Task 1: Forecasting Model Proposal

Problem Statement: Develop a proposal for a long-range forecast model that will primarily be used to generate forecasts within a 10-year horizon

Solution:

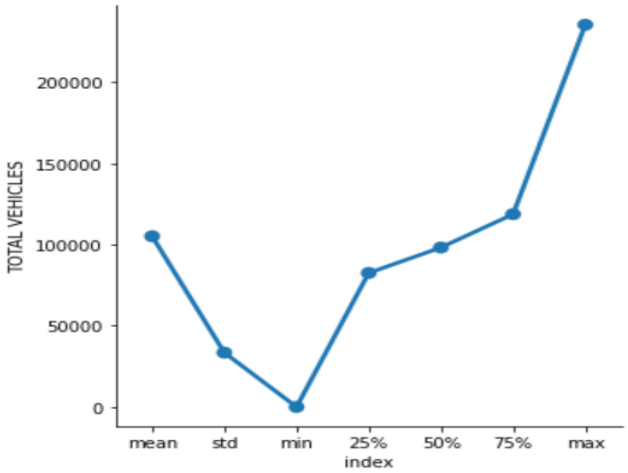
The data provided for the task is a 19-years worth historical monthly data of traffic flow between Vancouver and Nanaimo. In order to create a long-range forecast for this data, we start with exploring the data and it's features

1. **Exploratory Data Analysis** - BC Ferries routes travelling between Vancouver and Nanaimo

```
Index(['Date', 'TOTAL VEHICLES'], dtype='object')
```

The Data comprises of two columns: Date and TOTAL VEHICLES. Currently both have the datatype of 'object' but before proceeding to build a forecasting model, we would have to convert 'Date' column to 'datetime' format and reindex the dataset as well.

We are using Python's packages such as Pandas, Seaborn, and matplotlib to explore this data. Please feel free to view the code at https://colab.research.google.com/drive/1D9szm6jgCUCvMbQURtPf4_jwfT34Mn1u?usp=sharing



Let's look at the data description which would include the standard deviation, mean and other statistical measures:

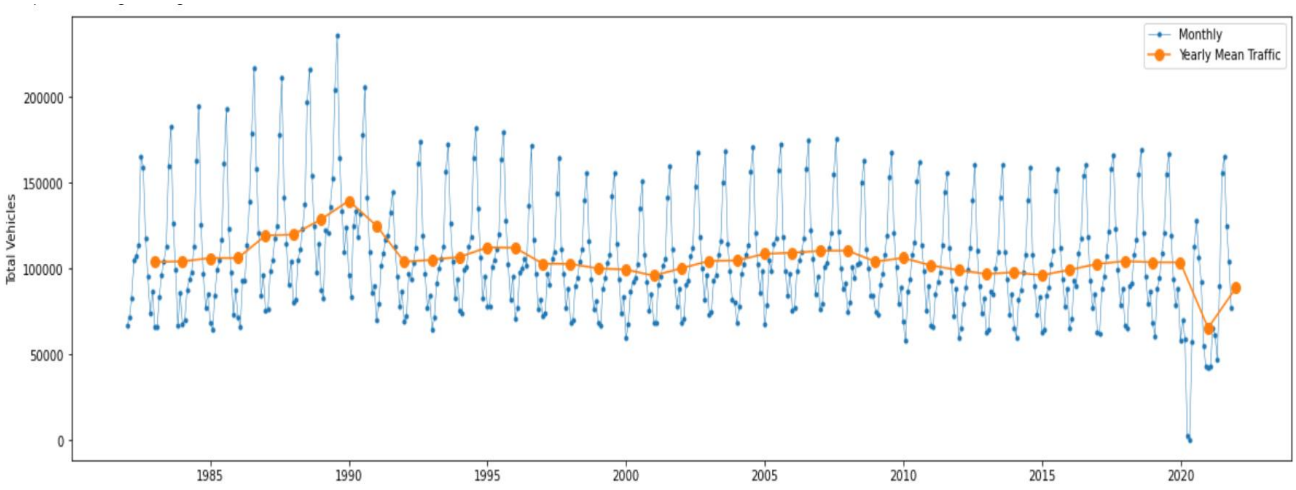
As the graph above shows, mean vehicles travelling every year is a little above 100000 and the standard deviation is around 25000.

Checking the data for Null Values

We also checked for null values in the dataset, but there were none. So we don't have to refine/clean the data at this point.

