Application of machine learning techniques for supply chain demand forecasting

M.Sc. - Project 1(IE685)

by

Priyansu (23N0464)

under the guidance of

Prof. Priyank Sinha

Interdisciplinary Programme in

Industrial Engineering and Operations Research

Indian Institute of Technology, Bombay

Mumbai 400 076

Declaration

I declare that this written submission represents my ideas in my own words and where other ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented, fabricated, or falsified idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been taken when needed.

In keeping with the general practice of reporting scientific observations, due acknowledgment has been made wherever the work described here has been based upon the work of other investigators. Any oversight due to an error of judgement is regretted.

Priyansu
(23N0464)

Abstract

Accurate demand forecasting is essential for efficient supply chain management in the dynamic transactional industry, where precise predictions help optimize production, inventory, and transportation. Collaboration between supply chain participants is ideal, but various challenges often limit the exchange of full information. This creates a need for reliable demand forecasting even when participants lack complete data on others' demands.

In this study, we explore the use of advanced machine learning techniques, including recurrent neural networks, LSTM, Random Forest, and XGBoost to forecast distorted demand (bullwhip effect) at the end of a supply chain. These methods are compared with traditional ones, such as moving average, exponential smoothing, ARIMA, and linear regression.

Our experiments are carried out on two datasets: one generated from a simulated supply chain and the other based on transactional sales data. The results indicate that among traditional models, the exponential weighted moving average performs best, while in machine learning methods, the random forest and XGboost achieves superior accuracy. We used hyperparameter tuning and cross-validation to optimize and validate the models, ensuring robust and reliable predictions.

Contents

List of Figures

Chapter 1

Supply chain demand forecasting techniques

In this chapter, we are looking at the concept and math behind the different techniques,

- Naive Forest
- Moving Average
- Exponentially Weighted Moving Average(EWMA)
- Autoregressive Integrated Moving Average(ARIMA)
- Multiple Linear Regression
- Elastic Net
- Random Forest
- XGboost
- Long Short-Term Memory (LSTM)

1.1 What is Naive Forest?

Naive methods belong to the group of simple forecasting methods and are used to build short-term forecasts. They are most often used at a constant level of the phenomenon and small random fluctuations, but can be extended to take into account seasonality, in which case the forecast at time t + h is the last value of the observation in the corresponding period:

$$y_{t+h} = y_{t+h-m(k+1)}$$

where:

m—seasonal period (number of cycle phases)
k—the integer part of $\frac{h-1}{m}$

Figure 1.1: Formulatio of the naive forecast

1.2 What is Moving Average?

A moving average process is a technique in time series analysis that is used to predict long-term trends from a data set while 'smoothing out' short-term fluctuations. It solves a problem we face when dealing with time-series data: It differentiates spikes from an actual trend. Moving averages are essentially nonparametric methods for trend (or signal) estimation and are not for model building. Moving averages are useful for several reasons. Smoothing data, identifying trends, and forecasting.

$$SMA = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

where:

A = Average in period n

n = Number of time periods

Figure 1.2: Formula for the Moving Average

1.3 What is the EWMA?

Exponentially Weighted Moving Averages (EWMA) is a statistical method used for data analysis, particularly in time series forecasting and financial modeling. It offers a powerful and flexible way to understand and predict trends, providing a balance between historical and recent data, allowing analysts to adapt to changing patterns over time. This paper will delve into the concept of EWMA, its mathematical formulation, practical applications, and the advantages it offers in various fields.

$$\mathrm{EWMA}_t = \alpha * r_t + (1 - \alpha) * \mathrm{EWMA}_{t-1}$$

Where:

- **Alpha** = The weight decided by the user
- **r** = Value of the series in the current period

Figure 1.3: EWMA Formula

1.4 What is the ARIMA?

The ARIMA (Auto Regressive Moving Average) model is a very common time series forecasting model. It is a more sophisticated extension of the simpler ARMA (Auto Regressive Moving Average) model, which in itself is just a merger of two even simpler components:

$$y_t = c + \sum_{i=1}^{p} \phi_i \, y_{t-i} + \sum_{j=1}^{q} \theta_j \, \epsilon_{t-j} + \epsilon_t$$

Where:

p and q are the orders of the AR and MA models, respectively.

p and q are the orders of the AR and MA models, respectively.

