Time Series Final Project

Deep dive into analysis and forecasting of pharmaceutical products sale

By: Hadi Nazari

May 2023

Executive Summary

Background Information - Critical Topics

Pharmaceutical companies operate in a dynamic market that is heavily influenced by various factors such as consumer demand, government regulations, and supply chain disruptions. Therefore, it is essential for these companies to have accurate sales forecasts to optimize production, inventory management, and pricing strategies. This Time Series Project uses Python to analyze and forecast pharmaceutical product sales, providing insights into the factors that drive sales and trends that impact the industry.

Key messages of the presentation

The Time Series Project provides a deep dive into the analysis and forecasting of pharmaceutical product sales using Python. Through this project, we can:

- Analyze the historical sales data and identify trends and patterns that impact the industry.
- Use time series models such as ARIMA, SARIMA, and Prophet to forecast future sales accurately.
- Identify factors that influence sales, such as seasonality, trends, and other factors.
- Optimize production, inventory management, and pricing strategies based on sales forecasts and market trends.

Results / Recommended decisions

By analyzing and forecasting pharmaceutical product sales, we can gain valuable insights into market trends and make data-driven decisions. Based on these insights, we recommend the following decisions:

- Use accurate sales forecasts to optimize production and inventory management, ensuring that the right products are available when customers need them.
- Develop targeted marketing campaigns and pricing strategies based on the analysis of historical sales data to drive revenue growth.
- Monitor market trends and adjust strategies to stay ahead of competitors and adapt to changing consumer demand.
- Stay informed about government regulations and other external factors that may impact the industry and adjust strategies accordingly.

Agenda

- Analytics workflow
- Data overview
- Statistical analysis results
- AI modelling results and recommendations
- Conclusion and next steps

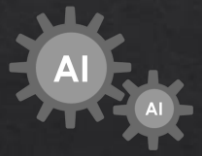
Analytics workflow

Data Extraction

- Extracting data
- Understanding features
- Understating target

Data Pre-processing

- Preparing data
- Identifying data structure for modeling
- Statistical Analysis of data



AI Modeling

Applying:

- ARIMA
- SARIMA
- Facebook Prophet
- Deep Learning (RNN-LSTM)

Evaluation

- Forecasting results
- Evaluate different methods
- Choose the best model

Reporting & Generalization

- Key findings
- Possible flaws in the model and a plan of action
- Reporting the results
- Generalization

Intro

- The analysis and forecasting of pharmaceutical product sales using Python provides a powerful tool for companies to gain insights into market trends and make data-driven decisions that drive business growth. By using accurate sales forecasts, pharmaceutical companies can optimize production, inventory management, and pricing strategies, ensuring that they stay competitive in a dynamic market.
- In this project we are trying to explore a dataset of pharmaceutical product sales in Iran from 2019 to 2023 using time series models in python, in order to forecast the sale for upcoming periods.

Data Overview

- 127k records of a strategic pharmaceutical product sale were collected from 2019 to 2023
- Frequency: At first hourly but after preprocessing transformed to monthly.
- Company: Barij Essence https://barijessence.com/en/

The dataset consist of two columns:
Date (datetime64[ns]) : date from 2019-03-20 to 2023-03-20
Sale (float64) : quantity of sales

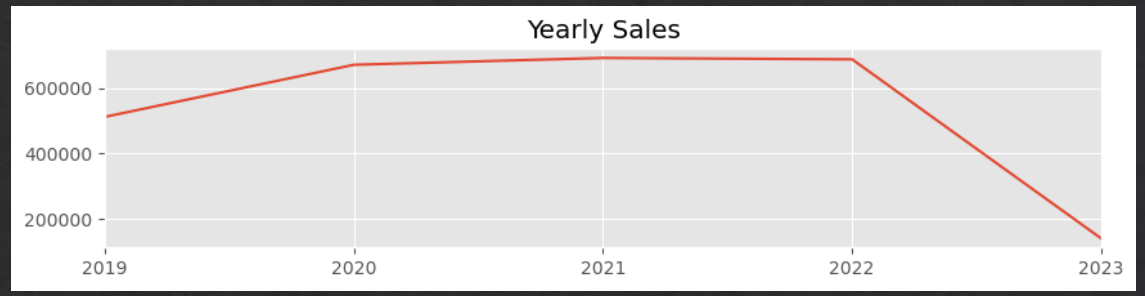
Target: Pre-processing to extract features and understating patterns or relations of features

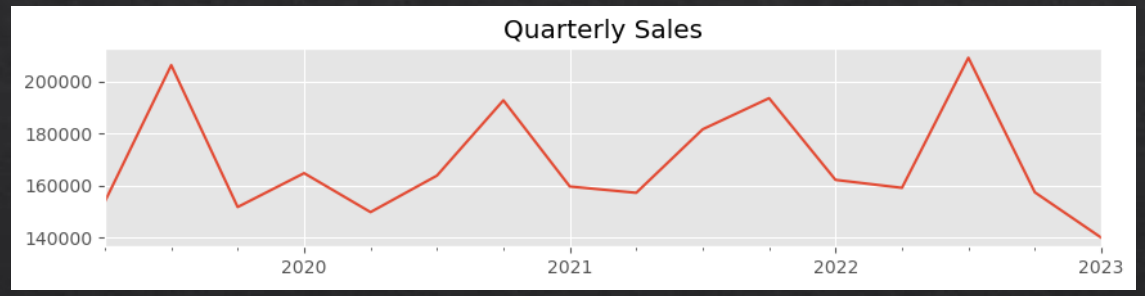
	date	sale
0	2019-04-06 10:54:36.120	6.0
1	2019-04-06 11:52:27.140	6.0
2	2019-04-06 14:00:06.110	5.0
3	2019-04-06 14:15:40.650	24.0
4	2019-04-06 14:29:10.087	6.0

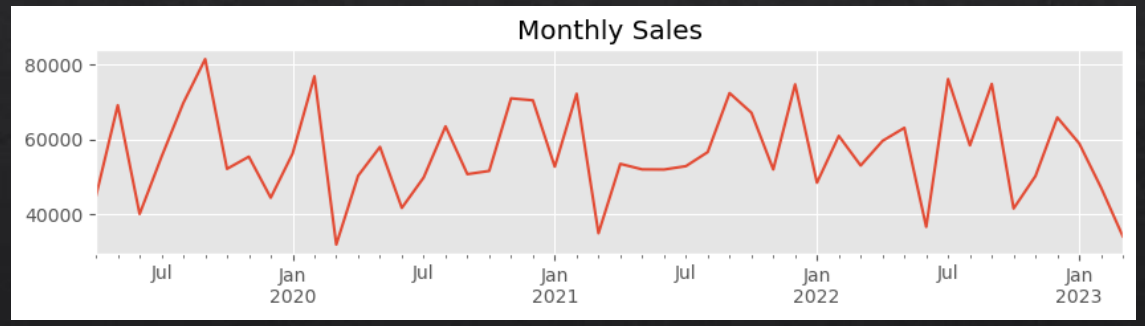
	sale
count	127455.000000
mean	21.209376
std	74.175624
min	-2000.000000
25%	5.000000
50%	10.000000
75%	24.000000
max	4000.000000

