SUPPORT ATX MOTHERBOARD ONLY





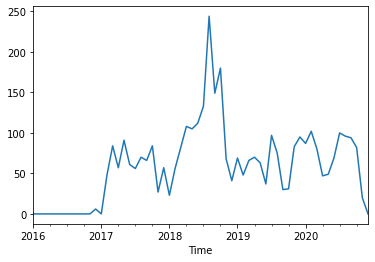


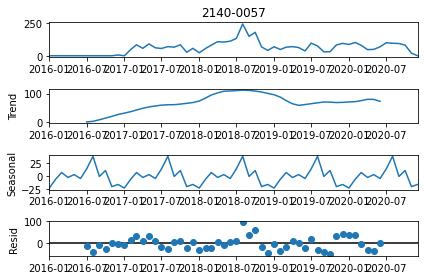


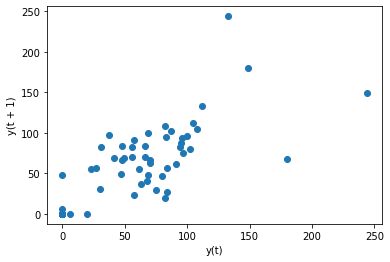
Best ARIMA(0, 1, 0) RMSE=20.664

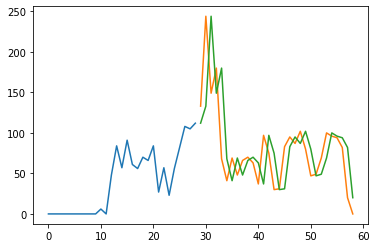
1. Part ID: 2140-0057

Part Desc: Power Supply, 750W ATX, 110V-240V, 80 Plus Gold





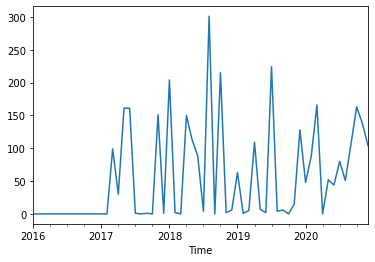


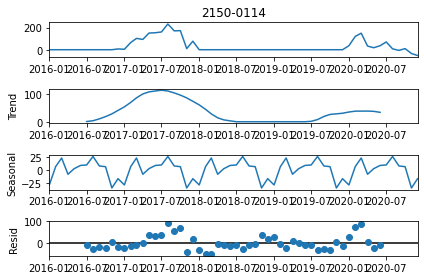


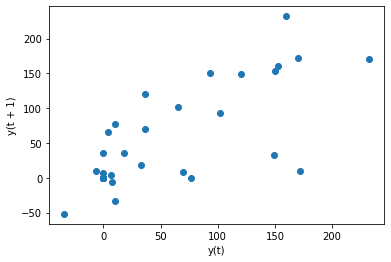
Best ARIMA(1, 0, 0) RMSE=41.389

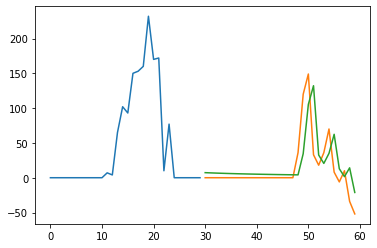
1. Part ID: 2150-0114

Part Desc: SATA Transit Cable: one connector 20cm





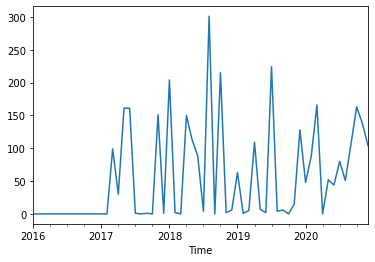


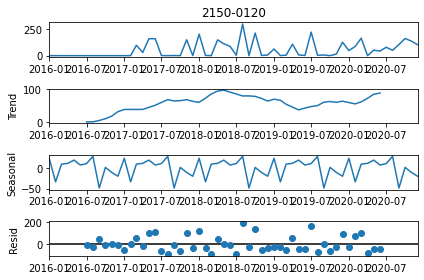


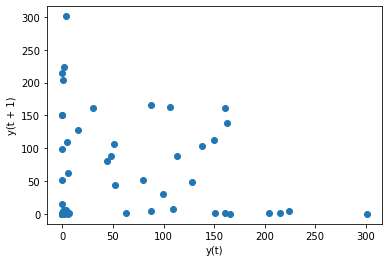
Best ARIMA(1, 0, 0) RMSE=31.072

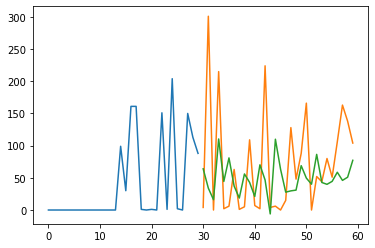
1. Part ID: 2150-0120

Part Desc: FLASH DRIVE USB 2.0, 32GB







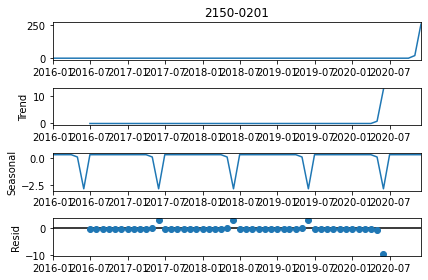


Best ARIMA(0, 0, 2) RMSE=82.748

1. Part ID: 2150-0201

Part Desc: PC INTERNAL PERIF CABLE, INTERNAL CABLE, 2 PORT USB BRACKET (4+4PINS), 30CM





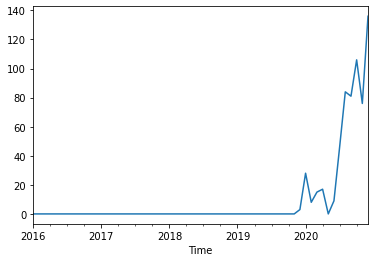


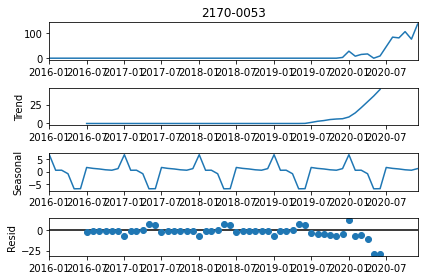


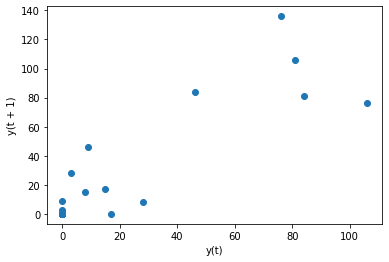
Best ARIMA(0, 2, 0) RMSE=40.268

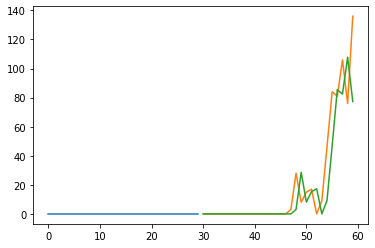
1. Part ID: 2170-0053

Part Desc: MOTHERBOARD,ATX, LGA1151, IMBA-Q370-R10-VIT





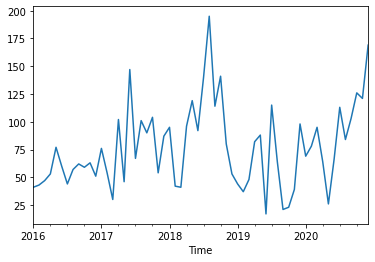


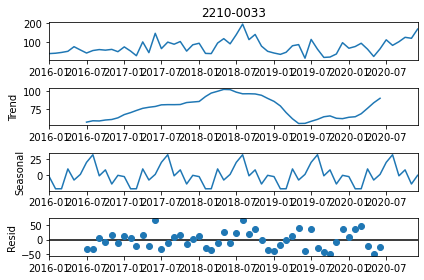


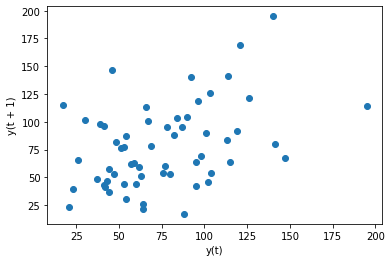
Best ARIMA(0, 1, 0) RMSE=17.566

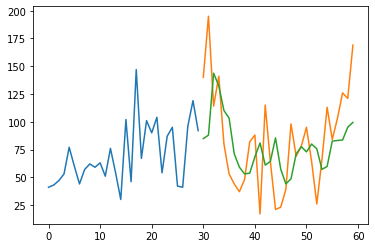
1. Part ID: 2210-0033

Part Desc: Industrial IO - PCI High Speed Digital Isolated 16-Ch Input & 16-Ch Output Card