Figure 1.4: ARIMA

1.AR (Auto Regressive): models attempt to predict future values based on past values. AR models require the time series to be stationary.

$$y_t = c + \phi_1 y_{t-1} + \epsilon_t$$

Where:

y_t Is the value at time step t, c is a constant, ϕ_1 is a coefficient, and ϵ_t is a white noise error term with $\epsilon_t \sim N\ (0, \sigma^2)$.

Figure 1.5: AR Formula

2.MA (Moving Average): models attempt to predict future values based on past forecasting errors. MA models assume that an autoregressive model can approximate the given series.

An *MA(q)* model can be expressed as:

$$y_t = c + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \ldots + \theta_q \epsilon_{t-q} + \epsilon_t$$

Where:

y_t Is the value at time step t, c is a constant, θ_1 is a coefficient, and ϵ_{t-1} is a previous white noise error term.

This can be finalized to:

$$y_t = c + \sum_{j=1}^{q} \theta_j\, \epsilon_{t-j}$$

Figure 1.6: MA Formula

1.5 What is Multiple Linear Regression?

Multiple linear Regression is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical (dummy coded as appropriate).

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon$$

where, for $i = n$ observations:

y_i = dependent variable

x_i = expanatory variables

β_0 = y-intercept (constant term)

β_p = slope coefficients for each explanatory variable

ϵ = the model's error term (also known as the residuals)

Figure 1.7: MLR

1.6 ElasticNet Regression Model

Elastic Net Regression is a linear regression technique that combines Lasso (Least Absolute Shrinkage and Selection Operator) and Ridge Regression. Like Lasso and Ridge Regression, Elastic Net aims to minimize the sum of squared errors between observed and predicted values while also incorporating a regularization term.

The ElasticNet mathematical model is defined as:

$$L_{\text{enet}}(\hat{\beta}) = \frac{\sum_{i=1}^{n}(y_i - x_i^\top \hat{\beta})^2}{2n} + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^{m} \hat{\beta}_j^2 + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right),$$

where:

- y_i represents the observed value,
- $x_i^\top$ is the transpose of the predictor vector,

- $\hat{\beta}$ are the estimated coefficients,
- n is the number of observations,
- λ is the regularization parameter,
- α controls the balance between L1 (lasso) and L2 (ridge) penalties.

1.6.1 Lasso Regression Model

The main purpose in Lasso Regression is to find the coefficients that minimize the error sum of squares by applying a penalty to these coefficients

$$L_{\text{lasso}}(\hat{\beta}) = \sum_{i=1}^{n} \left(y_i - \sum_{j} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

where:

- λ denotes the amount of shrinkage.
- $\lambda = 0$ implies all features are considered.
- $\lambda = \infty$ implies no feature is considered.
- The bias increases with an increase in λ.
- The variance increases with a decrease in λ.

1.6.2 Ridge Regression Model

Ridge Regression Model is a version of the classical regression equation with a correction function.

$$L_{\text{ridge}}(\hat{\beta}) = \sum_{i=1}^{n} \left(y_i - \sum_{j} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2,$$

1.7 What is Random Forest?

The Random Forest Algorithm is composed of different decision trees, each with the same nodes, but using different data that leads to different leaves. It merges the decisions of multiple decision trees in order to find an answer, which represents the average of all these decision trees.