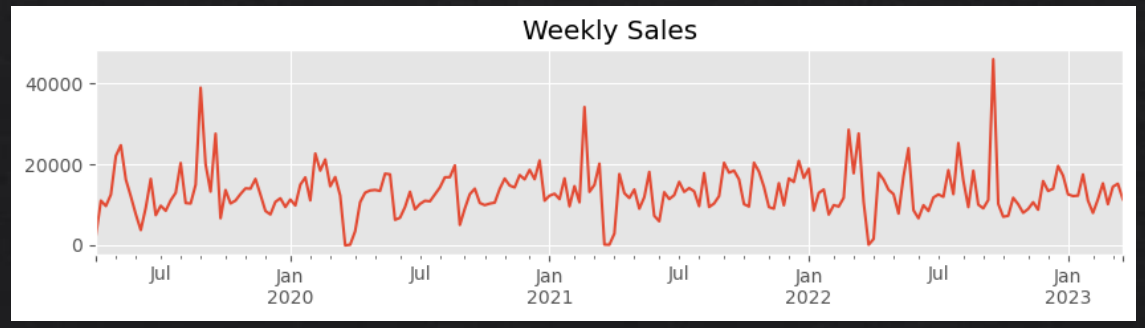
Feature Engineering and EDA

- After preprocessing we use “ Resampling” techniques to get the dataset into other periods like weekly, monthly, quarterly and yearly.





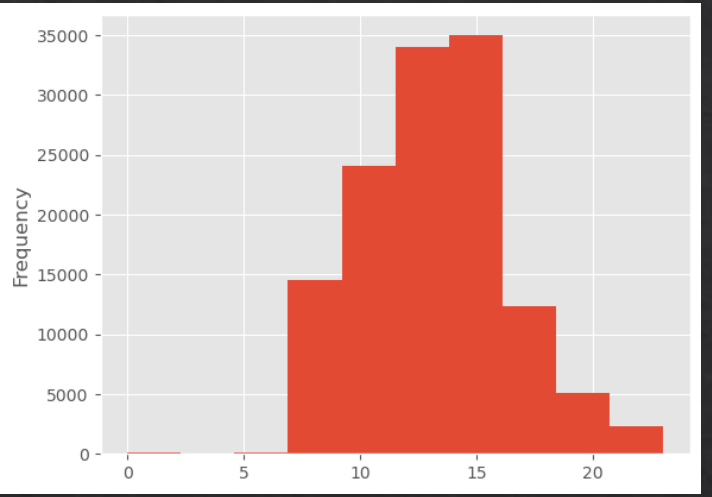


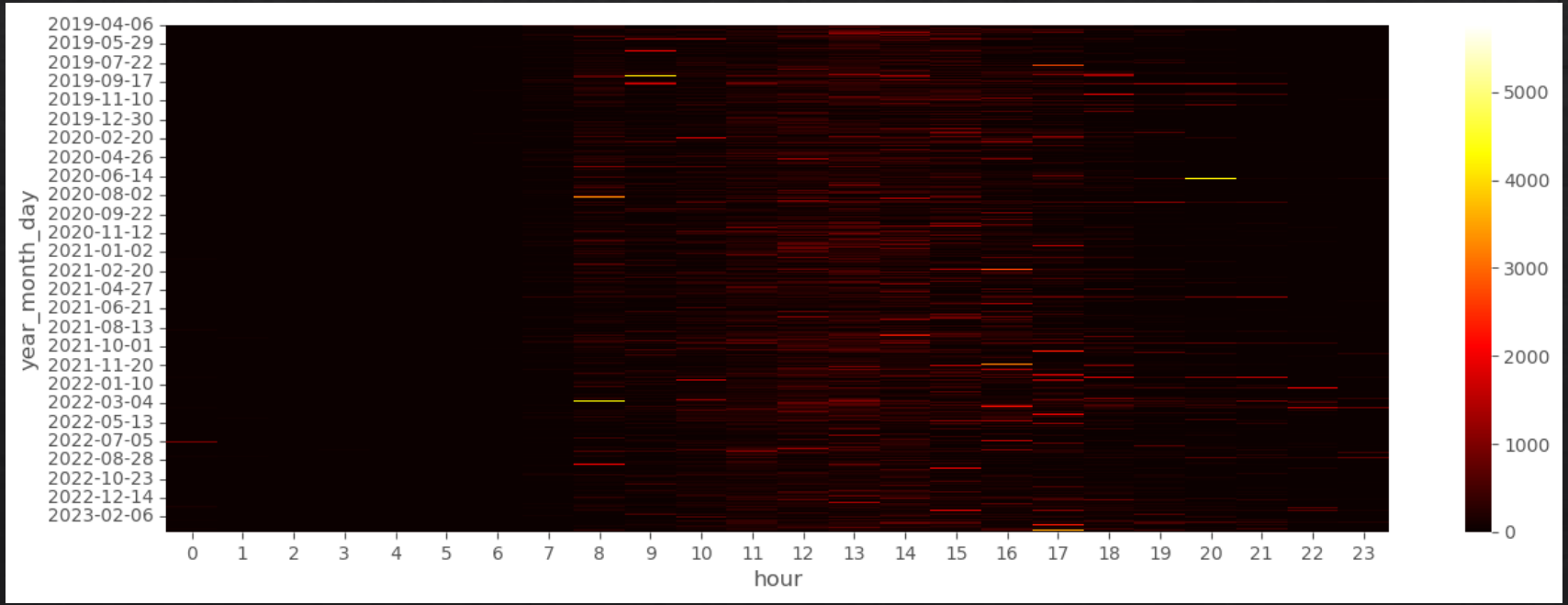


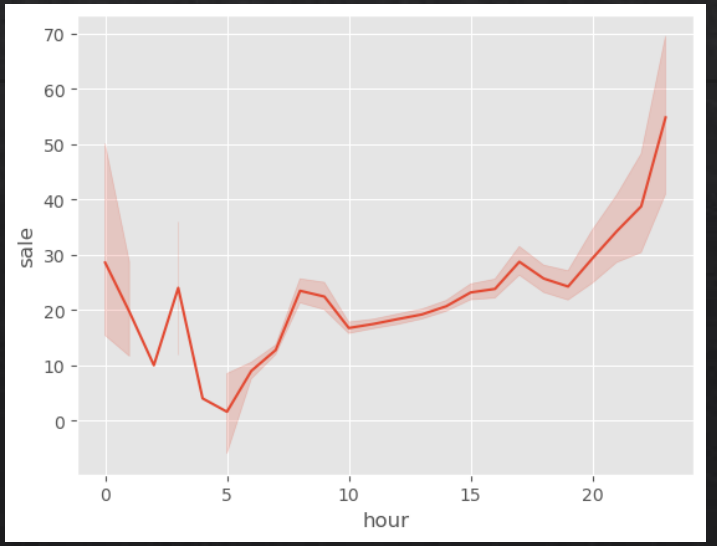
Statistical analysis

We can get some insights through this dataset:

- The highest amount of sales occurred in the last hours of the day.
- But the most frequency of purchases occurred in the middle of the day between 11-15.