Visualizing Time Series for Total Vehicles

The time series visualization for the total number of vehicles in a particular period will further give us a high-level overview of the trend that our data follows. And if there are any other anomalies that may need solving



By looking at the graph above, we can tell that it is a Seasonal Trend, which repeats every year except for a few anomalies such as in 1990, and 2021. The trend though stays the same over time, at least by looking at the visualization.

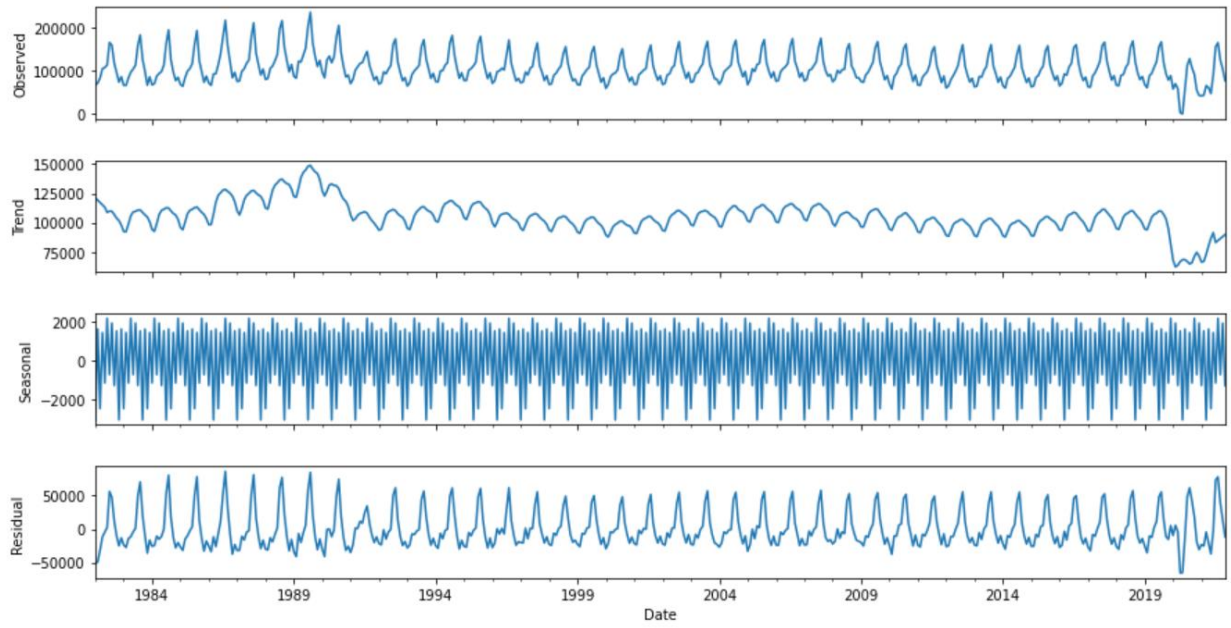
2. Identification of key attributes in the data – boon/bane

Until now we have seen what kind of data do we have and what is the trend that it follows, in this section we are going to look at some of the key attributes in the data that might support our model or create problems in it.

Decompose the Data

By looking at the graph of traffic data above, we can see a general trend with no clear pattern of seasonal or cyclic changes. The next step is to decompose the data to view more of the complexity behind the linear visualization. A Python library called 'statsmodels' helped us decompose the data into four different components:

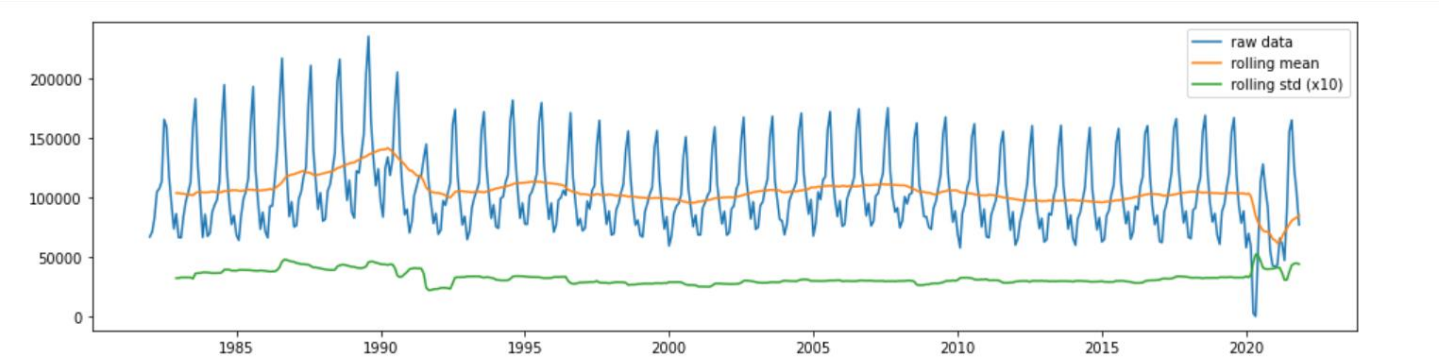
a. Observed
b. Trended
c. Seasonal
d. Residual



