Time Series Final Project:

Analysis and Forecasting of Light Weight Vehicle Sales Time Series.

IBM Machine Learning Professional Certificate

Course 06: Specialized Models: Time Series and Survival Analysis

By ARPAN SANKESH

IBM

Contents

- Dataset Description
- Main objectives of the analysis.
- EDA, Data Cleaning, Feature Engineering
- Time series Forecasting & ML/DL Analysis and Findings
- Models flaws and advanced steps.

Abstract

In this report we are trying to explore a dataset of light weight vehicle sales in USA from 1976 to 2022 using time series techniques , in order to forecast the sales for future periods to help the owners of lightweight vehicles showrooms in USA to draw conclusions and insights and make right decision of their business.

Data Description Section

Dataset Description

Dataset Info:

Label : Light Weight Vehicle Sales (LTOTALNSA)

Release: Supplemental Estimates, Motor Vehicles

Source: U.S. Bureau of Economic Analysis

Units: Thousands of Units, Not Seasonally Adjusted

Frequency: Monthly

Citation: U.S. Bureau of Economic Analysis, Light Weight Vehicle Sales [LTOTALNSA], retrieved from FRED,

Federal Reserve Bank of St. Louis;
https://fred.stlouisfed.org/series/LTOTALNSA, May 23, 2022.

Dataset Description

Importing the dataset into a data frame & describing the attributes The dataset consists of two columns:

- DATE
- LTOTALNSA (relabeled to “SALES”)

DATE : contains monthly dates from 1976 – 2022
For instance: 1976-04-01 (YYYY-MM-DD)

SALES: contains monthly light weight vehicle sales in thousands of units

Number of records :

- 555 rows

DATE	SALES
1976-01-01	864.600
1976-02-01	973.300
1976-03-01	1216.100
1976-04-01	1163.200
1976-05-01	1176.100
...	...
2021-11-01	1014.411
2021-12-01	1203.993
2022-01-01	989.560
2022-02-01	1045.307
2022-03-01	1246.336

555 rows × 1 columns

For instance : 1163.2 thousand of vehicle (1163200 units) are sold in this date.

Main Objective of the analysis:

In this analysis we will explore the dataset of monthly Lightweight Vehicle Sales in USA from 1976 to 2022 in more details for the sake of approaching time series techniques and modelsin order to help the ownersof lightweight vehicles showrooms in USA to draw the conclusions and insights and make the right decision of their business.

Exploratory Data Analysis (EDA) + Feature Engineering Section

Exploratory Data Analysis

We are going to apply different tests & analysis to check each of :

White Noise:

Time series can be considered as white noise (can't be modeled for forecasting) if it satisfies three conditions :

1. **Approximately zero mean over the time series.**
2. **Constant standard deviation over the time series.**
3. **Specific patterns in the correlations between the time series and its lags.**

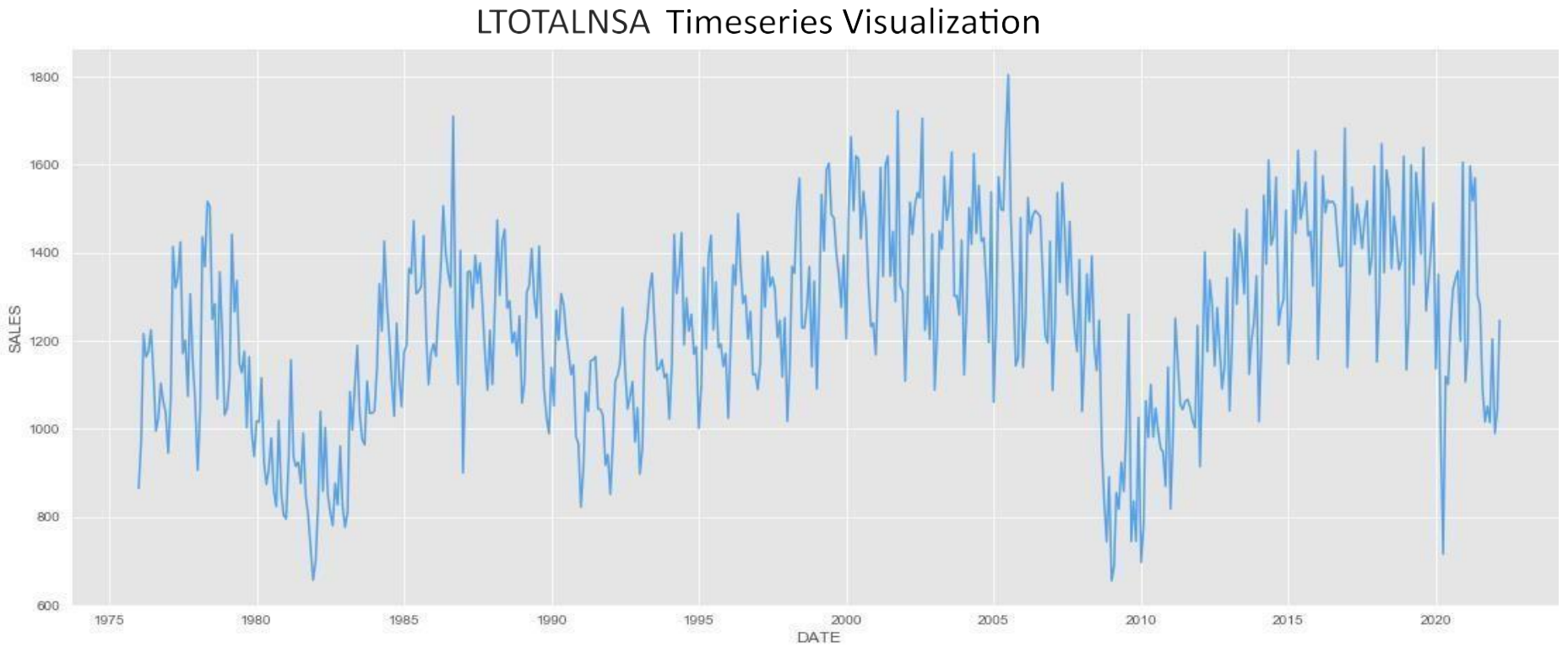
Stationarity:

In order a time series data to be stationary, the data must exhibit four properties over time:

1. **constant mean**
2. **constant variance**
3. **constant autocorrelation structure**
4. **no periodic component**

Exploratory Data Analysis

Time series visualization

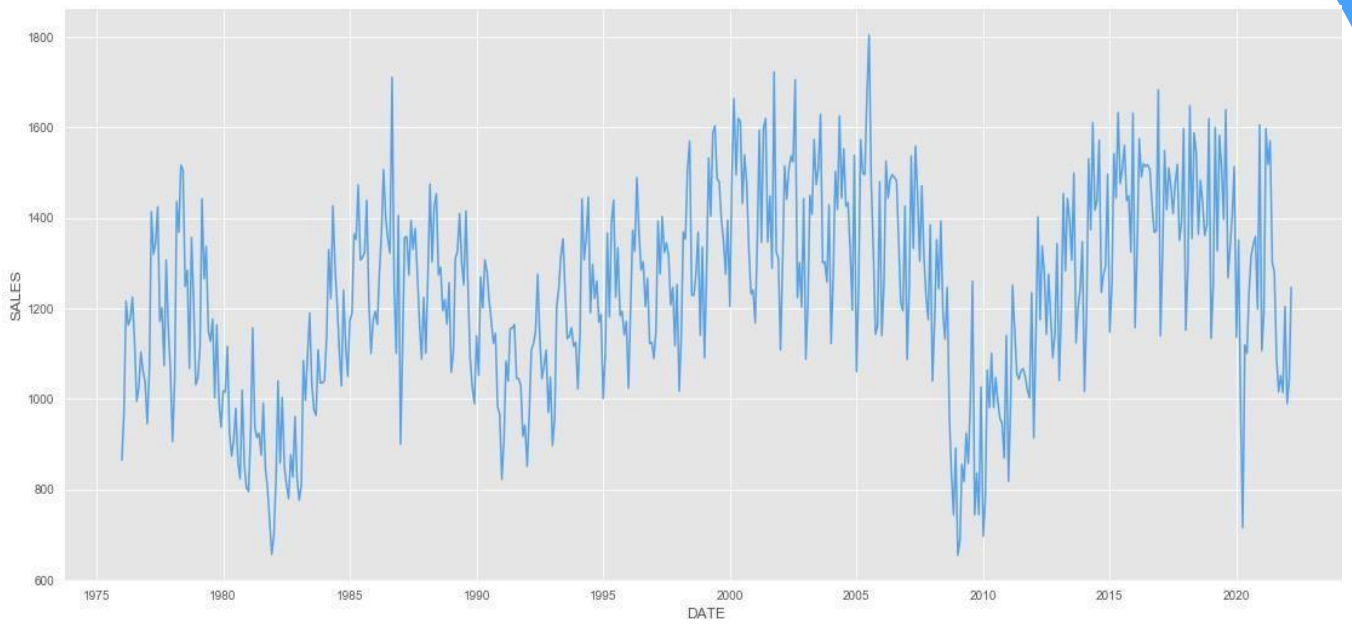


Exploratory Data Analysis

Time Series General Features :

we can extract from the graph the following features:

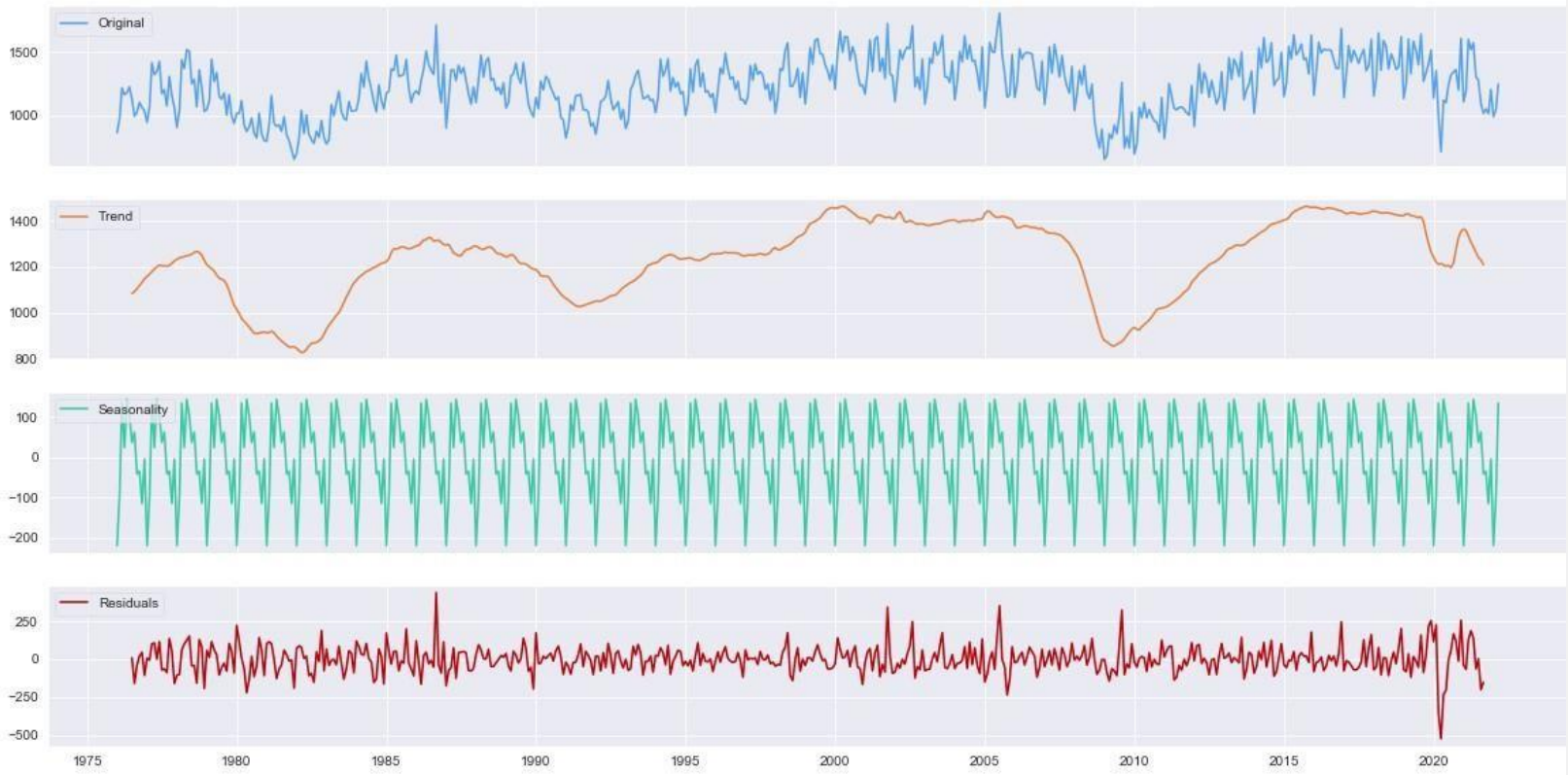
1. There is no trend in other words Stationary trend which indicates a constant mean.
2. The variance is constant.
3. The graph doesnot show component (no seasonality) a periodic



Exploratory Data Analysis

Timeseries Decomposition :

the decomposition process has the same results if it is either done by additive or multiplicative model.



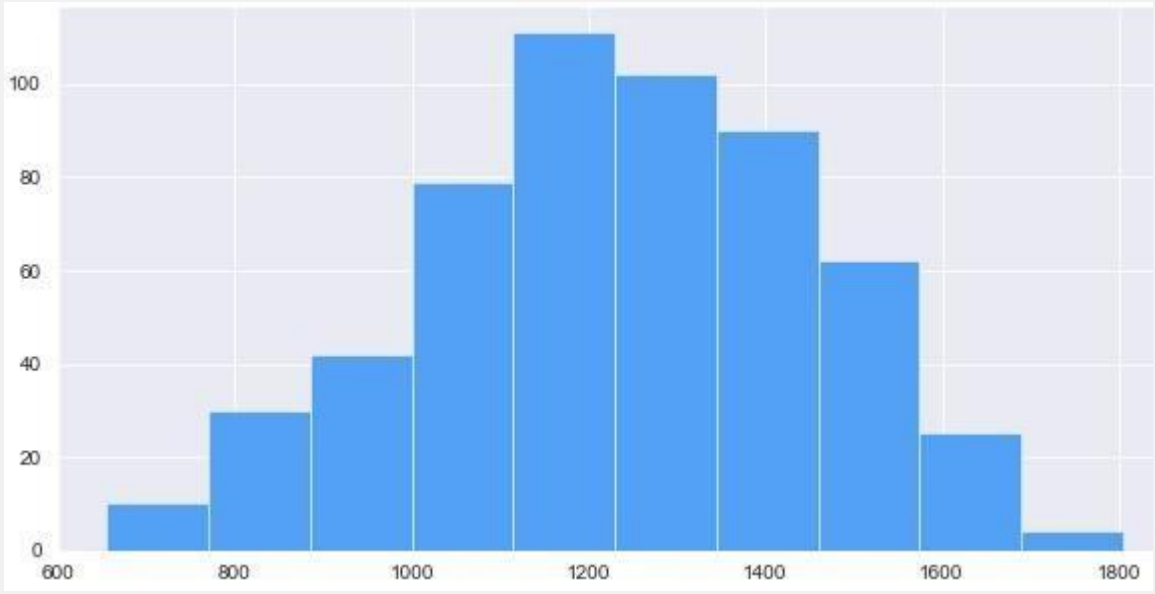
Trend: there is no trend (stationary trend)

Seasonality: It is estimated as an additive seasonality

Note: The decomposition model estimates the existence of a seasonality or a seasonal component, but this doesn't mean it has one (it is better to extract the seasonality from the original series in the cyan color).