Statistical Analysis

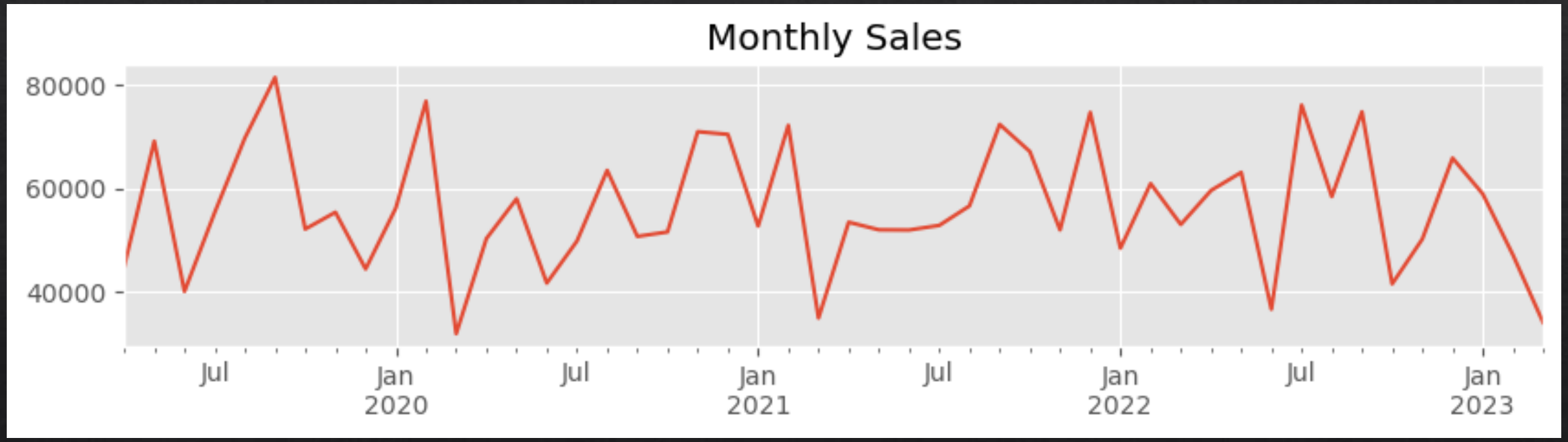
- We are going to apply different analysis and statistical tests to check the Stationarity in the data set:

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

- **constant mean**
- **constant variance**
- **constant autocorrelation structure**
- **no periodic component or seasonality**

Statistical Analysis

Time Series Visualization

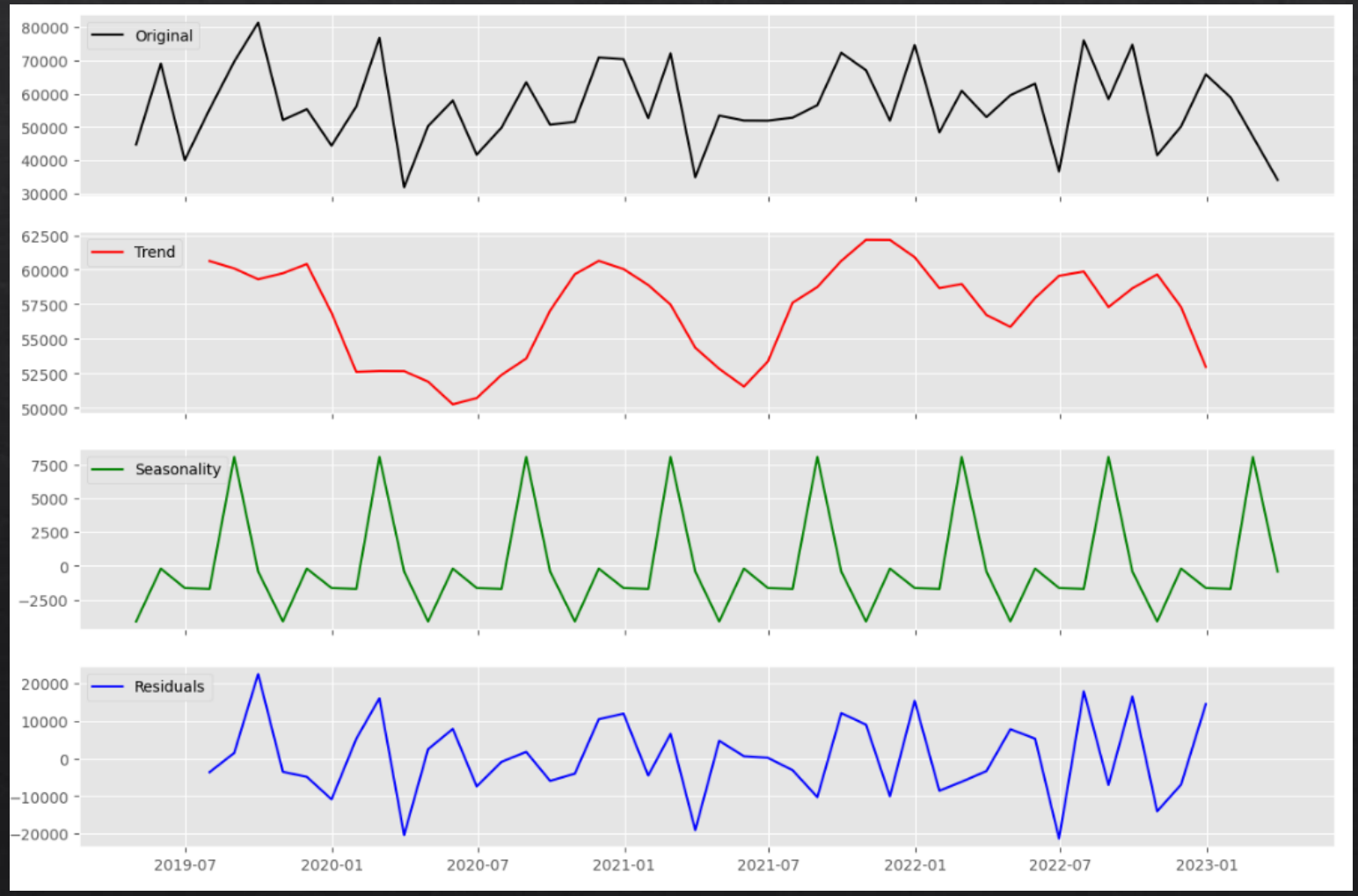


we can extract from the graph the following features:

1. There is no trend in other words Stationary trend which indicates a constant mean.
2. The variance is constant.
3. The graph does not show a periodic component (no seasonality)