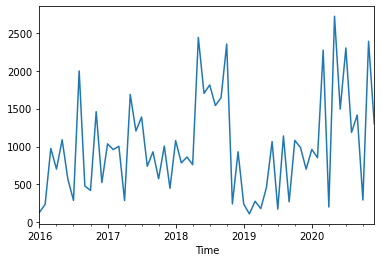


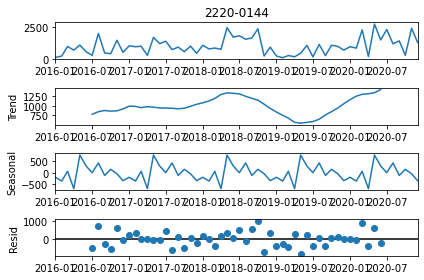


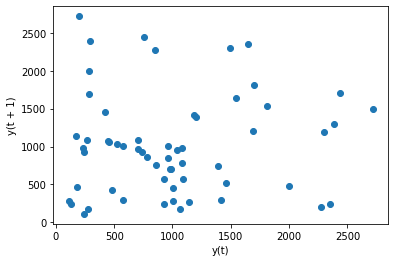
Best ARIMA(2, 0, 0) RMSE=40.814

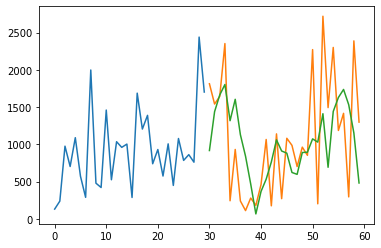
1. Part ID: 2220-0144

Part Desc: RES,SMD,150R,1%,0.25W,1206





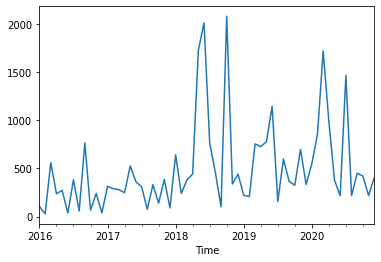


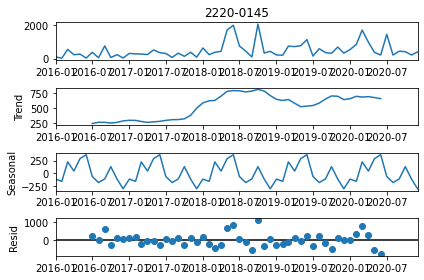


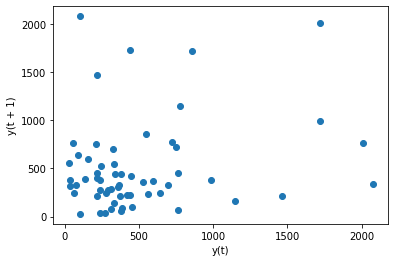
Best ARIMA(0, 0, 5) RMSE=697.553

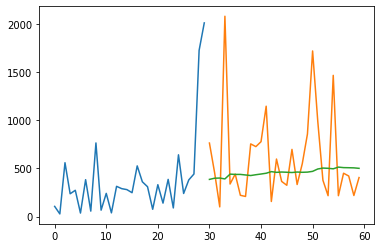
1. Part ID: 2220-0145

Part Desc: RES,SMD,47R,1%,0.25W,1206





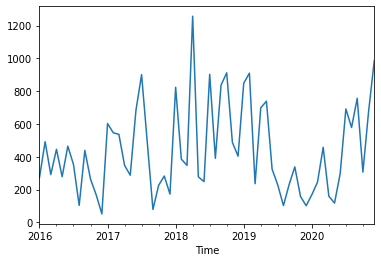


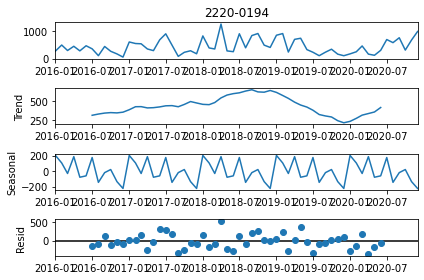


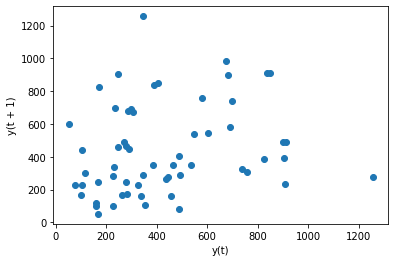
Best ARIMA(0, 0, 0) RMSE=496.279

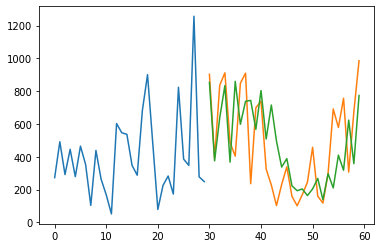
1. Part ID: 2220-0194

Part Desc: RES, SMD, 20R, 1%, 2W, 2512