After looking at the four pieces of decomposed graphs, we can tell that our traffic dataset has an overall same trend as well as a yearly seasonality. Depending on the components of your dataset like trend, seasonality, or cycles, the choice of model will be different.

Check for Stationarity

A dataset is stationary if its statistical properties like mean, variance, and autocorrelation do not change over time. Most time series datasets related to business activity are not stationary since there are usually all sorts of nonstationary elements like trends and economic cycles. But since most time series forecasting models use stationarity and mathematical transformations related to it to make predictions, we need to stationarize the time series as part of the process of fitting a model



Both the mean and the standard deviation for stationary data does not change over time (except for the outliers). But in this case, since the y-axis has such a large scale, we can not confidently conclude that our data is stationary by simply viewing the above graph. Therefore, we should do another test of stationarity.

So we move on to perform some tests on our data in order to see whether it is stationary of not.

Augmented Dickey-Fuller Test

The ADF approach is essentially a statistical significance test that compares the p-value with the critical values and does hypothesis testing. Using this test, we can determine whether the processed data is stationary or not with different levels of confidence.

```
ADF_test(y,'raw data')
```

```
 > Is the raw data stationary ?
Test statistic = -3.220
P-value = 0.019
Critical values :
        1%: -3.4446148284445153 - The data is not stationary with 99% confidence
        5%: -2.8678299626609314 - The data is  stationary with 95% confidence
        10%: -2.5701203107928157 - The data is  stationary with 90% confidence
```

If the p-value in the ADF test is less that 0.05, then we can reject the alternate hypothesis and accept that the data is stationary. Only data with 1% confidence is not stationary and that is because of the outliers caused by external factors. We will deal with the outliers at a later stage during model building.

3. The additional information we have gathered until now for this model is that it is a seasonal data with the trend remaining the same except for outliers caused by external factors. We have also established that the data is already stationary and does not need de-trending or differencing which are techniques used in the case of non-stationary data.
4. **Choosing appropriate Forecasting Model:**

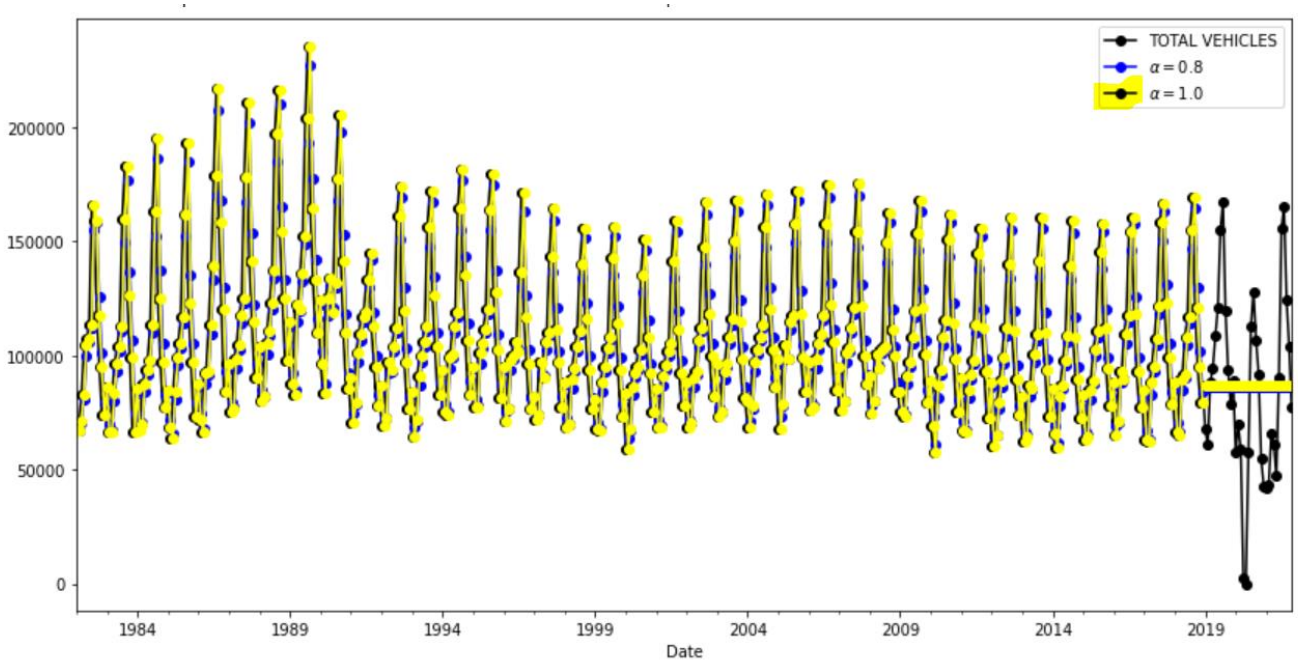
   We will look at four prediction models:

   a. Simple Exponential Smoothing (SES)
   b. Holt
   c. Seasonal Holt-Winters
   d. Seasonal ARIMA (SARIMA)

Then we will evaluate these forecasting models to determine which is best for our sample dataset. Not all of these models are suitable for our dataset, but we are going to walk through them to describe the options available and show why not all models are appropriate of all datasets. The appropriate model depends upon the time-series data and the data's particular characteristics.

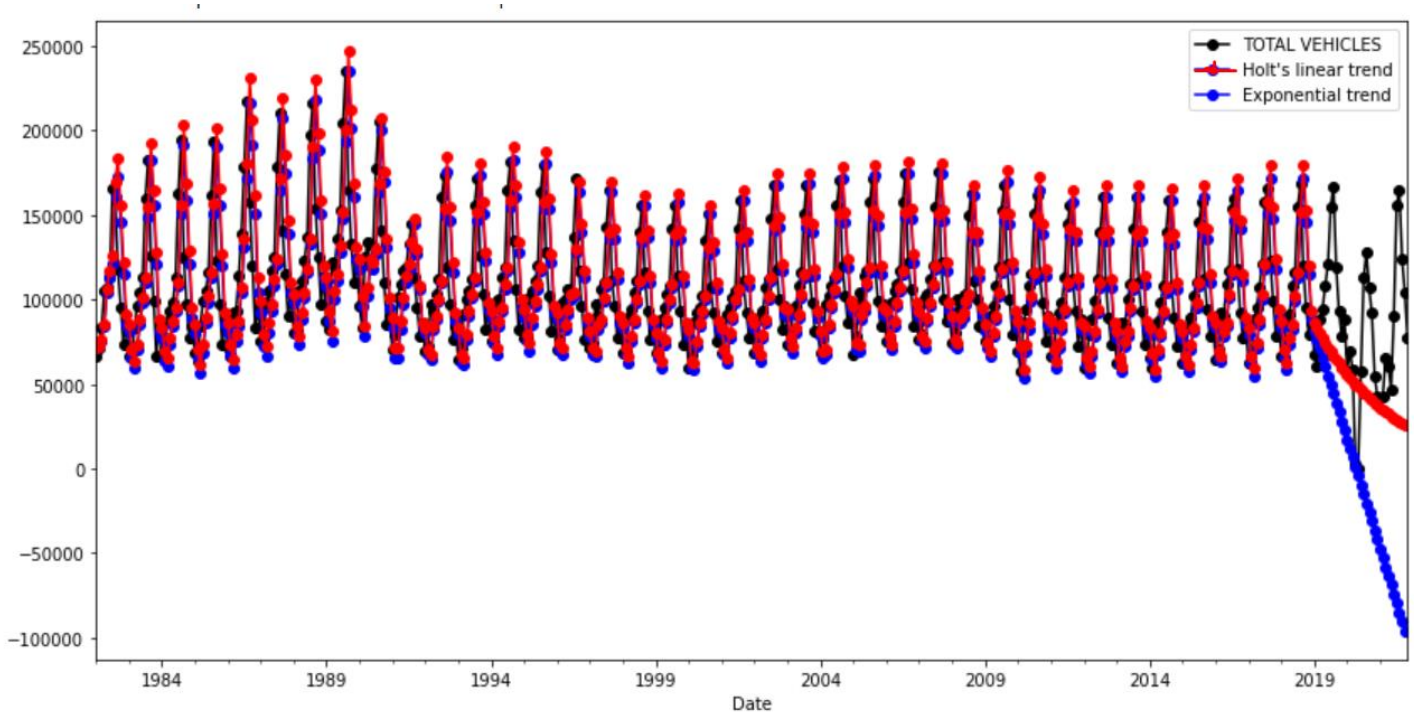
To evaluate the performance of these models, we have used a measurement-metric called Root Mean Squared Error (RMSE) to measure the difference between predicted values and the actual or observed values.