Statistical Analysis

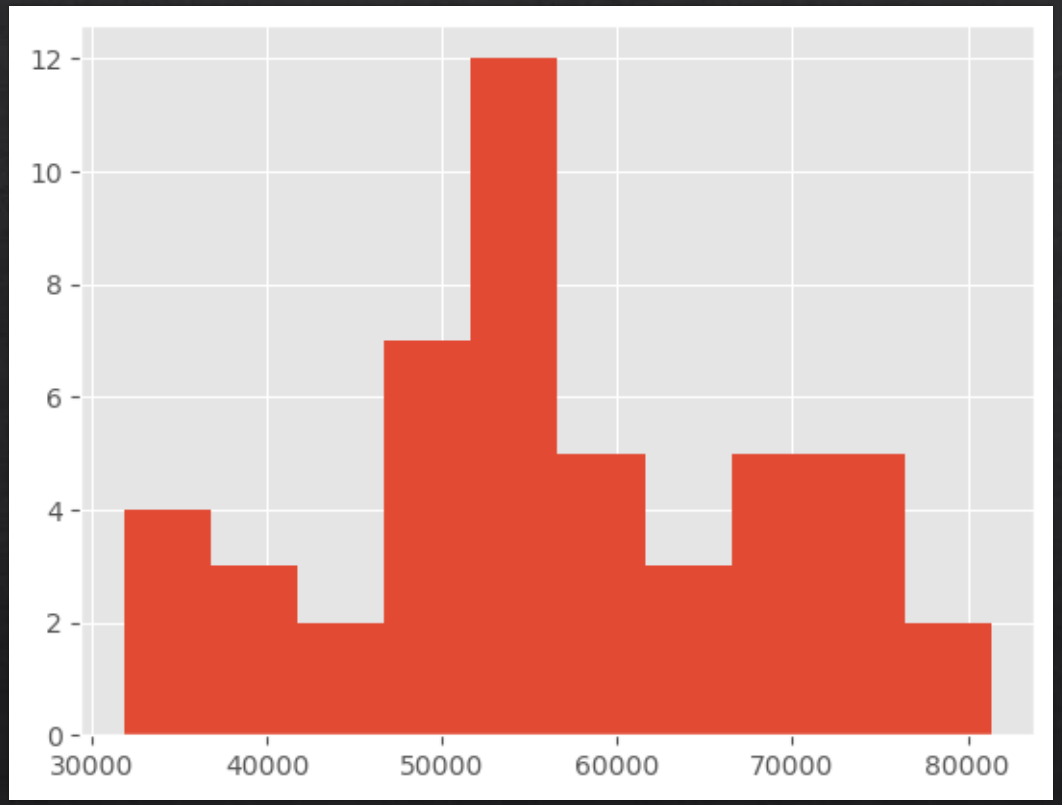
Time Series Decomposition



There is no trend

Statistical Analysis

Time Series normality




```
#Test normal distribution
def normaltest_stat(data,alpha=0.05):
    k2, p_value = normaltest(data)
    if p_value < alpha:
        print("The null hypothesis can be rejected -  data is not normally distributed  ")
    else:
        print("The null hypothesis cannot be rejected -data comes from a normal distribution")
```

```
normaltest_stat(sales_monthly)
```

```
The null hypothesis cannot be rejected -data comes from a normal distribution
```

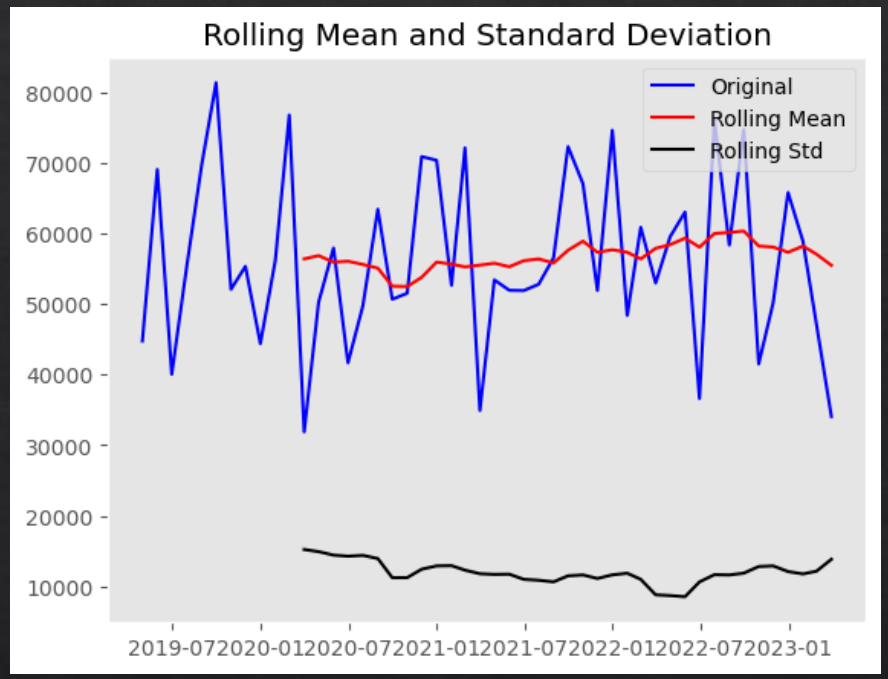
The result of test approve normal distribution of our dataset

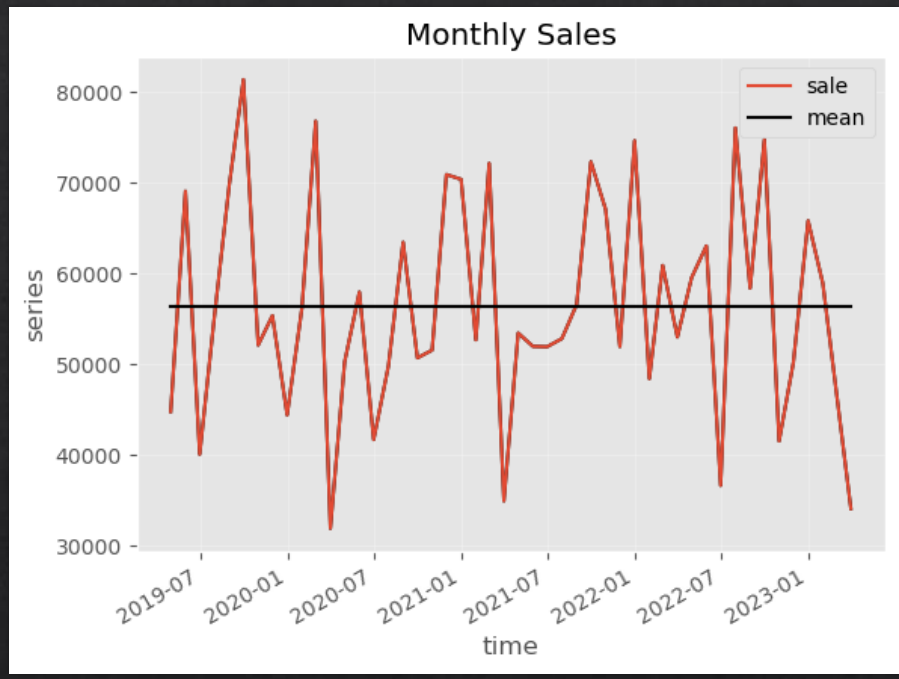
A Normal distribution gives confidence that mean, and variance are constant. It's certainly not definitive but gives you a good indication

Statistical Analysis

Finding means & standard deviations of time series chunks and rolling:

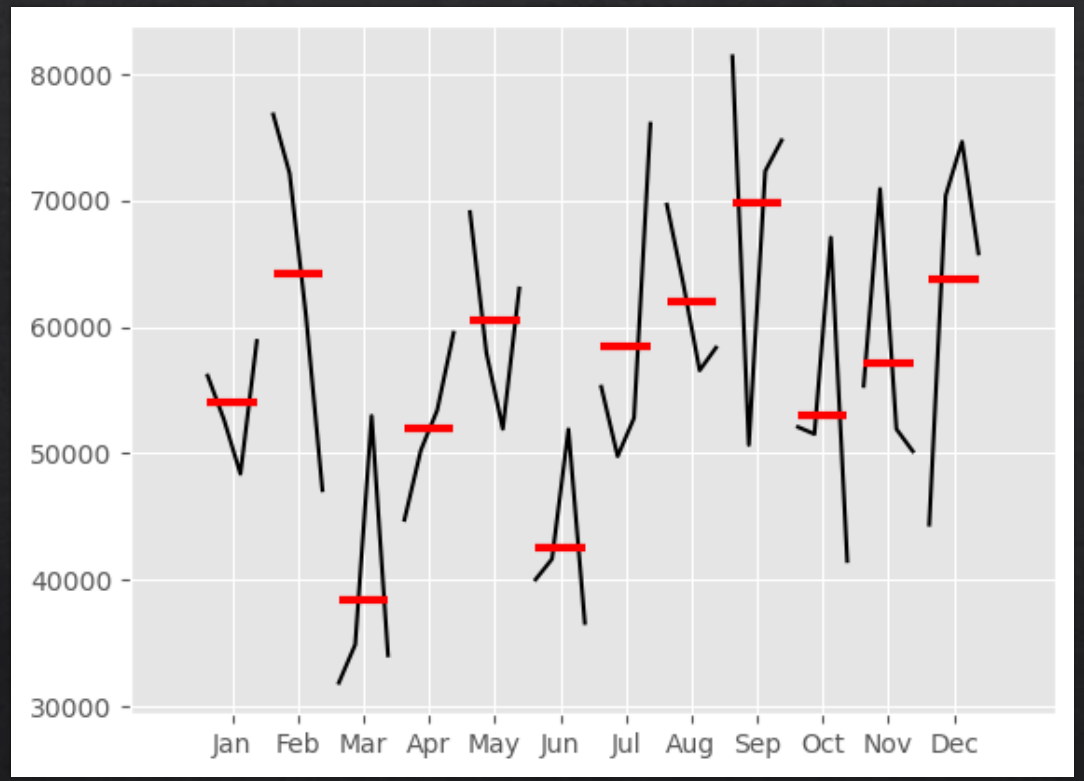
Chunk	Mean	Variance
1	60018.8	2.1415e+08
2	52762.0	1.83611e+08
3	52280.8	4.69928e+07
4	58740.5	1.86494e+08
5	56481.8	5.24711e+07
6	59299.2	8.51783e+07
7	61378.5	1.70346e+08
8	49578.5	1.10379e+08