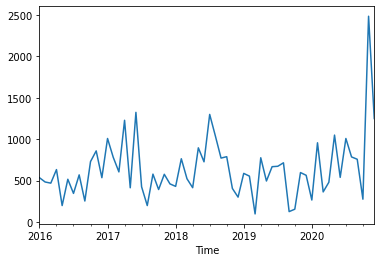


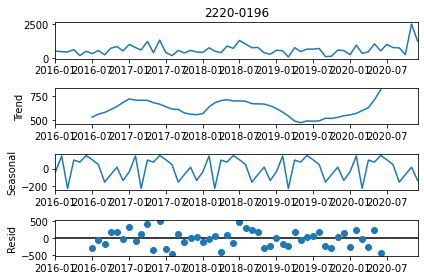


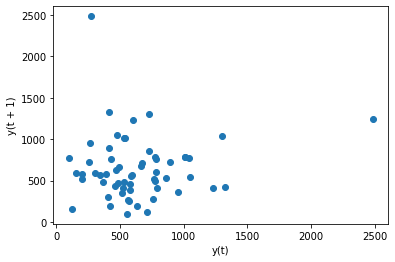
Best ARIMA(4, 1, 0) RMSE=250.138

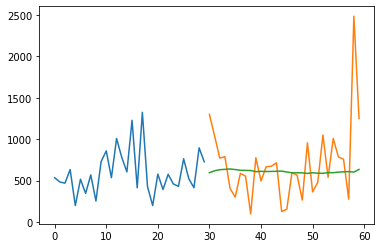
1. Part ID: 2220-0196

Part Desc: RES, SMD, 68R, 1%, 0.25W, 1206





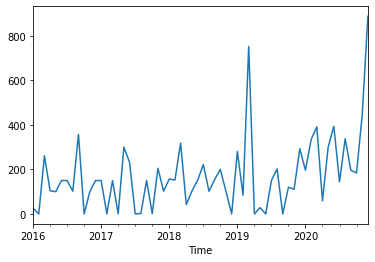


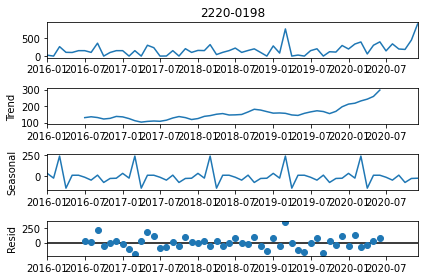


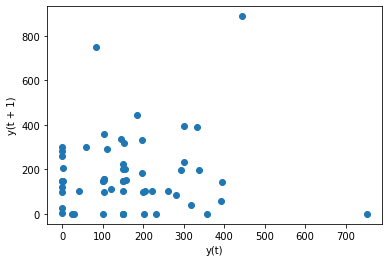
Best ARIMA(0, 0, 0) RMSE=464.158

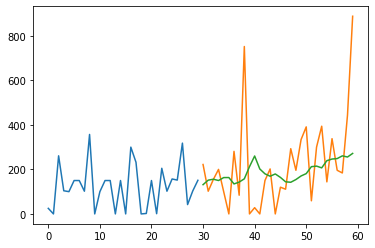
1. Part ID: 2220-0198

Part Desc: RES, MF, 47R, 1%, 0.5W





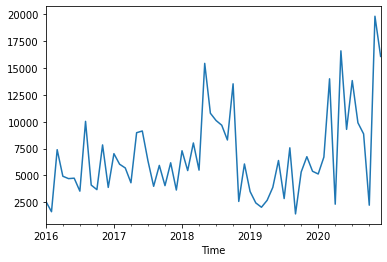


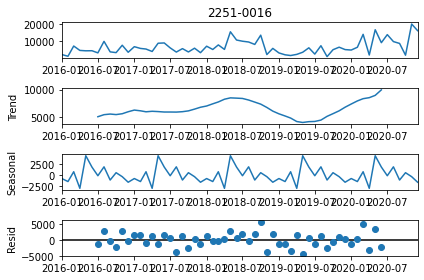


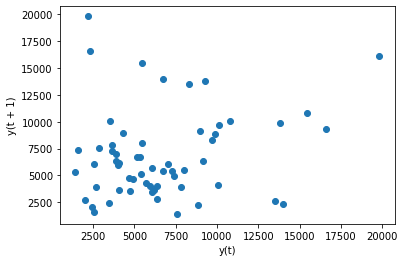
Best ARIMA(0, 2, 2) RMSE=200.112

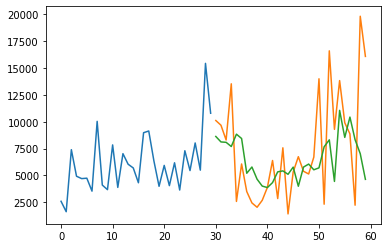
1. Part ID: 2251-0016

Part Desc: LED, THRU-HOLE, RED, TOP, 3.0MM