- **Simple Exponential Smoothing (SES):** Suitable for time series data without trend or seasonal components.
  This model calculates the forecasting data using weighted averages. One important parameter this model uses is the smoothing parameter: alpha and we have picked a value between 0 and 1 to determine this value. Our code also allows us to auto-optimize this value and we have plotted the graph for both.



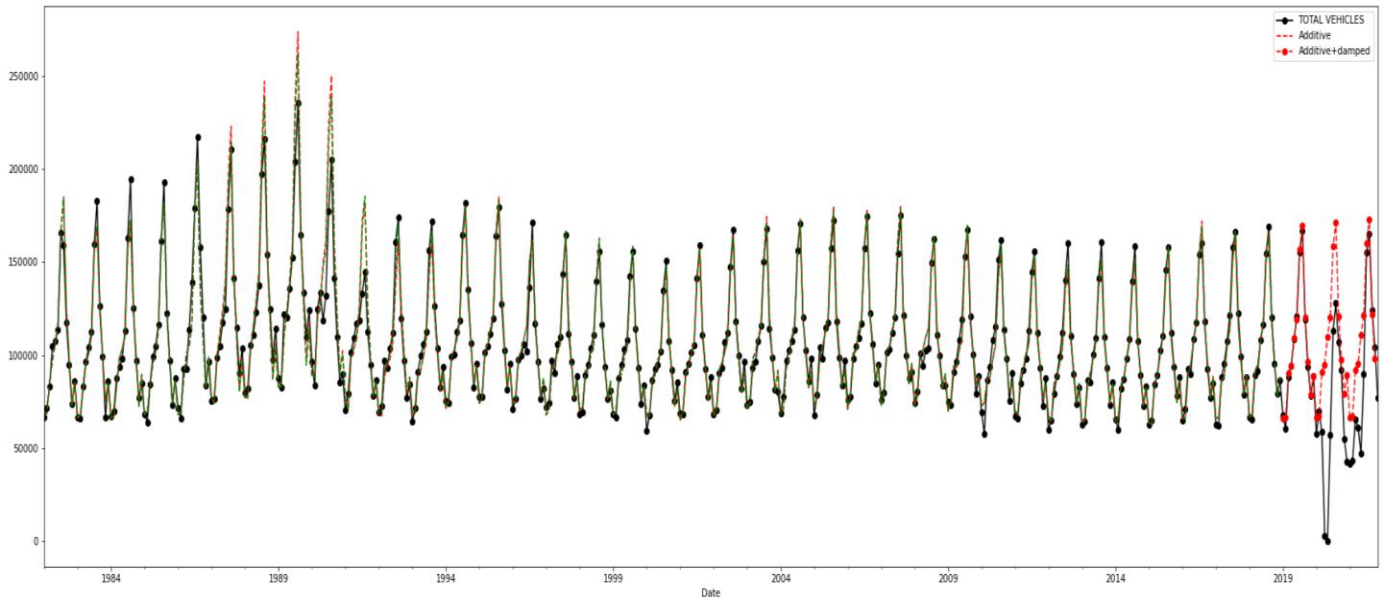
The graph above shows that even after a low Root-Mean-Squared value, both the alpha = 0.8(blue line) and auto-optimized values(yellow line) give us a straight line forecast because SES will predict a flat, forecasted line since the logic behind it is using a weighted-average. So it can't capture any seasonality or trend.

- **Holt's Linear Trend Method:** Suitable for time series with a trend component but without a seasonal component
  Expanding the SES method, the Holt method helps forecast time series data that has a trend. In addition to the level smoothing parameter alpha along with the SES method, the Holt adds the trend smoothing parameter Beta. Like alpha, the range of Beta is also between 0 and 1. We have again visualized two variants of Holt (additive and multiplicative)



The visualization above shows additive trend(red line) and exponential trend(blue line). Compared to SES, Holt captures more trend of the data but since our data has an overall same trend, a dramatic decreasing trend is not really valid in this case, and unlikely in real life.