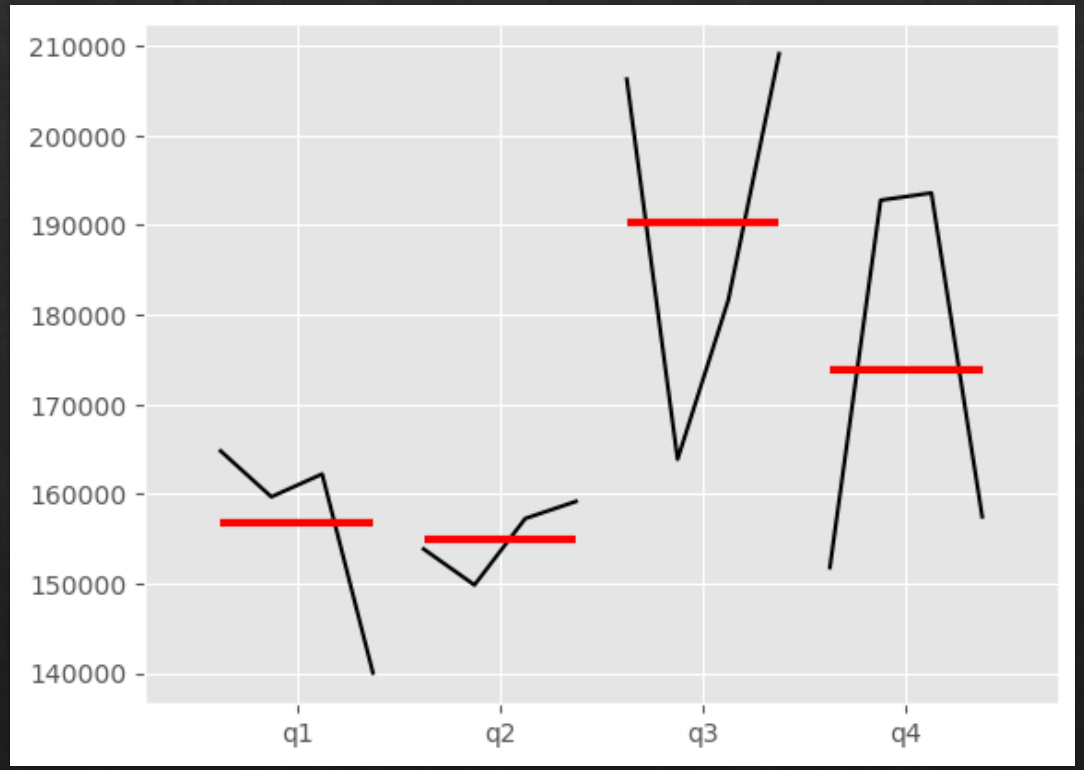




All of the result show the constant mean and variance in dataset

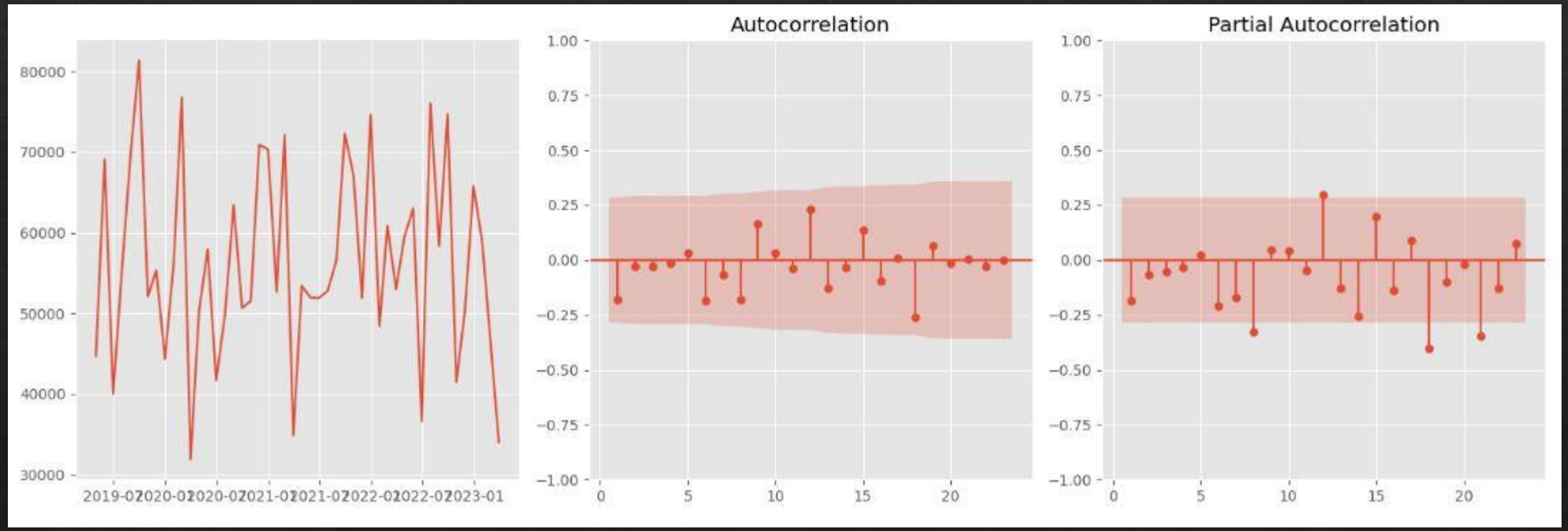
Statistical Analysis





There is no seasonality

Statistical Analysis



Based on ACF and PACF plot , we don't have autocorrelation in the data set and as the patterns shows alternating positive and negative lags that decaying to zero, AR model would be suitable for it.

Statistical Analysis

Augmented Dickey-Fuller Test

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

- Null hypothesis: the series is nonstationary.
- Alternative hypothesis: the series is stationary.

```
#Augmented Dickey-Fuller Test

def dftest(timeseries):
    dftest = ts.adfuller(timeseries,)
    dfoutput = pd.Series(dftest[0:4],
                         index=['Test Statistic','p-value','Lags Used','Observations Used'])
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print(dfoutput)
    #Determing rolling statistics
    rolmean = timeseries.rolling(window=12).mean()
    rolstd = timeseries.rolling(window=12).std()

    #Plot rolling statistics:
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean and Standard Deviation')
    plt.grid()
    plt.show(block=False)
```

```
dftest(sales_monthly)
```

```
Test Statistic          -7.948379e+00
p-value                 3.174423e-12
Lags Used               0.000000e+00
Observations Used       4.700000e+01
Critical Value (1%)     -3.577848e+00
Critical Value (5%)     -2.925338e+00
Critical Value (10%)    -2.600774e+00
dtype: float64
```

Based on the ADF test, we can reject the null hypothesis, so there's no autocorrelation in the dataset.

Analysis Summary

Stationarity:

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

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

- **Final Decision: Time series is considered stationary.**

AI modeling

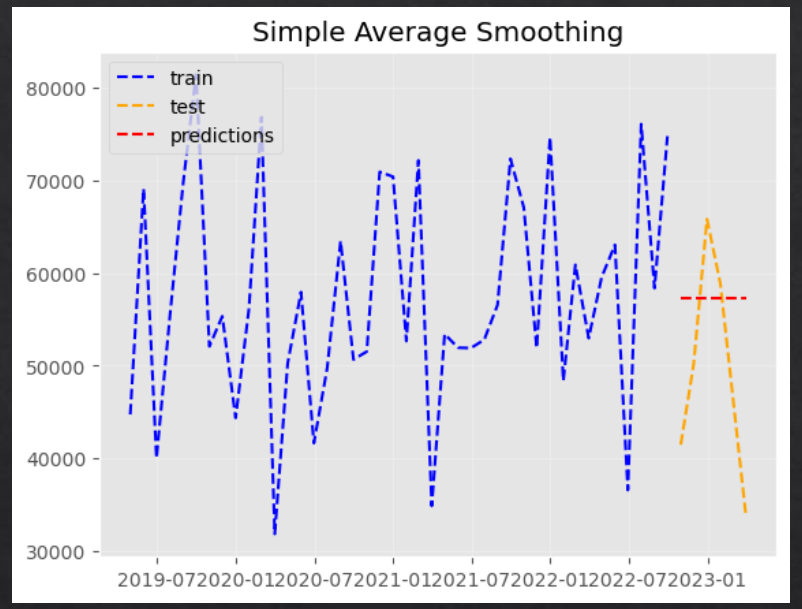
In the upcoming slides, first we split our dataset into train and test and after that we run these models. Then we evaluate the MSE of model to choose the best model for forecasting.

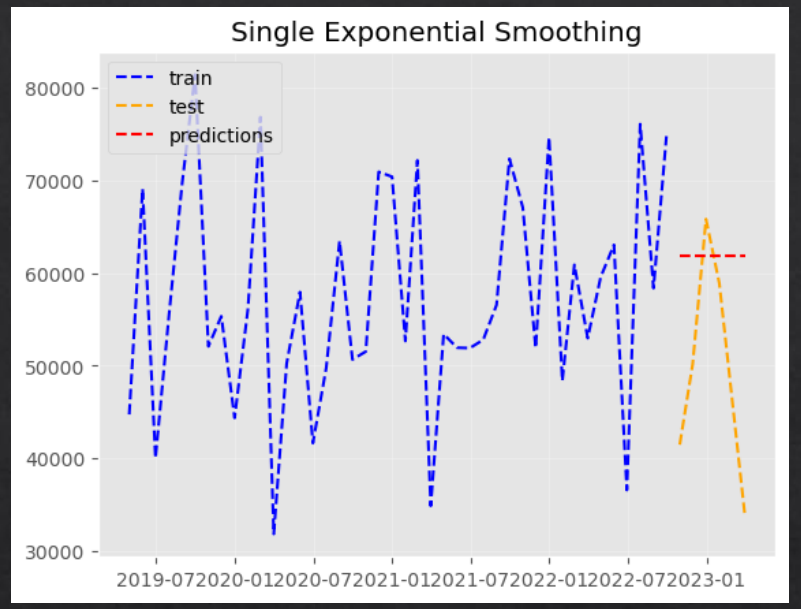
- Simple Average Smoothing
- Double Exponential
- Triple Exponential
- AR
- MA
- ARMA
- ARIMA
- SARIMA
- Facebook Prophet
- RNN
- LSTM