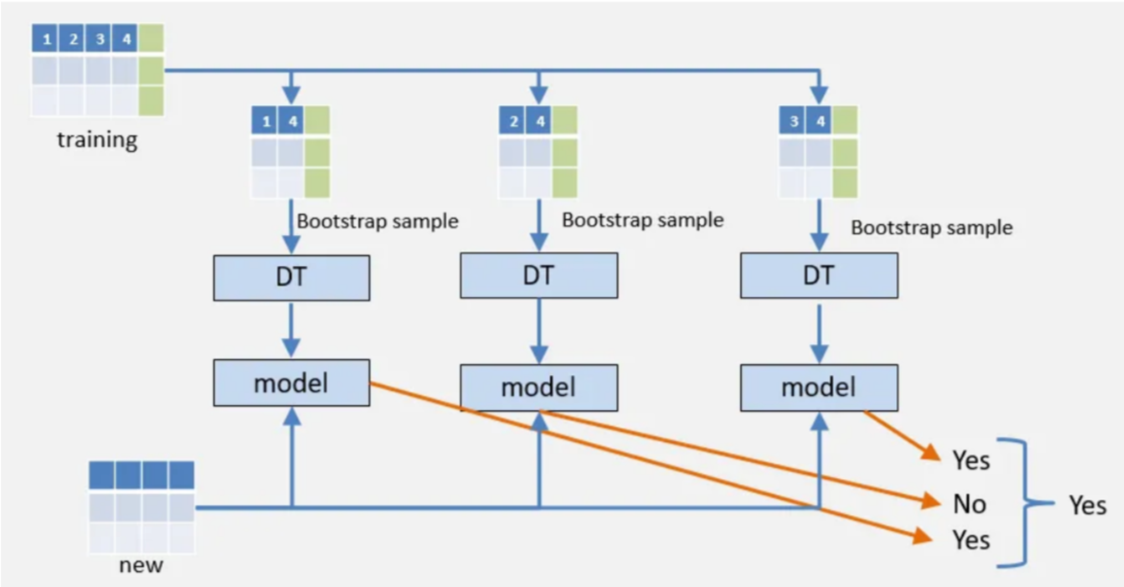

Figure 1.8: Enter Caption

Stopping criteria for splitting into leaf nodes:

- **Maximum depth of the tree:** The tree stops splitting once a predefined depth is reached.
- **Minimum samples in a leaf:** If the number of samples in a node falls below a certain threshold, it is converted into a leaf.
- **Minimum improvement in impurity:** If the reduction in impurity from a potential split is below a threshold, the node becomes a leaf.
- **Maximum number of leaf nodes:** The tree stops growing once it reaches a predefined number of leaf nodes

1.8 What is XGBoost?

XGBoost (eXtreme Gradient Boosting) is an advanced implementation of gradient boosting algorithm. It's a powerful machine learning algorithm especially popular for structured or tabular data. XGBoost has gained fame for its performance in a wide range of machine learning competitions and tasks.