- **Holt-Winters' Seasonal Method:** Suitable for time series with trend and/or seasonal components. This model extends Holt but also takes into consideration seasonality as well as trend. This method also considers smoothing parameter, gamma.
  Two types of seasonality: Additive (seasonal changes in data stays roughly the same over time and don't fluctuate in relation to overall data). Multiplicative (seasonal variation changes in relation to overall changed in the data)



The visualization of the results for the Holt-Winters method shows the additive (red line) compared to the additive + damped (green line) trends. Based on the visualization, we see that the Holt-Winters model fits the actual data best, so far. However, the RMSE is not better than the results from the simple SES model. And we can also tell that the forecast starts to drop off towards the end.

Our data has a yearly seasonal pattern with 19 years of data, and we aggregated it by month so each data point is one month, m = 12

- **SARIMA** - Suitable for time series data with trend and/or seasonal components

  While exponential smoothing models use weighted averages of past observations to forecast new values, Auto-Regressive Integrated Moving Average or ARIMA models look at autocorrelations or serial correlations in the data. In other words, ARIMA models look at differences between values in the time series. SARIMA builds upon the concept of ARIMA but extends it to model the seasonal elements in your data.

  Trend Elements:

  p: Trend autoregression order.
  d: Trend difference order.
  q: Trend moving average order.

  Seasonal Elements:
  P: Seasonal autoregressive order.
  D: Seasonal difference order.
  Q: Seasonal moving average order.
  m: The number of time steps for a single seasonal period.

In order to get the best prediction it is important to find the values of SARIMA(p,d,q)(P,D,Q)m that optimize a metric of interest. For the purposes of this task we will just use grid search functionality offered by Python's library to iteratively explore different combinations of parameters. The evaluation metric used is AIC(Akaike

Information Criterion) value. The AIC measures how well a model fits the data while considering the complexity of the model as well. We would want to pick the lowest AIC

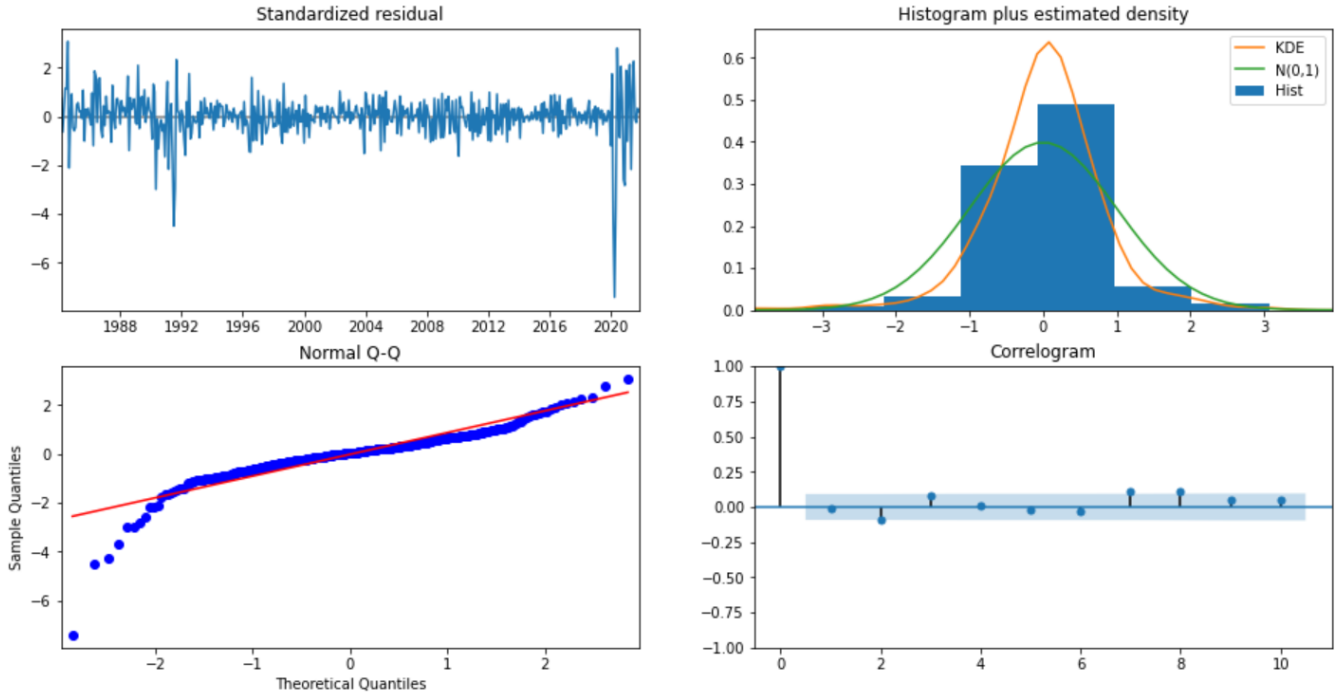
```
The set of parameters with the minimum AIC is: SARIMA(1, 0, 1)x(1, 1, 1, 12) - AIC:9408.75245426461
```