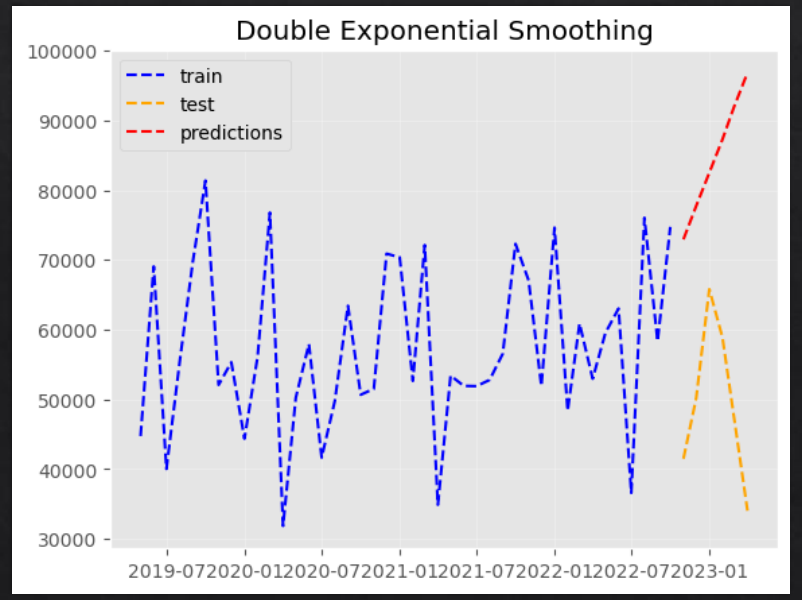
Smoothing methods

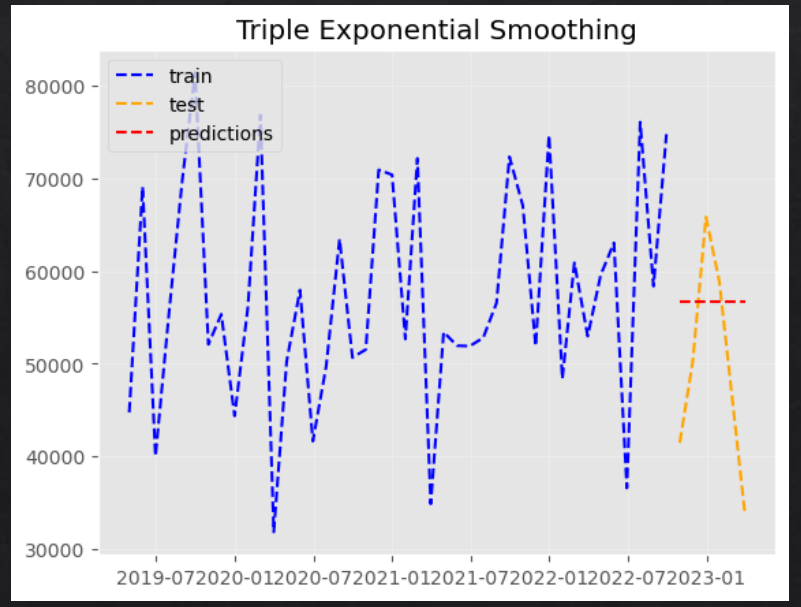
As the plots show, we can't get good results through the smoothing methods.

	MSE
triple	1.609368e+08
simple	1.696956e+08
single	2.592503e+08
double	1.465818e+09



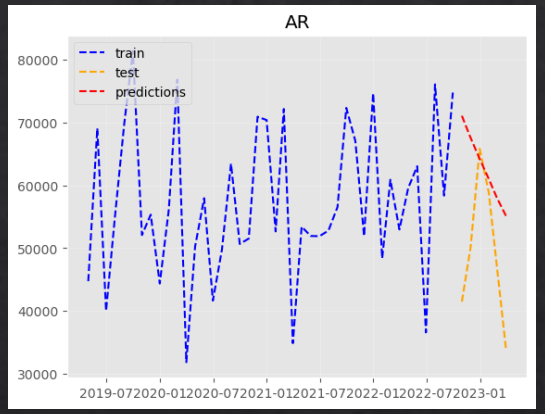


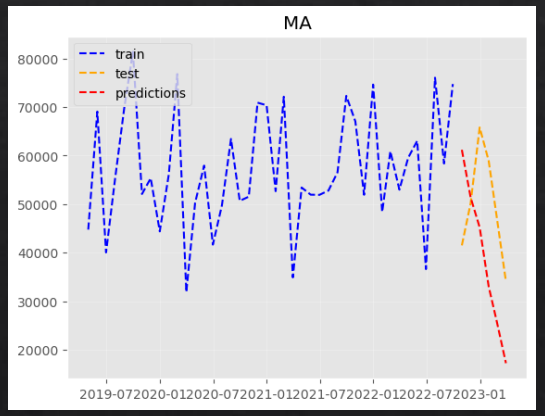




AR-MA-ARMA-ARIMA-SARIMA

The result show that we get the least MSE from ARIMA model
ARIMA (4,1,7)