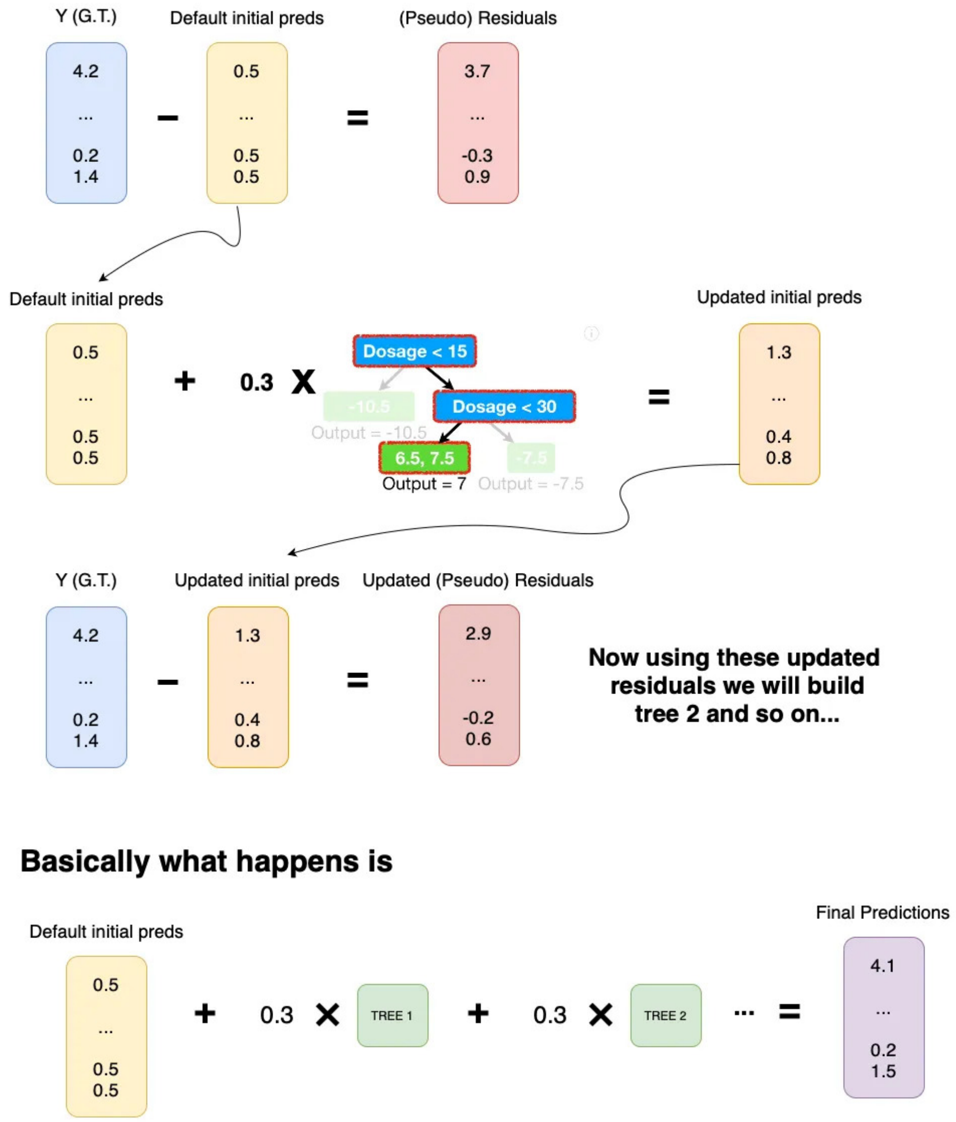

Figure 1.9: XGBoost

1.9 What is LSTM?

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem and capture long-term dependencies in sequential data. LSTMs have been widely used in various natural language processing tasks, time series analysis, speech recognition, and more.

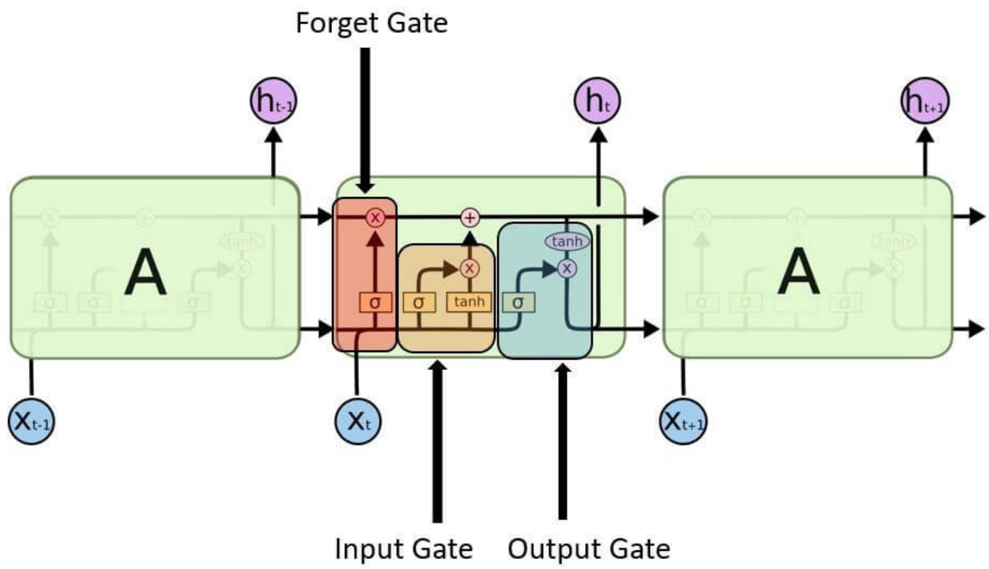

Figure 1.10: LSTM

Evaluation Metrics

1. MAE (Mean Absolute Error):

Formula:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Description: It represents the average of the absolute differences between the actual (y_i) and forecasted ($\hat{y}_i$) values.

2. RMSE (Root Mean Squared Error):

Formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Description: RMSE gives more weight to larger errors and is useful when large errors are particularly undesirable.

3. MAPE (Mean Absolute Percentage Error):

Formula:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

Description: MAPE expresses the error as a percentage of the actual values, making it easier to interpret the forecast accuracy.