Exploratory Data Analysis

Plotting Time Series Values as Histogram Graph

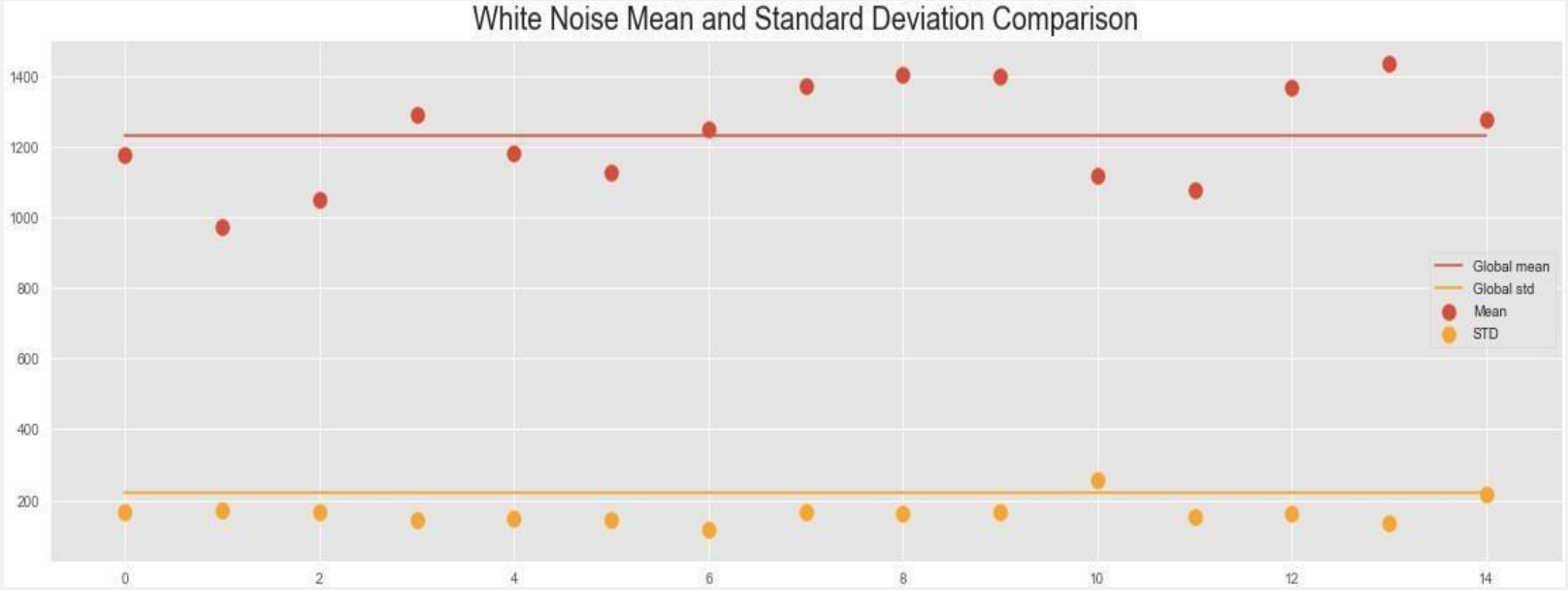


Plotting a histogram of the time series gives important clues into its underlying structure. A Normal distribution gives confidence that mean, and variance are constant. It's certainly not definitive but gives you a good indication.

Exploratory Data Analysis

Finding means & standard deviations of time series chunks:

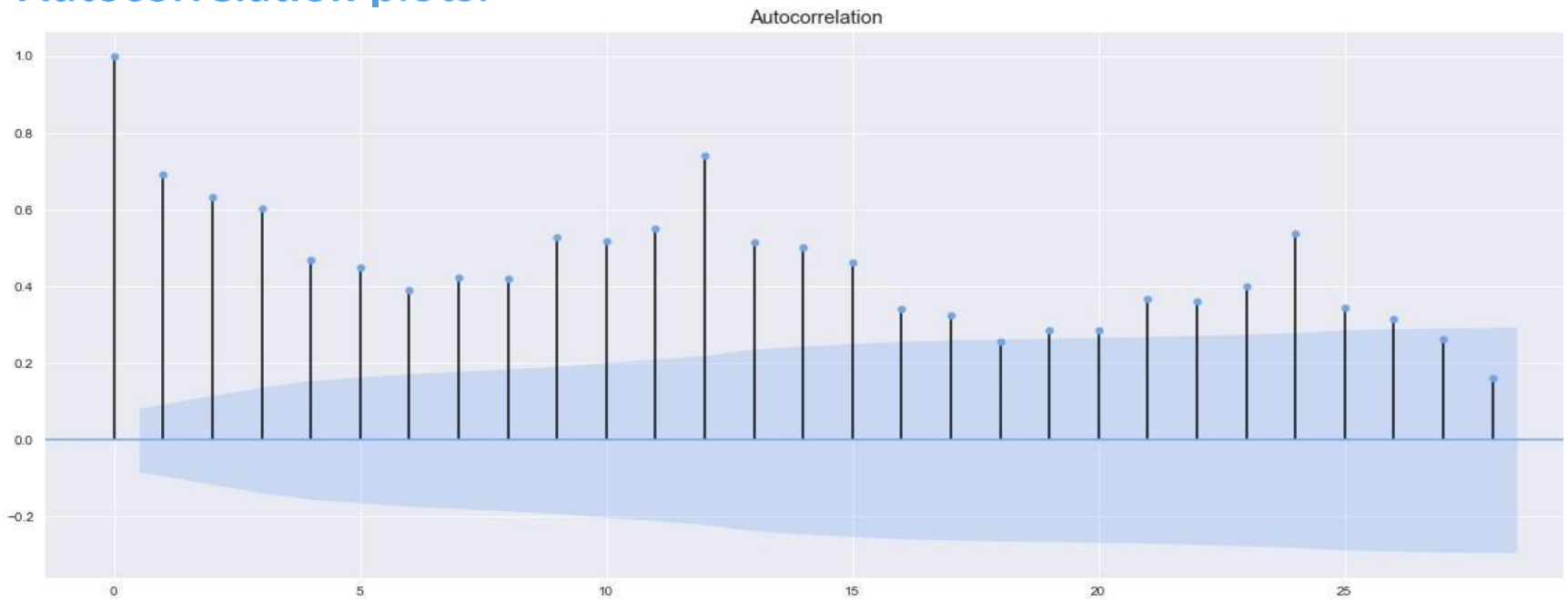
Chunk	Mean	Std
1	1175.11	167.12
2	972.916	172.35
3	1050.63	167.828
4	1291.27	144.99
5	1180.35	146.791
6	1128.57	143.983
7	1246.83	114.675
8	1368.84	165.205
9	1401.86	160.676
10	1396.98	165.801
11	1115.7	257.82
12	1074.49	153.788
13	1366.17	162.822
14	1434.6	132.686
15	1277.71	216.257



As shown all chunks have approximately closed mean values where they range between 1100 & 1400 (non-zero mean) which refers to a time series does not behave as white noise, standard deviations range mostly between 140 & 160 with existence of a little bit of variance.

Exploratory Data Analysis

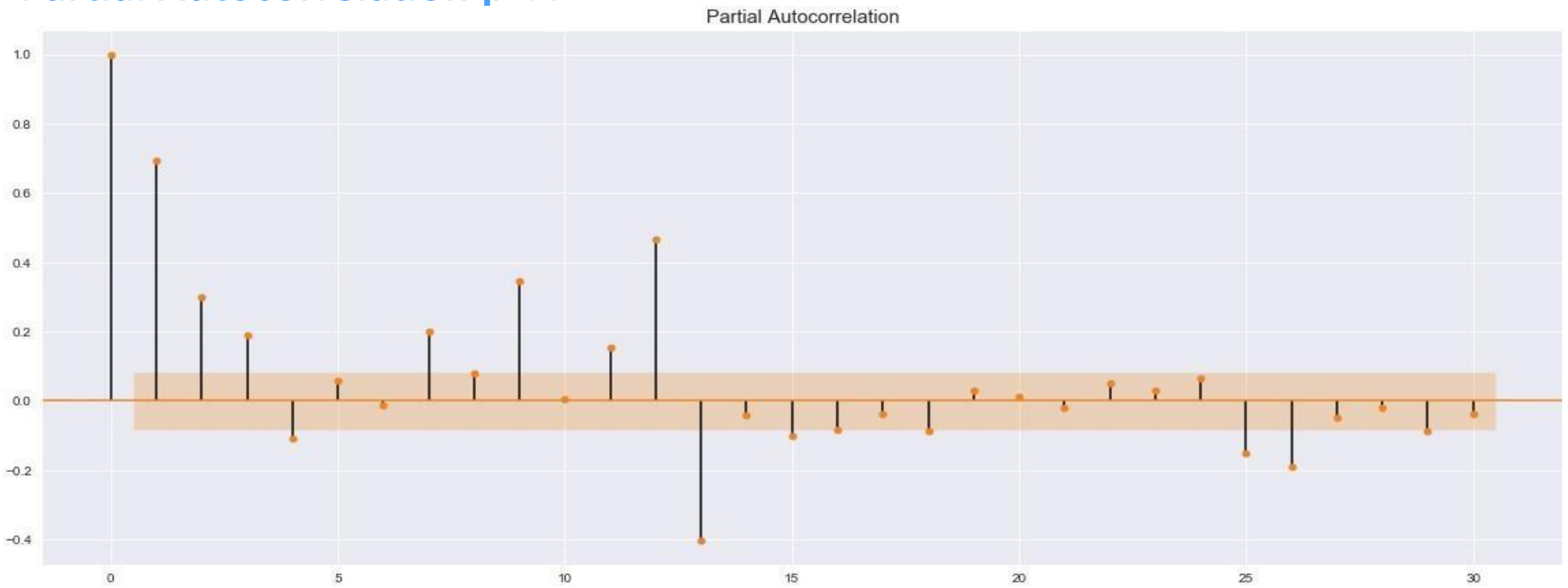
Autocorrelation plots:



As shown in the plot above the most majority of the correlations between the time series lags are statistically significant since it is out of the confidence interval (shaded area), and they are decreasing gradually.

Exploratory Data Analysis

Partial Autocorrelation plot



We have a moderate correlation between original time series and lag 12

Exploratory Data Analysis

Augmented Dickey-Fuller Test

This is a statistical procedure to discover whether a time series is stationary or not.

We won't go into all the nitty gritty details but here's what you need to know:

1. Null hypothesis: the series is nonstationary.
2. Alternative hypothesis: the series is stationary.

Like any statistical test you should set a significance level or threshold that determines whether you should accept or reject the null.

- The value 0.05 is common but depends upon numerous factors.

Let's see the result in the next slides.

Exploratory Data Analysis

Augmented Dickey-Fuller Test

```
from statsmodels.tsa.stattools import adfuller

adf, pvalue, usedlag, nobs, critical_values, icbest = adfuller(light_cars_sales['SALES'])
```

adf = -2.829277524541586

First, adf is the value of the test statistic. The more negative the value, the more confident we can be that the series is stationary. Here we see a value of -2.83. That may not mean anything to you just yet, but the p-value should. A brief discussion about the important outputs from the ADF test is in order. **Pvalue = 0.05421184907221919**

p-value is interpreted like any p-value. Once we set a threshold, we can compare this p-value to that threshold. Either we reject or fail to reject the null. Here p-value is very close to zero “0.054” so we reject the null that this data is nonstationary, and we can conclude that it is a stationary time series.

Exploratory Data Analysis

critical_values = {'1%': -3.442609129942274, '5%': -2.866947348175723, '10%': -2.569649926626197}

Finally, the critical_values variable provides test statistic thresholds for common significant levels. Here we see a test statistic of roughly -2.86 and lower is sufficient to reject the null using a significance level of 5%.

Analysis Summary:

White Noise:

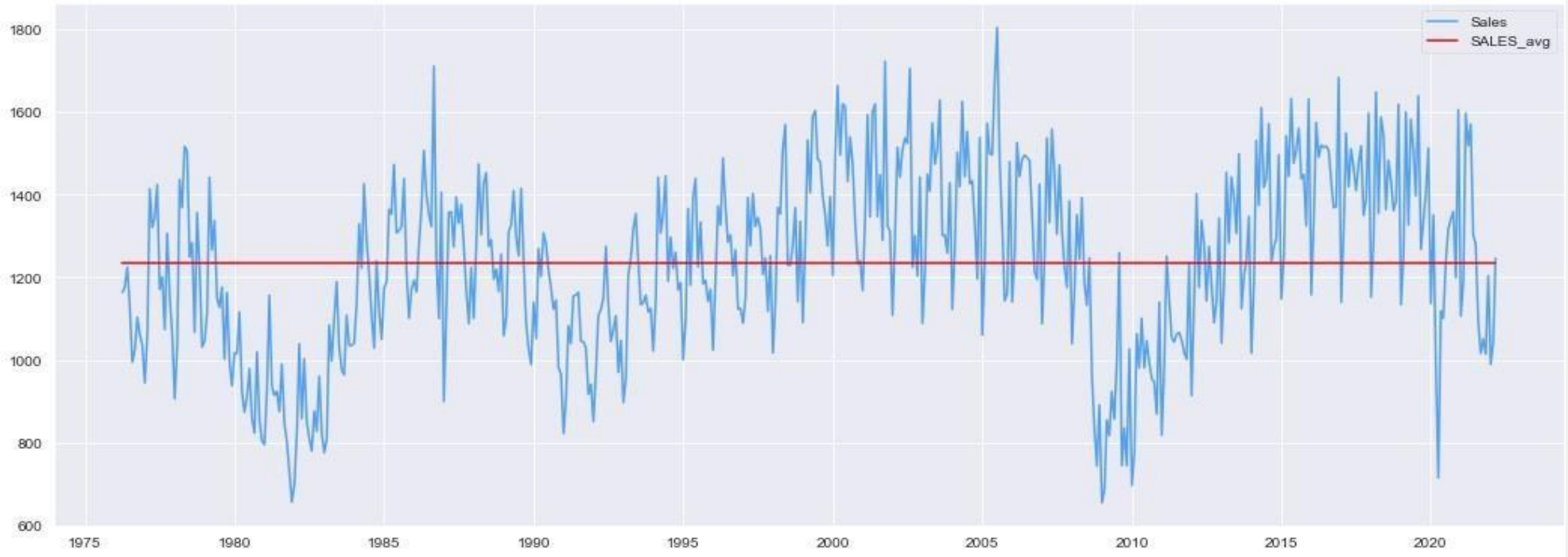
Time series can be considered as white noise (can't be modeled) if it satisfies three conditions :

1. **Approximately zero or exactly zero mean over the time series. [not satisfied]**
2. **Constant standard deviation over the time series. [satisfied]**
3. **correlations between the time series and its lags are not statistically significant. [not satisfied] Final Decision: Time series is not considered a noise.**

Feature Engineering

Smoothing Time Series

1 –Simple Smoothing

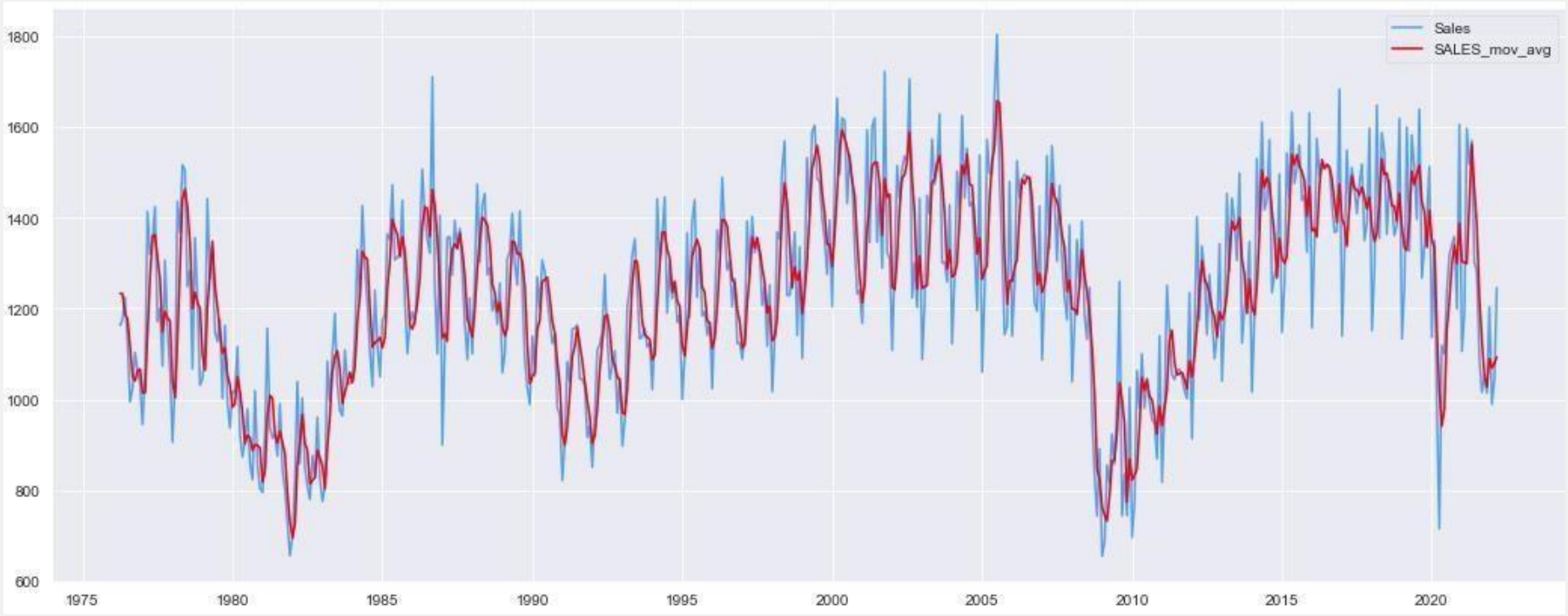



```
Mean Square Error: 26380517.872153617
```