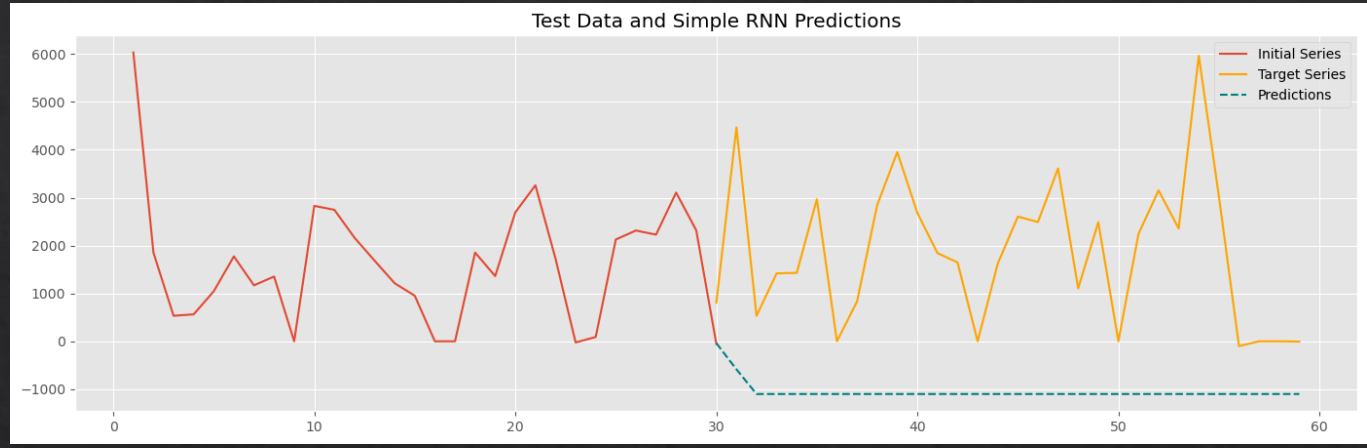


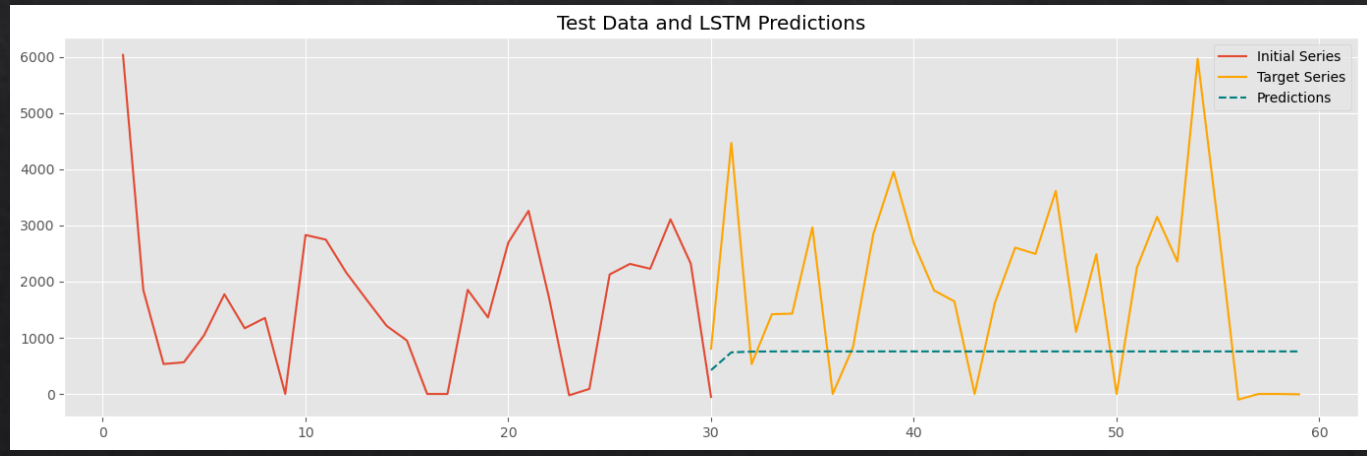


	MSE
ARIMA	9.371654e+07
ARMA	1.601739e+08
AR	2.908862e+08
MA	3.705213e+08
SARIMA	7.838969e+08

RNN-LSTM

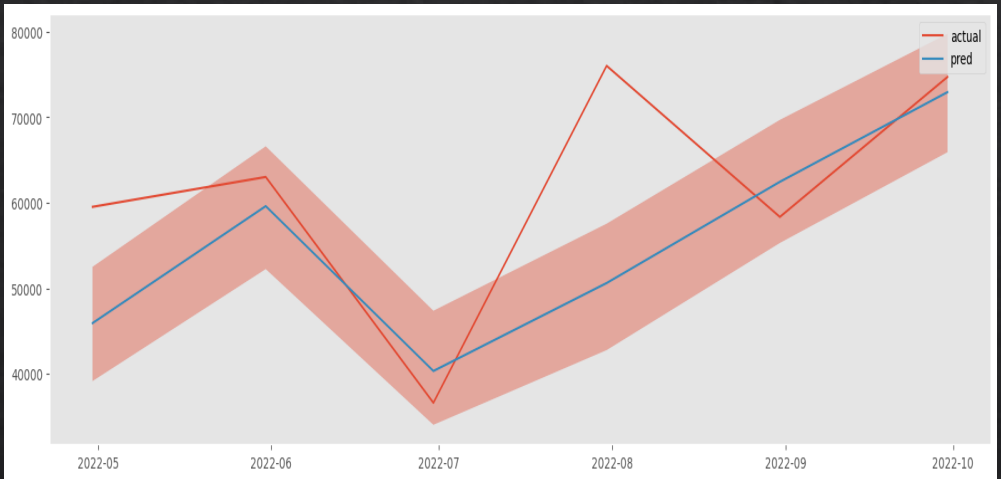
- Based on the fact that we need more observations to train in deep learning models, we can't get appropriate results through them.
- We can train daily observations instead.
- RNN and LSTM work poorly on the model.

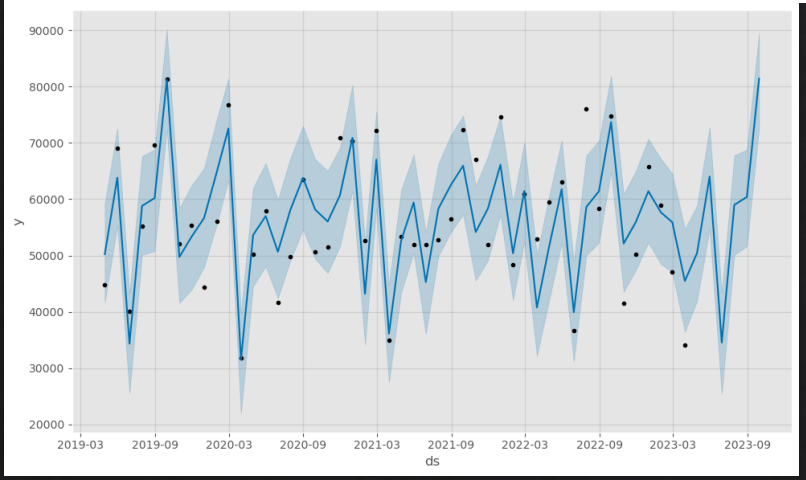




Facebook Prophet

We get just 20% mean absolute percentage error from Facebook Prophet model. It's the best result among other models.





	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	25 days	1.850908e+08	13604.806480	13604.806480	0.228537	0.228537	0.258021	0.0
1	27 days	5.710841e+08	23897.366395	23897.366395	0.314307	0.314307	0.372911	0.0
2	29 days	3.033802e+08	17417.812290	17417.812290	0.419737	0.419737	0.346928	0.0
3	56 days	1.163640e+07	3411.216897	3411.216897	0.054125	0.054125	0.055630	1.0
4	58 days	3.141442e+07	5604.857215	5604.857215	0.096053	0.096053	0.091651	1.0
5	59 days	1.022419e+08	10111.474758	10111.474758	0.201580	0.201580	0.183123	0.0
6	86 days	1.385824e+07	3722.666191	3722.666191	0.101704	0.101704	0.096782	1.0
7	88 days	7.134074e+04	267.096866	267.096866	0.003574	0.003574	0.003581	1.0
8	90 days	2.339581e+07	4836.921613	4836.921613	0.073518	0.073518	0.076324	1.0
9	117 days	6.459926e+08	25416.385069	25416.385069	0.334285	0.334285	0.401372	0.0
10	119 days	2.585592e+08	16079.775825	16079.775825	0.387492	0.387492	0.324602	0.0
11	121 days	1.040953e+06	1020.271112	1020.271112	0.017329	0.017329	0.017481	1.0
12	148 days	1.687488e+07	4107.904966	4107.904966	0.070399	0.070399	0.068005	1.0
13	149 days	1.599832e+08	12648.447256	12148.244899	0.252354	0.252354	0.221816	0.0
14	178 days	3.158565e+06	1777.235271	1777.235271	0.023782	0.023782	0.024069	1.0
15	180 days	3.573728e+08	18904.305977	16207.680222	0.430109	0.430109	0.327599	0.5

Conclusion and next steps

Results and Recommendation :

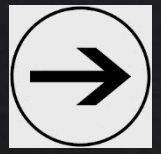
- We get the best results through Facebook Prophet model.
- By adding more features like holidays, promotions, weekends and so on, we can get better results through this model.
- After that, we can use accurate sales forecasts to optimize production and inventory management, ensuring that the right products are available when customers need them.
- Develop targeted marketing campaigns and pricing strategies based on the analysis of historical sales data to drive revenue growth.
- Monitor market trends and adjust strategies to stay ahead of competitors and adapt to changing consumer demand.

Forecasting ready for all scenarios:

By implementing these recommendations based on the forecasting results, we can ultimately develop targeted marketing campaigns and pricing strategies leading to increased in revenue.

benefits:

Monetary, time, resources, Inventory



Next steps:

- Improve the model by adding more features
- Executing the recommendations
- Do the same model for other products
- Gather much more dataset to get better results in deep learning models.

Challenge:

As the data is based on sales history, we need to reassess sales forecast again to consider changes.

Thank You

By: Hadi Nazari