Chapter 2

Data Set

In order to examine the effectiveness of various advanced machine learning techniques in forecasting supply chain demand, two data sets were prepared. The first one represents results of the generated sales data, and the second one contains actual transactional sales data.

2.1 Preprocessing

The original dataset comprises 2823 rows and 25 columns. For the purpose of analysis, only the relevant columns were retained. The data was then aggregated based on location. Due to significant variability in the dataset, scaling was performed, resulting in the current structure of the data.

CITY	COUNTRY	DEALSIZE	QUANTITYORDERED	PRICEEACH	SALES	MSRP	QTR_ID	MONTH_ID	YEAR_ID	PRODUCTLINE	ORDERDATE
San Francisco	USA	Small	0.331637	0.373975	0.170289	0.190389	1	3	2003	Vintage Cars	2003-03-31
Frankfurt	Germany	Medium	0.081660	1.000000	0.097567	0.780488	4	11	2004	Trucks and Buses	2004-11-30
Lule	Sweden	Medium	0.156627	1.000000	0.183632	0.845528	4	11	2004	Trucks and Buses	2004-11-30
Bruxelles	Belgium	Small	0.325714	0.671165	0.290320	0.434874	3	7	2004	Ships	2004-07-31
Singapore	Singapore	Medium	0.098388	0.965893	0.124912	0.733577	3	9	2004	Vintage Cars	2004-09-30

Figure 2.1: data set

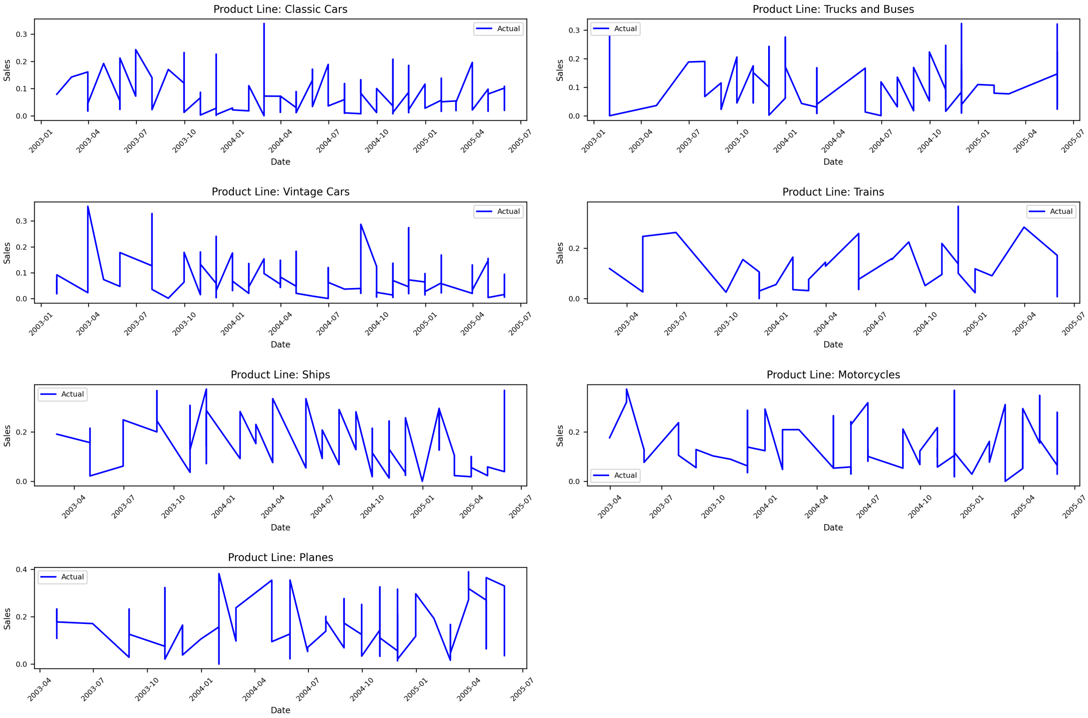

Figure 2.2: As observed in the image, the data for each product line is stationary

2.2 Simulated Data

To simulate the data, I fitted a distribution to the original data, extracted the parameters, and generated sample data based on the fitted distribution. For categorical data, I used sampling with replacement from an existing column in the DataFrame.