Feature Engineering

Smoothing Time Series

2 – Moving Average Smoothing

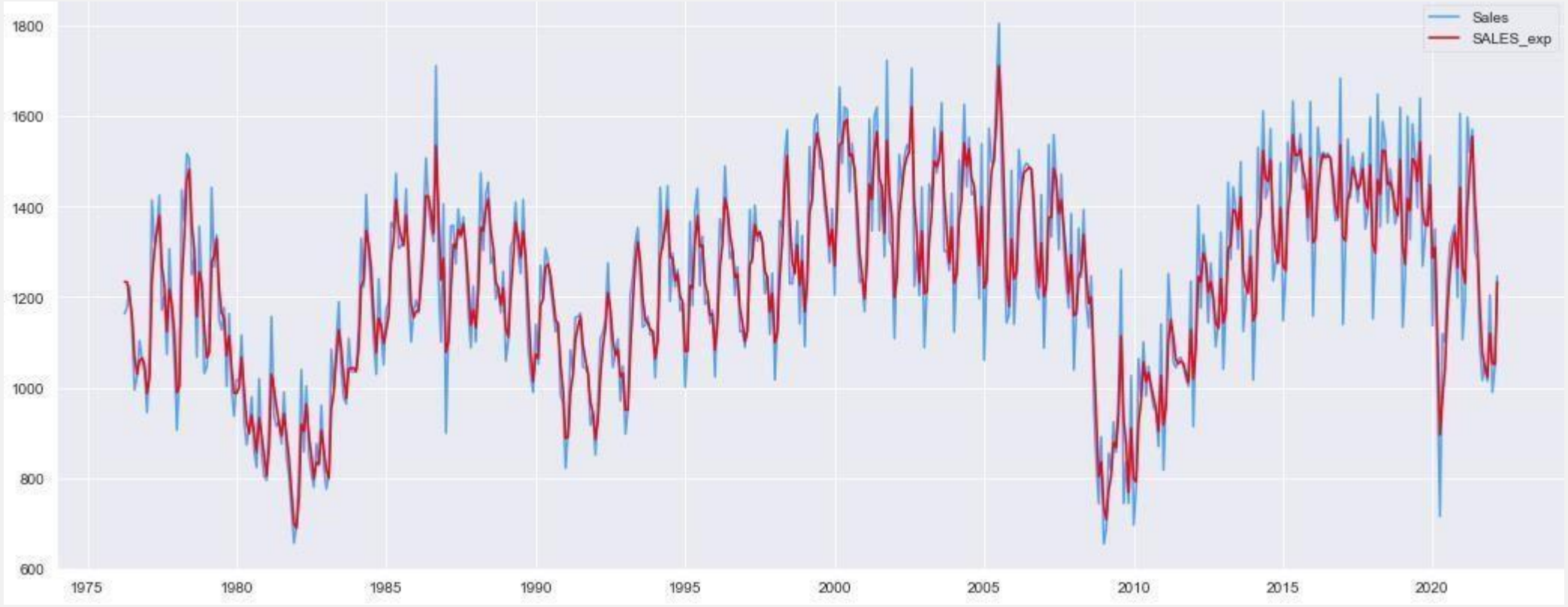



```
Mean Square Error: 6074348.431498482
```

Feature Engineering

Smoothing Time Series

3 – Exponential Smoothing




```
Mean Square Error: 2820822.767729523
```

Time series Forecasting & ML/DL Analysis and Findings

Time series Forecasting & ML/DL Analysis and Findings

1– Forecasting using Smoothing

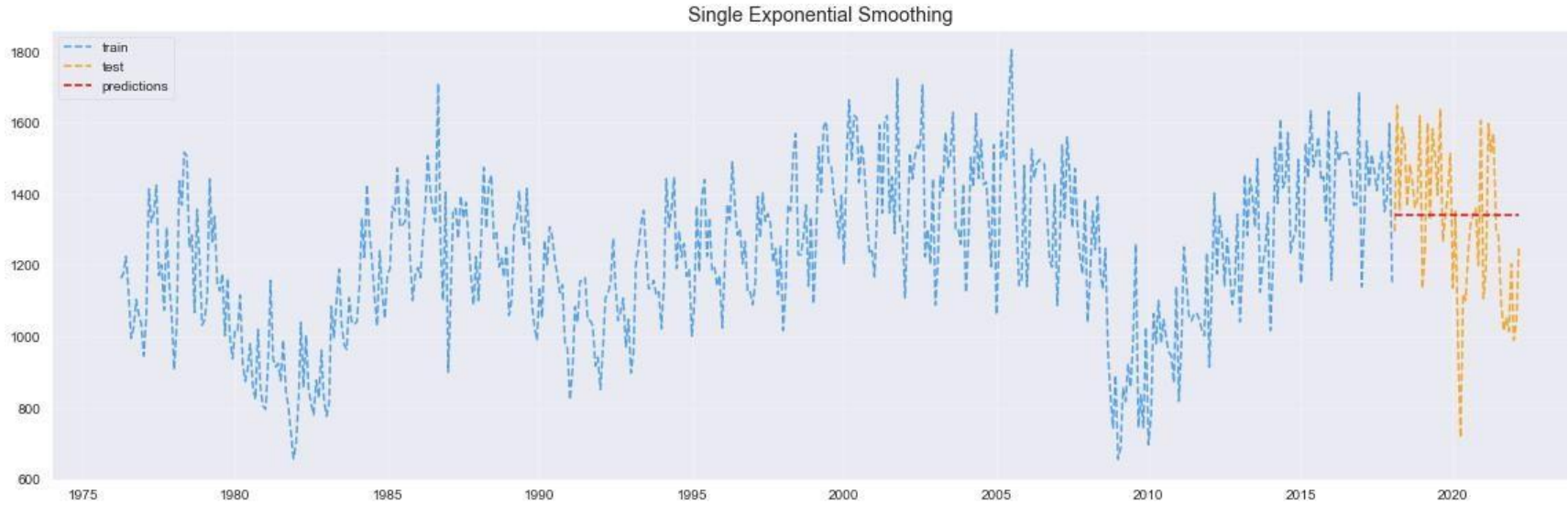
Splitting the data into train and test sets.

```
# Splitting the data
train = np.array(light_cars_sales['SALES'][:-50])
test = np.array(light_cars_sales['SALES'][-50:])
```

Training set : 502 observations.
Testing set : 50 observations.