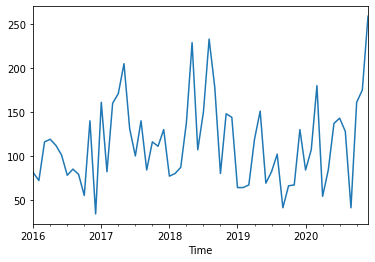


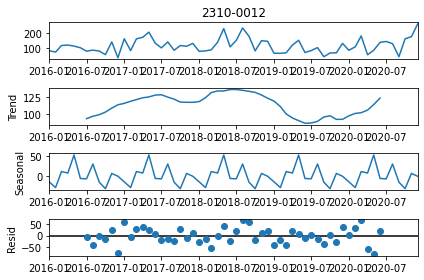


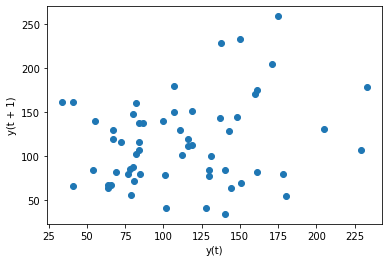
Best ARIMA(2, 0, 0) RMSE=4754.721

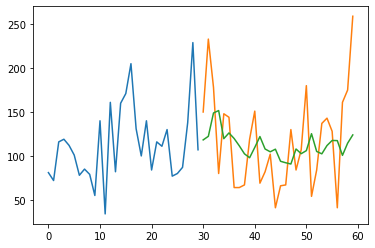
1. Part ID: 2310-0012

Part Desc: EXTENSION RING, 0.5-1-5-10-20MM





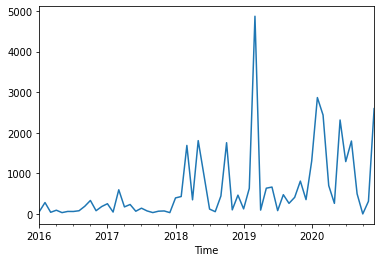


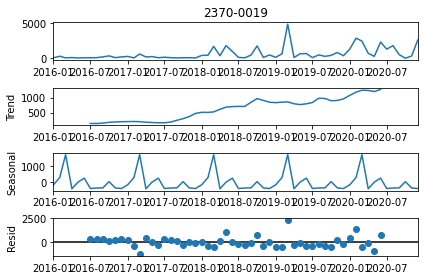


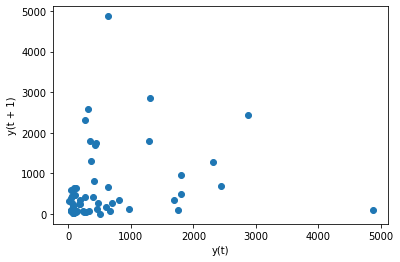
Best ARIMA(1, 0, 1) RMSE=53.320

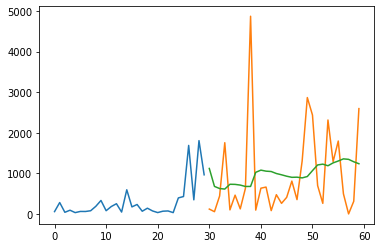
1. Part ID: 2370-0019

Part Desc: LED,THRU-HOLE,BLUE,TOP,3.8X3.5MM





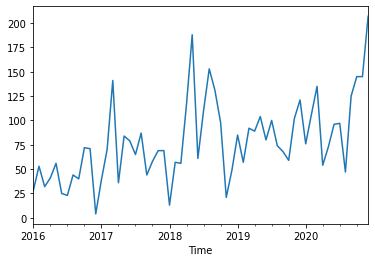


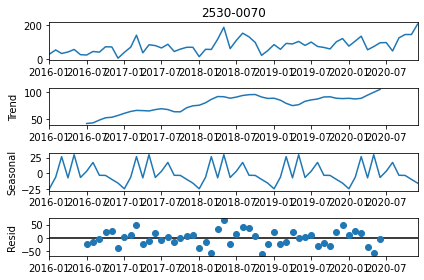


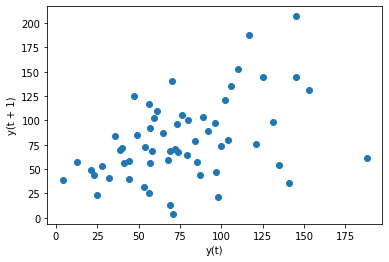
Best ARIMA(0, 1, 1) RMSE=1131.902

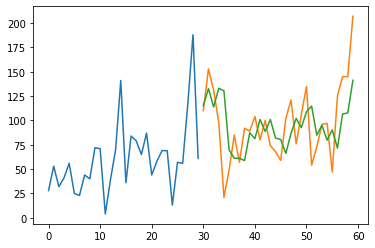
1. Part ID: 2530-0070

Part Desc: CABLES,EXTENSION,USB (Cab Ext-USB 3.0m)