We have obtained (1,0,1) x (1,1,1,12) as the best combination and lowest AIC

We also ran some additional diagnostics on our data apart from RMSE. This functionality is only available in ARIMA and SARIMA models and cannot be plotted in the other models we saw earlier.

```
model = sarima_eva(y,(1, 0, 1),(1, 1, 1, 12),12,'2021-06-01',y_to_val)
```

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.7605      0.029     25.908      0.000       0.703       0.818
ma.L1          0.1147      0.040      2.870      0.004       0.036       0.193
ar.S.L12      -0.2212      0.118     -1.876      0.061      -0.452       0.010
ma.S.L12      -0.2715      0.102     -2.658      0.008      -0.472      -0.071
sigma2      7.369e+07   5.77e-10   1.28e+17      0.000    7.37e+07    7.37e+07
==============================================================================
```



The top left graph shows the residuals over time. We do not want to see any obvious seasonality here and the messier it is, the better the trend and seasonality in our data and removed the noise.

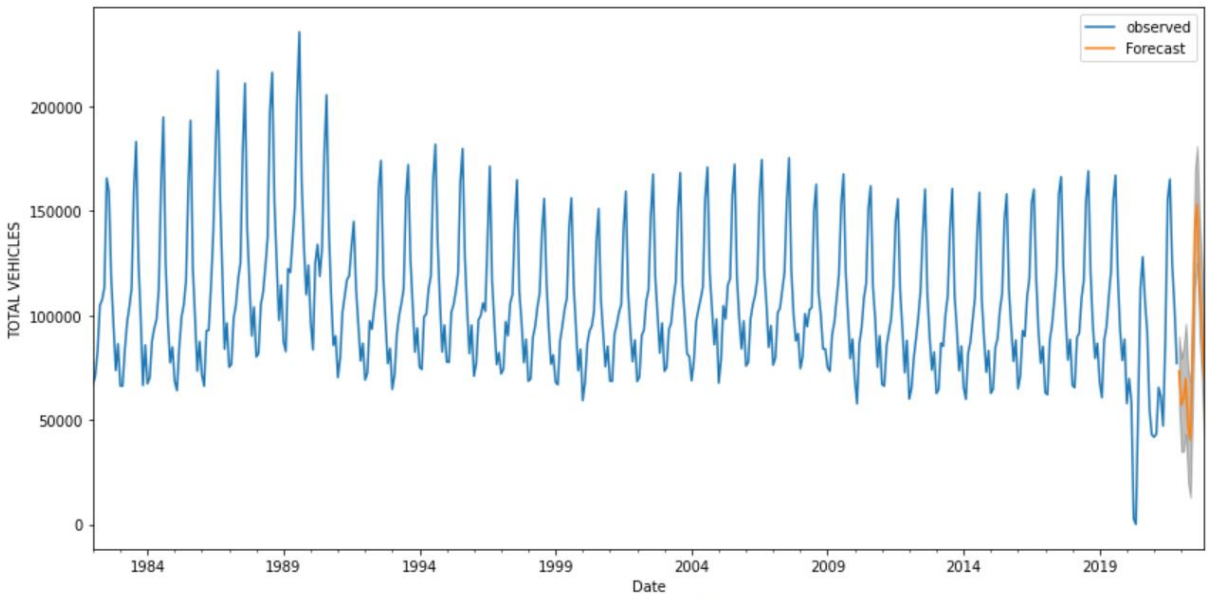
The top right plot, we want to see that the KDE follows closely with the N(0,1) line to indicate that the residuals are normally distributed. This line is standard notation for a normal distribution with a mean of 0 and a std of 1.

Bottom left qq-plot, ordered distribution of residuals should follow the linear trend(red line)