1– Forecasting using Smoothing

1.2 Forecasting by Single Exponential



	Actual	Predictions
0	1294.694	1340.355
1	1647.975	1340.355
2	1354.484	1340.355
3	1587.518	1340.355
4	1544.695	1340.355
5	1363.887	1340.355
6	1483.169	1340.355
7	1432.993	1340.355
8	1361.288	1340.355
9	1383.388	1340.355

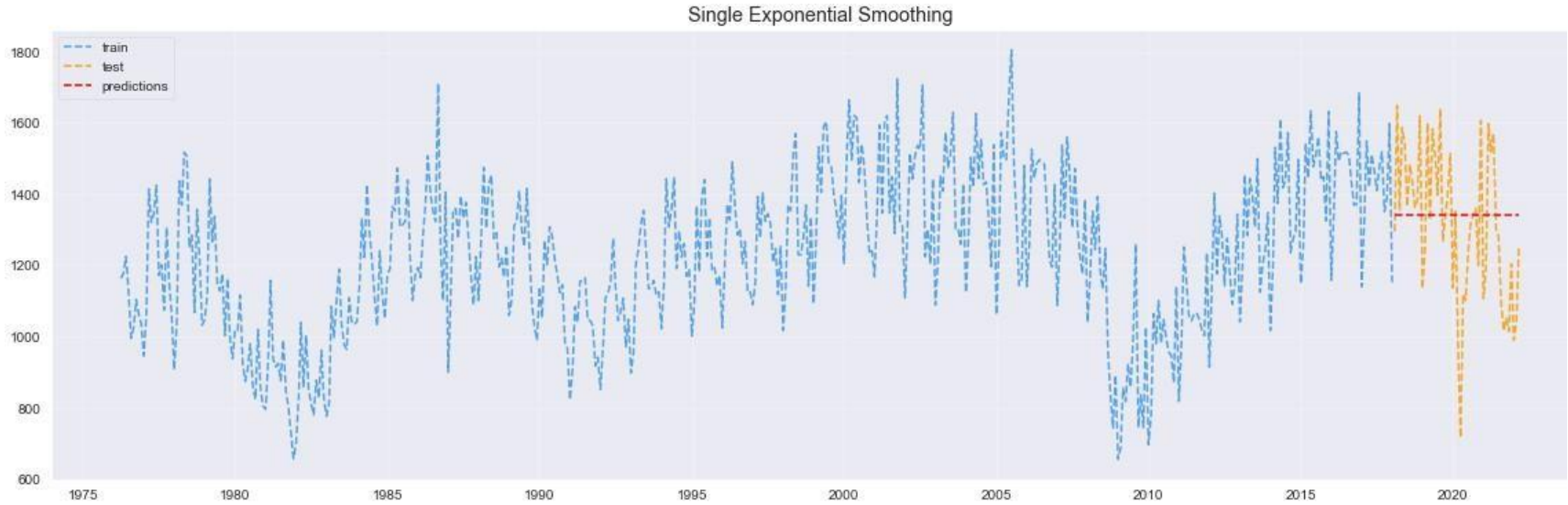
MSE: 2235283.269482

The results enhanced a little bit, but as the same time, we can see a high error relatively between the predictions and actual values.

Mean Square Error : 2235283 !

1– Forecasting using Smoothing

1.4 Forecasting by Triple Exponential



	Actual	Predictions
0	1294.694	1340.354723
1	1647.975	1340.354723
2	1354.484	1340.354723
3	1587.518	1340.354723
4	1544.695	1340.354723
5	1363.887	1340.354723
6	1483.169	1340.354723
7	1432.993	1340.354723
8	1361.288	1340.354723
9	1383.388	1340.354723

MSE: 2235282.558966974

Triple exponential technique achieved the same result of single exponential smoothing technique.

Mean Square Error : 2236924 !

Time series Forecasting & ML/DL Analysis and Findings

Comparison between smoothing techniques predictions.

As shown in the DataFrame on the right, single and triple exponential achieved the best results, where the worst forecasting was with double exponential smoothing, but all these forecasting techniques are considered unreliable since they led to very high error, in the next slides we will use better forecasting models and techniques.

	MSE
simple	2.537583e+06
single	2.236923e+06
double	5.197784e+06
triple	2.236924e+06

Time series Forecasting & ML/DL Analysis and Findings

Using Autocorrelation to choose appropriate model.

SHAPE	MODEL
Exponential Decaying to zero	
Alternating positive and negative decaying to zero	
One or more spikes, the rest are close to zero	MA
Decay after a few lags	MA
All zero or close to zero	Data is random
High values at fixed intervals	Include seasonal AR term
No decay to zero	Series is not stationary

Time series Forecasting & ML/DL Analysis and Findings

Using SARIMA model for forecasting.

Some rules to highlight from the Duke ARIMA Guide:

1. If the series has positive autocorrelations out to a high number of lags, then it probably needs a higher order of differencing.

2. If the lag-1 autocorrelation is zero or negative, or the autocorrelations are all small and pattern less, then the series does not need a higher order of differencing. If the lag-1 autocorrelation is 0.5 or more negative, the series may be over-differenced. BEWARE OF OVERDIFFERENCING!!

3. A model with no orders of differencing assumes that the original series is stationary (meanreverting). A model with one order of differencing assumes that

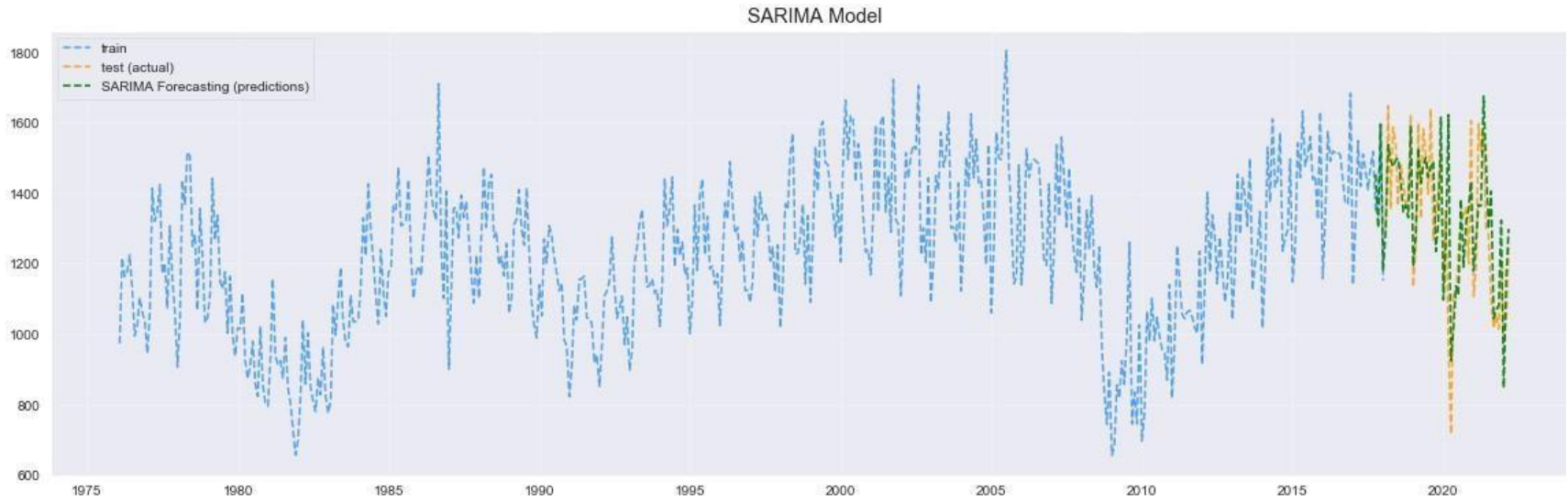
Time series Forecasting & ML/DL Analysis and Findings

the original series has a constant average trend (e.g. a random walk or SES-type model, with or without growth). A model with two orders of total differencing assumes that the original series has a time-varying trend (e.g. a random trend or LES-type model).

Time series Forecasting & ML/DL Analysis and Findings

2- Forecasting using SARIMA model (Seasonal Average Integrated Moving Average)