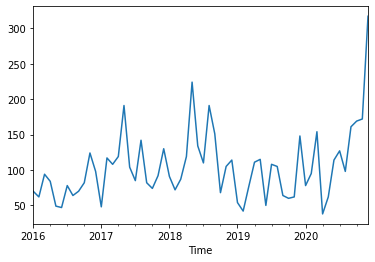


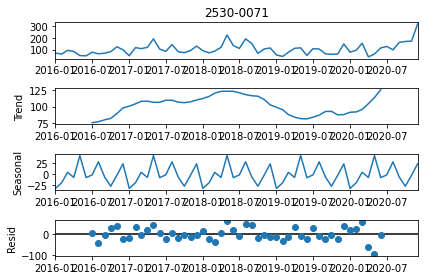


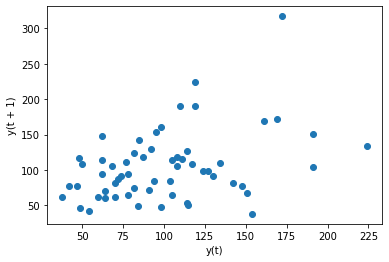
Best ARIMA(2, 1, 0) RMSE=35.850

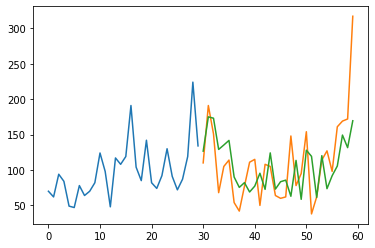
1. Part ID: 2530-0071

Part Desc: CABLES,EXTENSION,USB,MALE,3.0M





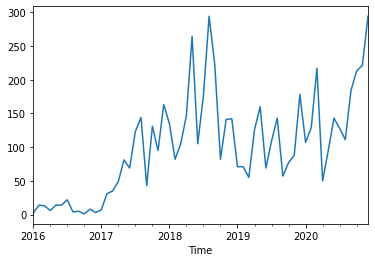


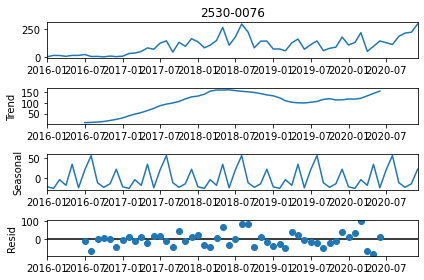


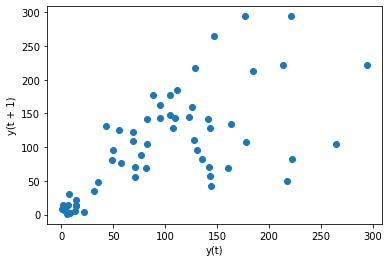
Best ARIMA(2, 1, 0) RMSE=46.124

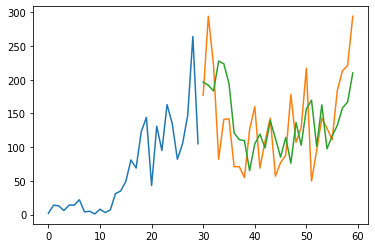
1. Part ID: 2530-0076

Part Desc: CAMERA CABLE, USB3.0, A-MICRO-B, 4M





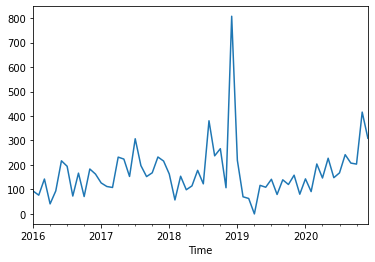


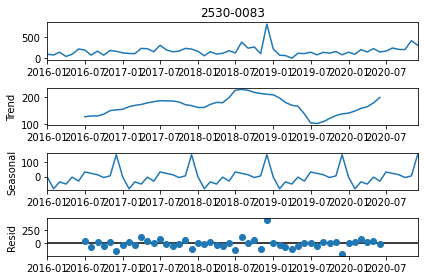


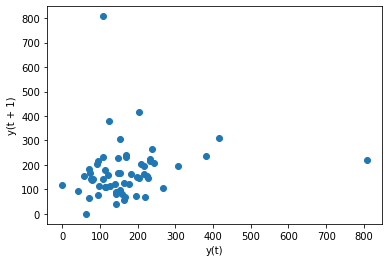
Best ARIMA(2, 1, 0) RMSE=60.661

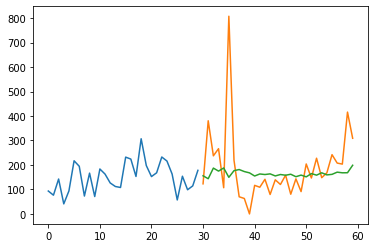
1. Part ID: 2530-0083

Part Desc: CABLE, FLEXI, 2C X 0.25, 500M





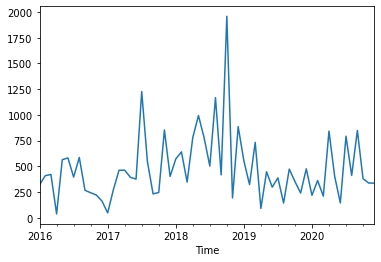


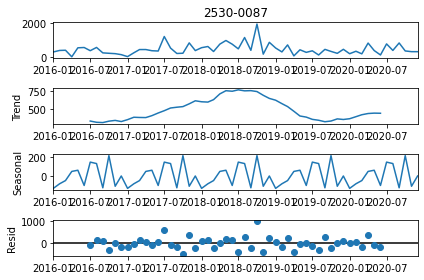


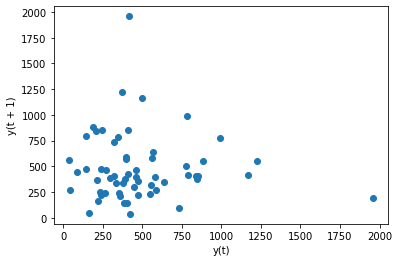
Best ARIMA(1, 0, 0) RMSE=150.322

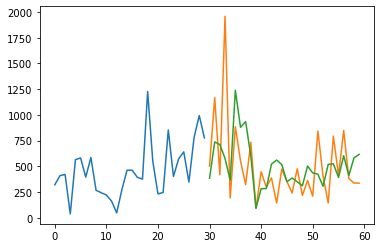
1. Part ID: 2530-0087