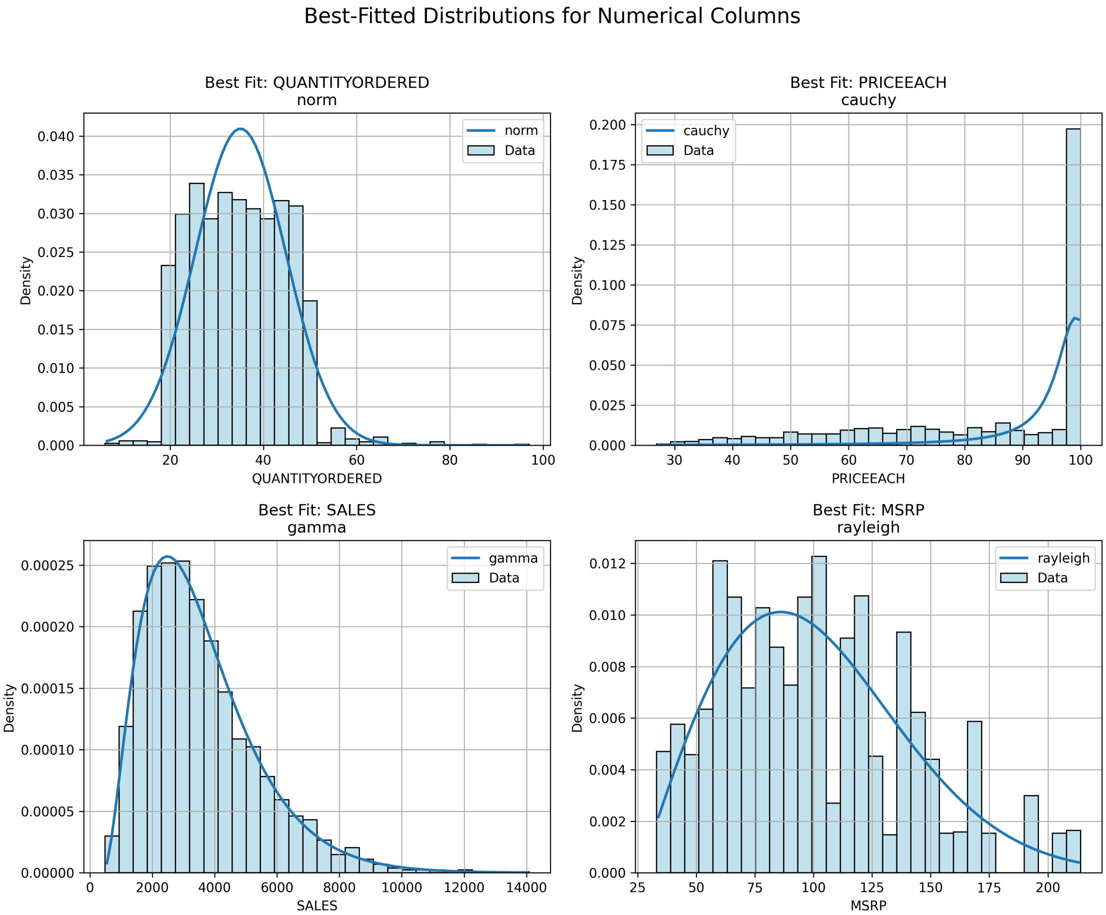

Figure 2.3: The distribution of the numerical columns in the actual data

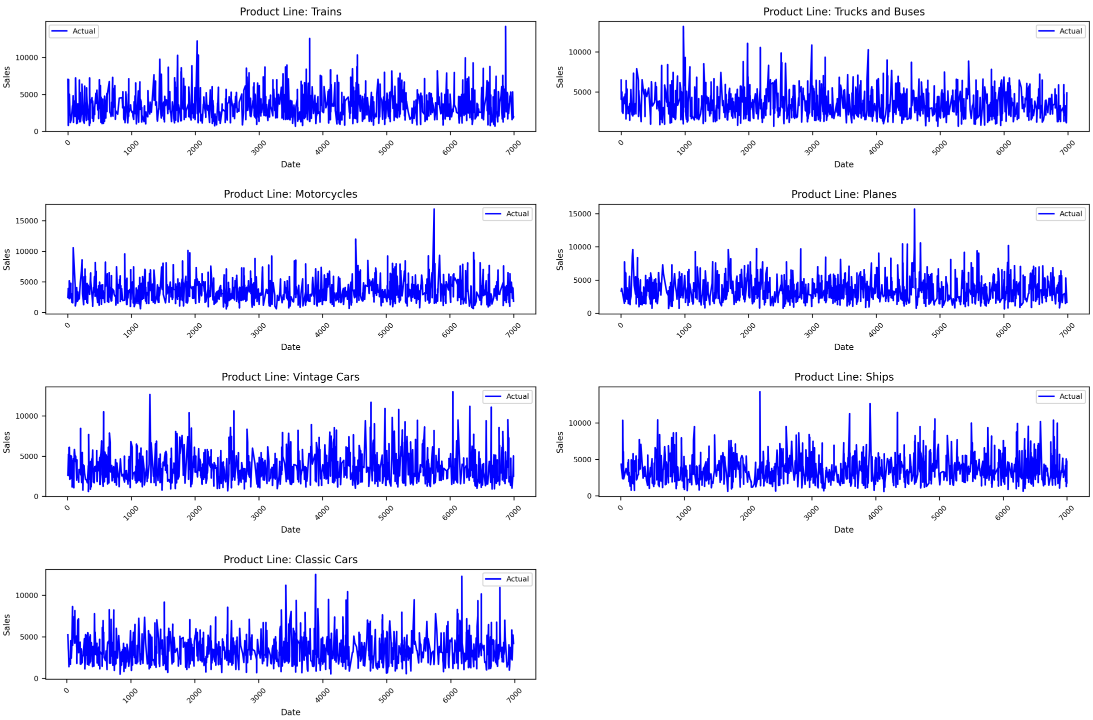

Figure 2.4: As observed in the image, the simulated data for each product line is also stationary

Chapter 3

Experimental results

This section describes the results of experiments using various forecasting techniques on the two data sets described in the previous sections

3.0.1 Comparison of the performance of forecasting techniques on the Original data set

Model	**MAE**	**RMSE**	**MAPE**
Naïve Forest	0.061	0.086	67,251.903
Moving Average	0.038	0.050	26,200,000.000
EWMA	0.006	0.008	6,167.896
ARIMA	0.034	0.043	94.416
Multiple LR	0.016	0.020	64.658
Elastic Net	0.035	0.052	93.030
Random Forest	0.012	0.026	19.913
Xgboost	0.014	0.027	19.980
LSTM	0.087	0.116	80.921

Table 3.1: Original Data: Model Performance Metrics for Classic Cars

Model	MAE	RMSE	MAPE
Naïve Forest	0.106	0.128	407,061.512
Moving Average	0.056	0.068	381.000