2.1 Forecasting using SARIMA model SARIMA (p, d, q) (P, D, Q)

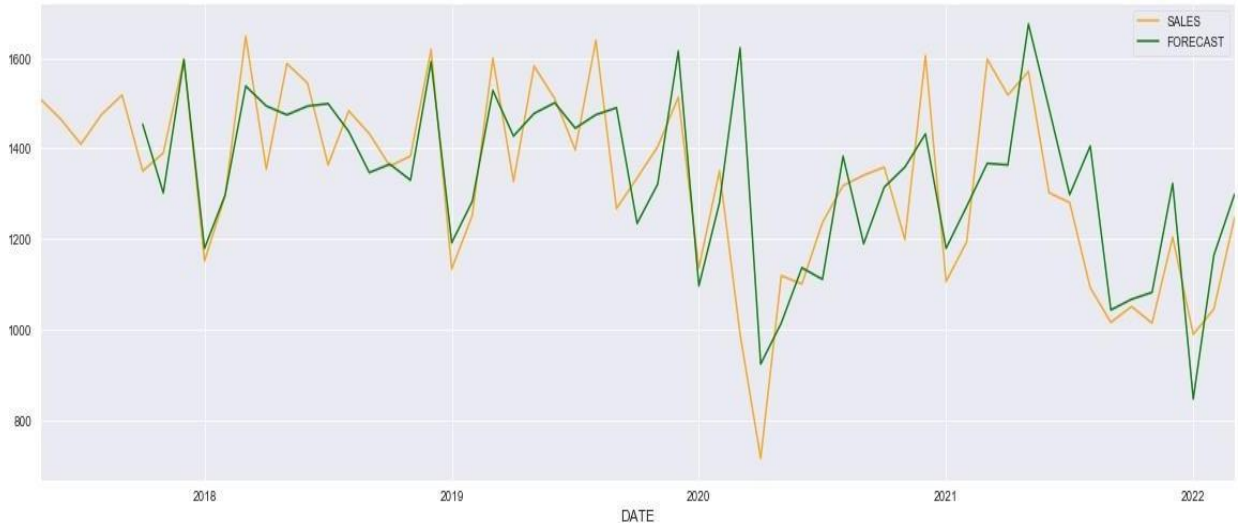



```
# fit SARIMA monthly based on helper plots
sar = sm.tsa.statespace.SARIMAX(light_cars_sales.SALES,
                                order=(1,0,0),
                                seasonal_order=(0,1,1,12),
                                trend='c').fit()
sar.summary()
```

As shown in the figure above we got much better predictions using SARIMA model

Time series Forecasting & ML/DL

Analysis and Findings



A closer look on last 50 values that forecasted using SARIMA model.

```
MSE:  1085655.1496082593
```

Mean Square Error : 1085655 !

```
 ARIMA(0,0,1)(0,1,1)[12] intercept   : AIC=6825.309, Time=0.63 sec
 ARIMA(0,0,0)(0,1,0)[12]             : AIC=7002.642, Time=0.02 sec
 ARIMA(1,0,1)(0,1,0)[12] intercept   : AIC=6735.274, Time=0.27 sec
 ARIMA(1,0,1)(1,1,1)[12] intercept   : AIC=6549.236, Time=2.23 sec
 ARIMA(1,0,1)(1,1,0)[12] intercept   : AIC=6669.500, Time=1.39 sec
 ARIMA(1,0,1)(2,1,1)[12] intercept   : AIC=6548.908, Time=4.51 sec
 ARIMA(1,0,1)(2,1,0)[12] intercept   : AIC=6629.749, Time=4.17 sec
 ARIMA(1,0,1)(2,1,2)[12] intercept   : AIC=inf, Time=8.11 sec
 ARIMA(1,0,1)(1,1,2)[12] intercept   : AIC=6550.034, Time=6.81 sec
 ARIMA(0,0,1)(2,1,1)[12] intercept   : AIC=inf, Time=5.52 sec
 ARIMA(1,0,0)(2,1,1)[12] intercept   : AIC=6609.090, Time=3.86 sec
 ARIMA(2,0,1)(2,1,1)[12] intercept   : AIC=6553.028, Time=7.07 sec
 ARIMA(1,0,2)(2,1,1)[12] intercept   : AIC=6544.209, Time=7.52 sec
 ARIMA(1,0,2)(1,1,1)[12] intercept   : AIC=6545.651, Time=3.61 sec
 ARIMA(1,0,2)(2,1,0)[12] intercept   : AIC=6629.567, Time=5.14 sec
 ARIMA(1,0,2)(2,1,2)[12] intercept   : AIC=inf, Time=9.02 sec
 ARIMA(1,0,2)(1,1,0)[12] intercept   : AIC=6669.053, Time=1.78 sec
 ARIMA(1,0,2)(1,1,2)[12] intercept   : AIC=6545.952, Time=8.77 sec
 ARIMA(0,0,2)(2,1,1)[12] intercept   : AIC=inf, Time=5.39 sec
 ARIMA(2,0,2)(2,1,1)[12] intercept   : AIC=6545.273, Time=7.70 sec
 ARIMA(1,0,3)(2,1,1)[12] intercept   : AIC=6545.800, Time=6.40 sec
 ARIMA(0,0,3)(2,1,1)[12] intercept   : AIC=inf, Time=5.59 sec
 ARIMA(2,0,3)(2,1,1)[12] intercept   : AIC=6547.926, Time=9.34 sec
 ARIMA(1,0,2)(2,1,1)[12]             : AIC=6543.426, Time=7.79 sec
 ARIMA(1,0,2)(1,1,1)[12]             : AIC=6543.901, Time=3.52 sec
 ARIMA(1,0,2)(2,1,0)[12]             : AIC=6627.575, Time=1.71 sec
 ARIMA(1,0,2)(2,1,2)[12]             : AIC=inf, Time=10.26 sec
 ARIMA(1,0,2)(1,1,0)[12]             : AIC=6667.055, Time=0.65 sec
 ARIMA(1,0,2)(1,1,2)[12]             : AIC=6544.345, Time=8.48 sec
 ARIMA(0,0,2)(2,1,1)[12]             : AIC=inf, Time=4.09 sec
 ARIMA(1,0,1)(2,1,1)[12]             : AIC=inf, Time=39.99 sec
...

Best model:  ARIMA(2,0,2)(2,1,1)[12]
Total fit time: 278.712 seconds
6543.363639474435
```

After about 5 mins of searching about the best SARIMAS Parameters that fits with our cars sales data we got the following results:

Best model: ARIMA(2,0,2)(2,1,1)[12]
Total fit time: 278.712 seconds **AIC:** 6543.363639474435

Now we are going to fit these parameters with a SARIMA model and test the model in terms of achieving correct forecasting.

Time series Forecasting &MVP Analysis and Findings

2– SARIMA model and parameters tuning


```
# fit SARIMA monthly based on helper plots
sar = sm.tsa.statespace.SARIMAX(light_cars_sales.SALES,
                                order=(2,0,2),
                                seasonal_order=(2,1,1,12),
                                trend='c').fit()
sar.summary()
```

```
MSE:  1261730.081798068
```

the previous model achieved a better results in terms of MSE compared to this model, but still the difference is between two models is so small.

Time series Forecasting & ML/DL Analysis and Findings

```
 ARIMA(0,0,1)(0,1,1)[12] intercept   : AIC=6825.309, Time=0.63 sec
 ARIMA(0,0,0)(0,1,0)[12]             : AIC=7002.642, Time=0.02 sec
 ARIMA(1,0,1)(0,1,0)[12] intercept   : AIC=6735.274, Time=0.27 sec
 ARIMA(1,0,1)(1,1,1)[12] intercept   : AIC=6549.236, Time=2.23 sec
 ARIMA(1,0,1)(1,1,0)[12] intercept   : AIC=6669.500, Time=1.39 sec
 ARIMA(1,0,1)(2,1,1)[12] intercept   : AIC=6548.908, Time=4.51 sec
 ARIMA(1,0,1)(2,1,0)[12] intercept   : AIC=6629.749, Time=4.17 sec
 ARIMA(1,0,1)(2,1,2)[12] intercept   : AIC=inf, Time=8.11 sec
 ARIMA(1,0,1)(1,1,2)[12] intercept   : AIC=6550.034, Time=6.81 sec
 ARIMA(0,0,1)(2,1,1)[12] intercept   : AIC=inf, Time=5.52 sec
 ARIMA(1,0,0)(2,1,1)[12] intercept   : AIC=6609.090, Time=3.86 sec
 ARIMA(2,0,1)(2,1,1)[12] intercept   : AIC=6553.028, Time=7.07 sec
 ARIMA(1,0,2)(2,1,1)[12] intercept   : AIC=6544.209, Time=7.52 sec
 ARIMA(1,0,2)(1,1,1)[12] intercept   : AIC=6545.651, Time=3.61 sec
 ARIMA(1,0,2)(2,1,0)[12] intercept   : AIC=6629.567, Time=5.14 sec
 ARIMA(1,0,2)(2,1,2)[12] intercept   : AIC=inf, Time=9.02 sec
 ARIMA(1,0,2)(1,1,0)[12] intercept   : AIC=6669.053, Time=1.78 sec
 ARIMA(1,0,2)(1,1,2)[12] intercept   : AIC=6545.952, Time=8.77 sec
 ARIMA(0,0,2)(2,1,1)[12] intercept   : AIC=inf, Time=5.39 sec
 ARIMA(2,0,2)(2,1,1)[12] intercept   : AIC=6545.273, Time=7.70 sec
 ARIMA(1,0,3)(2,1,1)[12] intercept   : AIC=6545.800, Time=6.40 sec
 ARIMA(0,0,3)(2,1,1)[12] intercept   : AIC=inf, Time=5.59 sec
 ARIMA(2,0,3)(2,1,1)[12] intercept   : AIC=6547.926, Time=9.34 sec
 ARIMA(1,0,2)(2,1,1)[12]             : AIC=6543.426, Time=7.79 sec
 ARIMA(1,0,2)(1,1,1)[12]             : AIC=6543.901, Time=3.52 sec
 ARIMA(1,0,2)(2,1,0)[12]             : AIC=6627.575, Time=1.71 sec
 ARIMA(1,0,2)(2,1,2)[12]             : AIC=inf, Time=10.26 sec
 ARIMA(1,0,2)(1,1,0)[12]             : AIC=6667.055, Time=0.65 sec
 ARIMA(1,0,2)(1,1,2)[12]             : AIC=6544.345, Time=8.48 sec
 ARIMA(0,0,2)(2,1,1)[12]             : AIC=inf, Time=4.09 sec
 ARIMA(1,0,1)(2,1,1)[12]             : AIC=inf, Time=39.99 sec
...

Best model:  ARIMA(2,0,2)(2,1,1)[12]
Total fit time: 278.712 seconds
6543.363639474435
```