Part Desc: CABLE,SHIELDED,6C,26AWG,2464 VW-1,100M





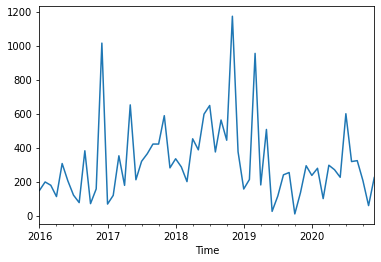


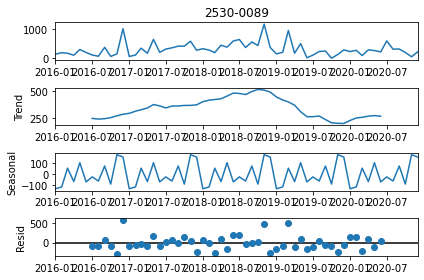


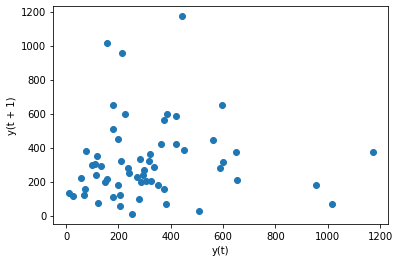
Best ARIMA(0, 0, 5) RMSE=355.498

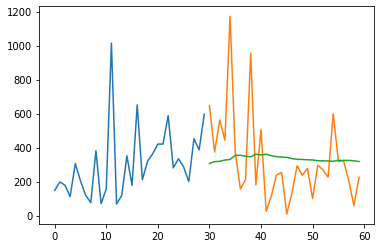
1. Part ID: 2530-0089

Part Desc: CABLE,SHIELDED,15C,26AWG,2464 VW-1,100M





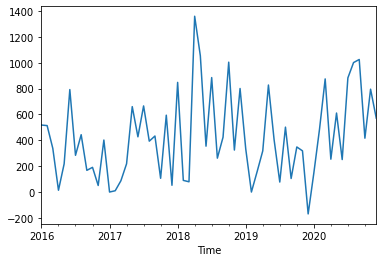


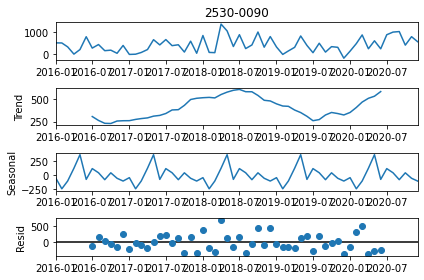


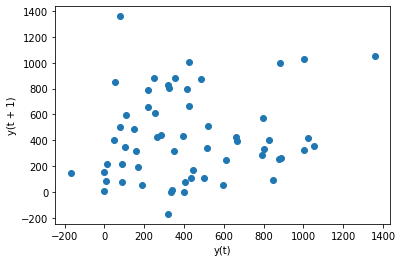
Best ARIMA(0, 0, 0) RMSE=256.552

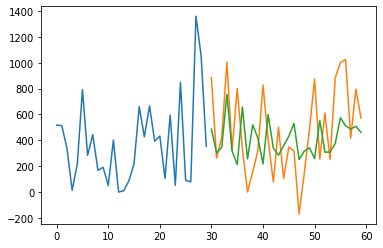
1. Part ID: 2530-0090

Part Desc: CABLE,BLACK,2C,28AWG,2547 VW-ISC,300M





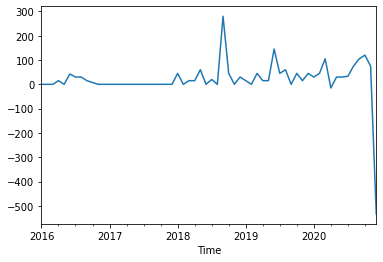


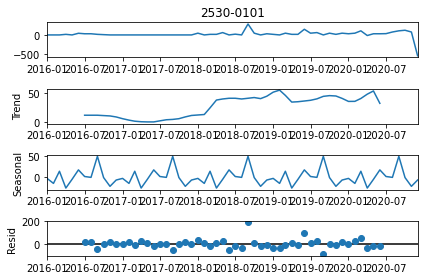


Best ARIMA(7, 0, 0) RMSE=322.547

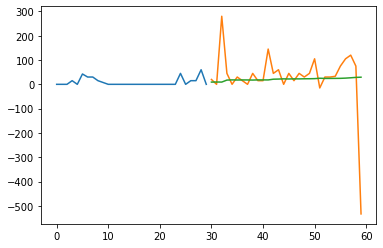
1. Part ID: 2530-0101

Part Desc: CABLE, HK09 INT LIY, GREY





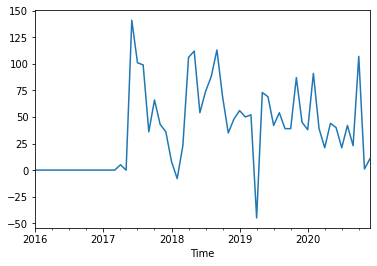


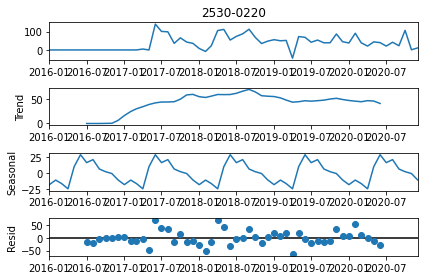


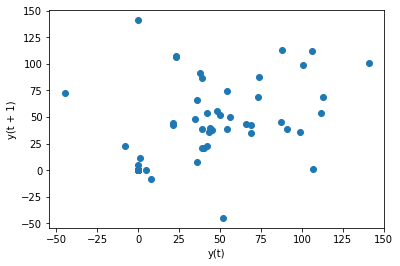
Best ARIMA(0, 0, 0) RMSE=121.167

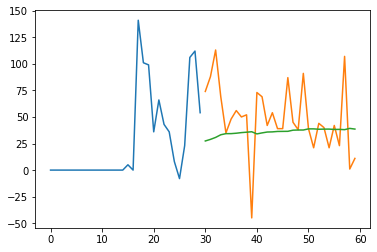