Bottom right also called correlogram, autocorrelation visual shows time series has low correlation with lagged version of itself.

With the above four points, we can conclude that this model's residuals are near normally distributed. This indicates a well-fit model suitable for our dataset.

Finally, we forecast the values for the next year using the SARIMA model we just created.

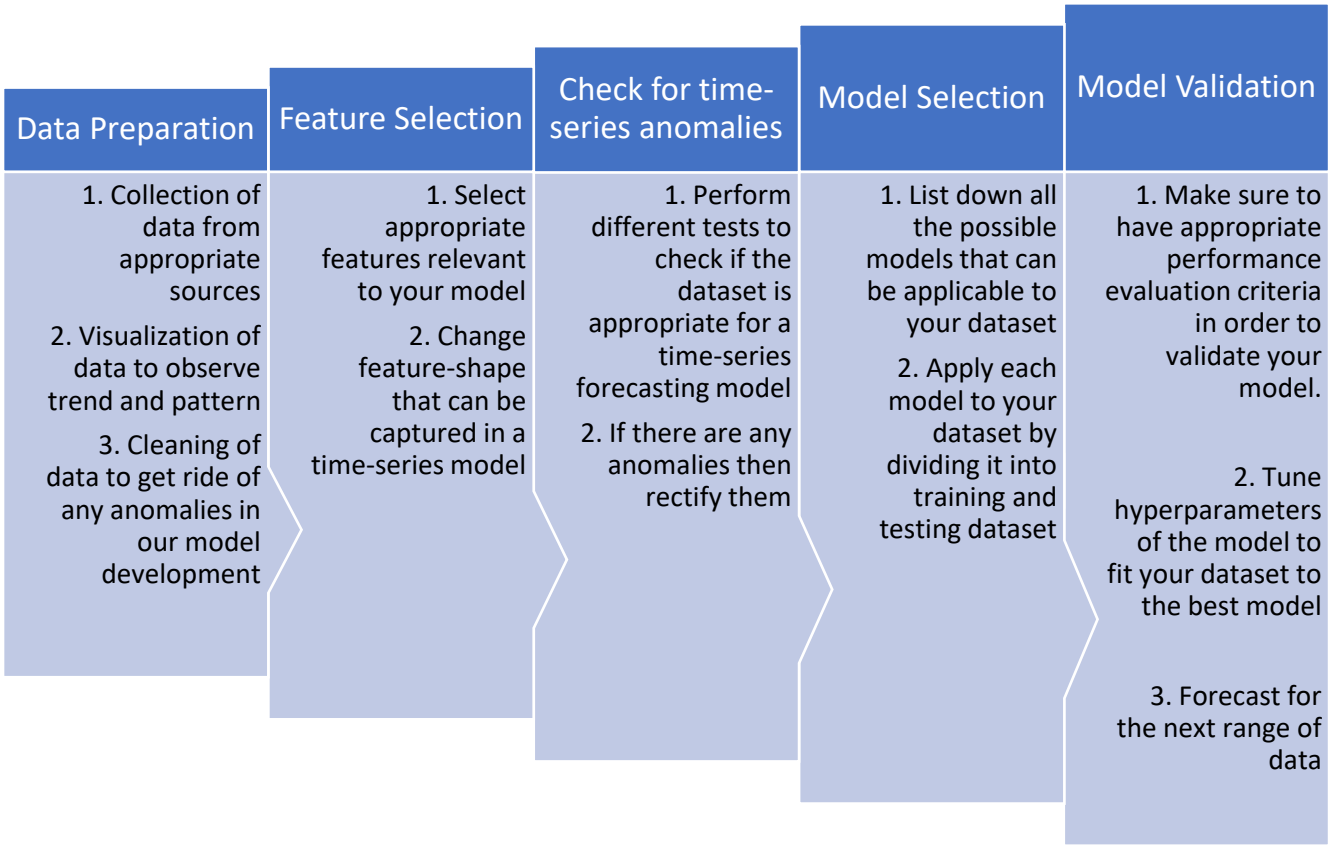


	Date	Predicted_Mean	Lower Bound	Upper Bound
0	2021-12-01	73146.96940671	56322.08019647	89971.85861695
1	2022-01-01	57053.96351734	34695.36130602	79412.56572866
2	2022-02-01	59820.26226368	34813.86505440	84826.65947296
3	2022-03-01	69634.88110397	43217.88577555	96051.87643239
4	2022-04-01	47042.12742696	19842.62601495	74241.62883896

This plot captures both the seasonality as well as no change in trend. Hence it is the best possible model for our dataset.

5. As listed above, to select the proposed model, we also need to evaluate the model on a test data which can be part of our historical data. In all the experiments above, we divided our historical data into training and test dataset. By training the model on the training dataset, we then use the same model to predict the test dataset. Since we already have the target values for the test dataset, we can clearly see what the forecast would look like even before using the model to forecast on future values. This step is very important to choose the preferred model.

6. Key Steps in Model Development and Assessment



7.