After about 5 mins of searching about the best SARIMAS Parameters that fits with our cars sales data we got the following results:

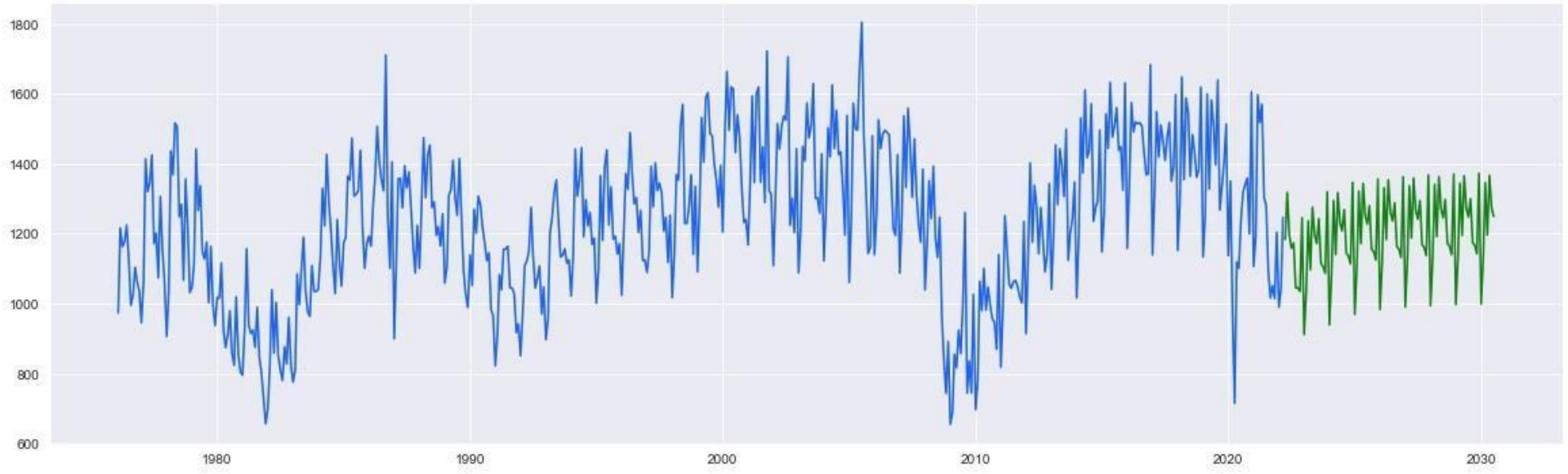
Best model: ARIMA(2,0,2)(2,1,1)[12]
Total fit time: 278.712 seconds **AIC:** 6543.363639474435

Now we are going to fit these parameters with a SARIMA model and test the model in terms of achieving correct forecasting.

Time series Forecasting & ML/DL Analysis and Findings

2– SARIMA model and parameters tuning

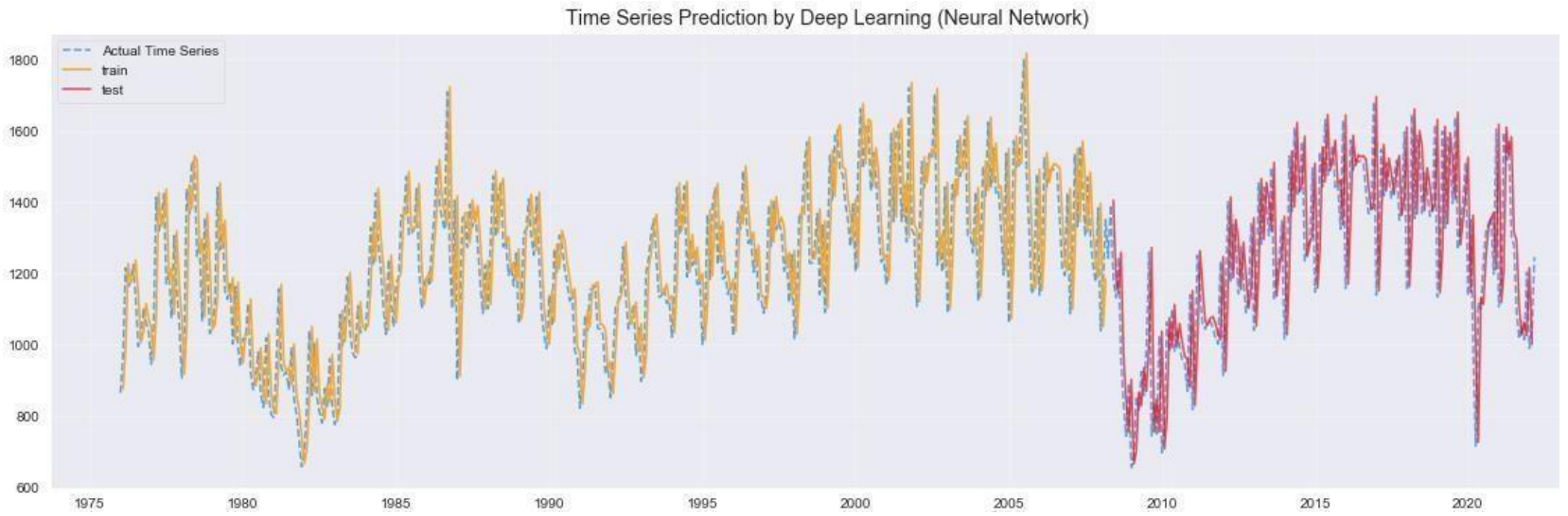
2.3 SARIMA Future Forecasting



Time series Forecasting & ML/DL Analysis and Findings

3– Time Series Prediction by Deep Learning (Neural Network)

3.1 Neural Network Prediction.




```
Train Score: 24400.56 MSE (156.21 RMSE)
Test Score: 41588.68 MSE (203.93 RMSE)
```

Models flaws and strengths and advanced steps

Models' strengths and flaws

Models Strengths and Flaws:

Generally, Time series modeling & forecasting is considered one of the hardest problems to approach in data science since it depends on large amount of analysis and work, as shown in the previous slides we exposed to several techniques, where we compare between them in terms of forecasting one of these techniques was forecasting by smoothing which provided the worst results but at the same time it gave us intuition that the time series can be predicted by more advanced techniques like SARIMA model which

provided very good results, and neural networks which gave us the best results, at the same time these models achieved good results to some extent in terms of forecasting future series.

Advanced steps

further suggestions:

As shown in the previous slides deep learning models achieved the best results in order of modeling and forecasting time series and we can go further with the results by using more advanced techniques such as :

- Recurrent neural network (RNN)

- Long short-term memory LSTM.

I am currently working on these models to find out its strengthens and weaknesses and will be uploaded on my GitHub account soon.

https://github.com/ArpanSankesh/ibm_machine_learning_coursera

Thank you

IBM Machine Learning Professional Certificate

Specialized Models: Time Series and Survival Analysis