1. Part ID: 2530-0220

Part Desc: SATA Cable Straight to Straight 60cm-Red





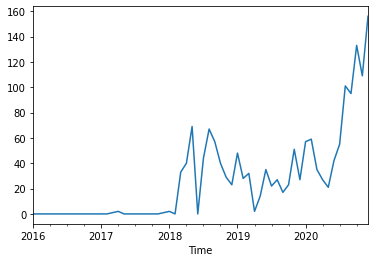


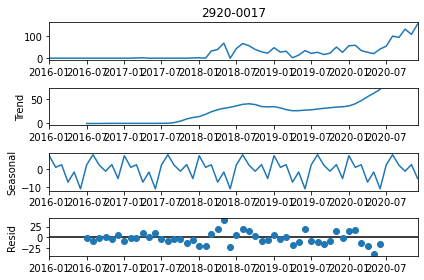


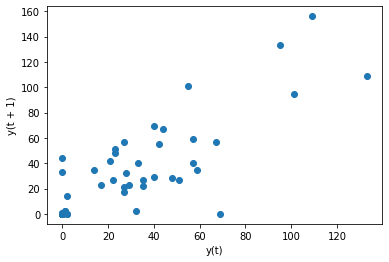
Best ARIMA(0, 0, 0) RMSE=35.492

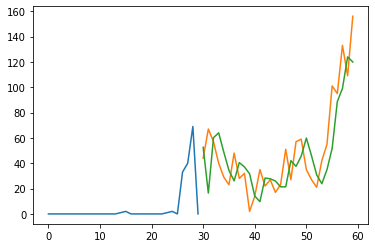
1. Part ID: 2920-0017

Part Desc: OS, WINDOWS 10 PRO 64-BIT OEM





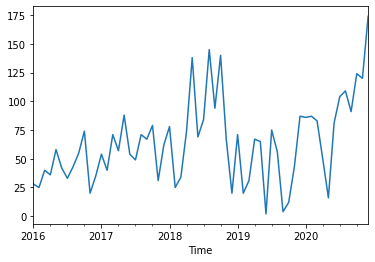


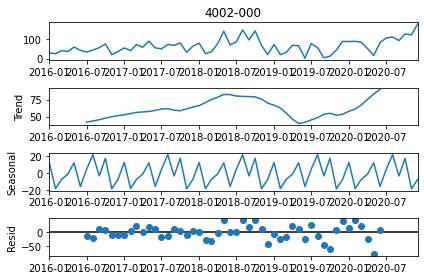


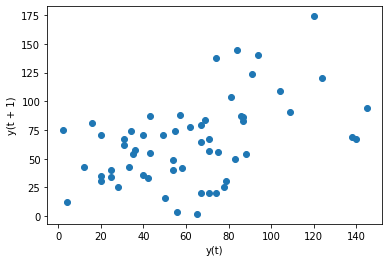
Best ARIMA(1, 1, 0) RMSE=22.098

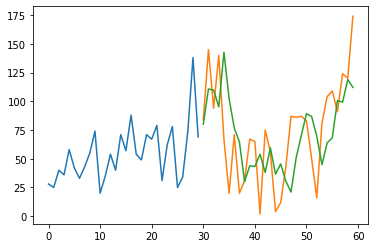
1. Part ID: 4002-000

Part Desc: PwC PCInt12V-4ways-ToDB9Fx4-0.25m





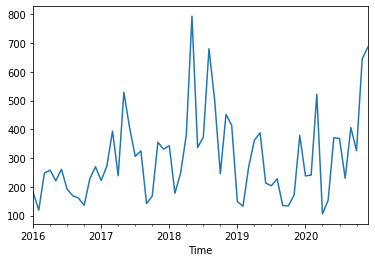


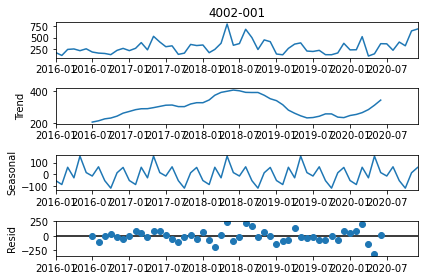


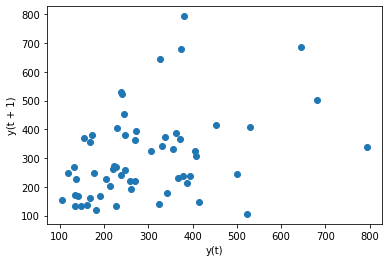
Best ARIMA(2, 1, 0) RMSE=38.580

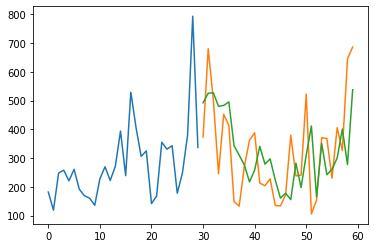
1. Part ID: 4002-001

Part Desc: PVC ULSC-LS Cables-2ways-BM-To2waysBF-3m-8022-008(Turnkey)





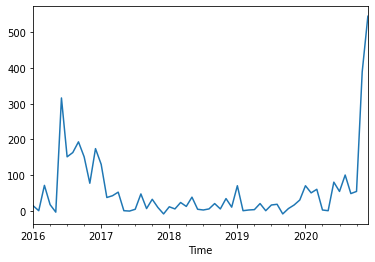


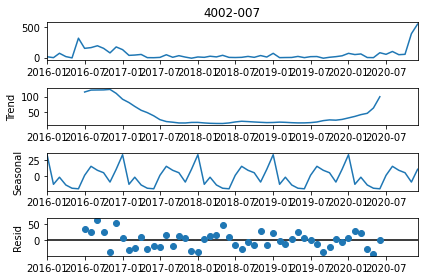


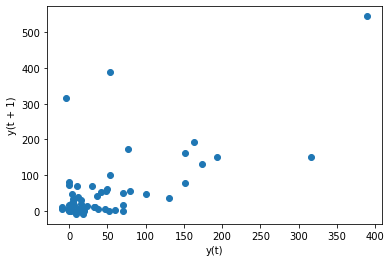
Best ARIMA(2, 1, 0) RMSE=144.500

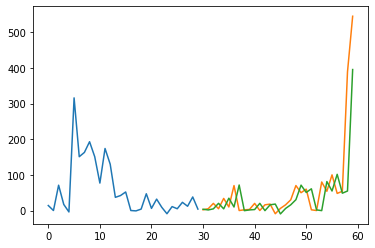
1. Part ID: 4002-007

Part Desc: PwC PCInt12V-4ways-Female-ToMalex2-0.3m





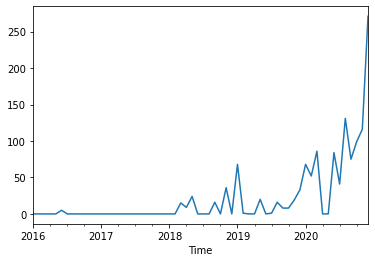


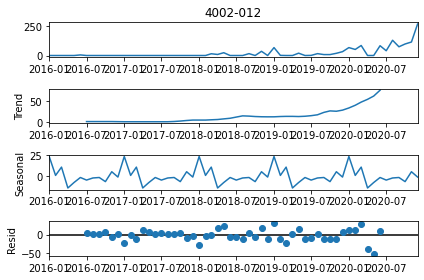


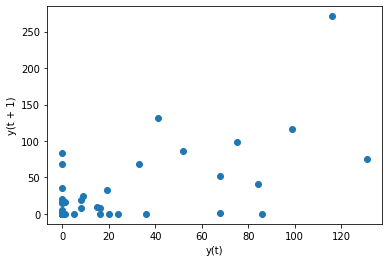
Best ARIMA(0, 1, 0) RMSE=74.108

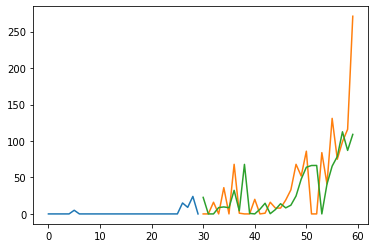
1. Part ID: 4002-012

Part Desc: Pwc ConV-MotherboardToCom(RUBY-9716VGAR)-0.7m









Best ARIMA(1, 1, 0) RMSE=44.276

# Conclusion

Time series forecasting has been used to predict parts’ consumption for the MVSS department in Vitrox. As with any machine learning, understanding the data and processing it is very important to understanding the underlying temporal structure of the time series. Many more modifications and improvements can be made to improve the prediction of the parts’ consumption through feature processing and trying other models.

# Reference

1. Brownlee, J. (2020). Introduction to time series with Python. (1.9 ed.). Machine Learning Mastery.