EWMA	0.010	0.012	40,127.855
ARIMA	0.068	0.098	83.126
Multiple LR	0.025	0.036	789.616
Elastic Net	0.028	0.040	321.044
Random Forest	0.020	0.030	160.562
Xgboost	0.016	0.025	31.617
LSTM	0.307	0.351	146.671

Table 3.2: Original Data: Model Performance Metrics for Trucks and Buses

Model	MAE	RMSE	MAPE
Naïve Forest	0.066	0.089	7,091.624
Moving Average	0.042	0.055	7,330,000.000
EWMA	0.006	0.009	893.767
ARIMA	0.048	0.053	263.289
Multiple LR	0.020	0.026	40.816
Elastic Net	0.035	0.048	95.823
Random Forest	0.012	0.020	17.509
Xgboost	0.014	0.021	17.791
LSTM	0.131	0.142	343.215

Table 3.3: Original Data: Model Performance Metrics for Vintage Cars

Model	MAE	RMSE	MAPE
Naïve Forest	0.092	0.116	270,997.629
Moving Average	0.058	0.071	229,000,000.000
EWMA	0.009	0.011	27,389.296
ARIMA	0.064	0.082	253.733
Multiple LR	0.025	0.030	83.268
Elastic Net	0.038	0.044	106.305
Random Forest	0.027	0.033	68.137
Xgboost	0.032	0.041	72.390
LSTM	0.223	0.286	307.059

Table 3.4: Original Data :Model Performance Metrics for Trains

Model	MAE	RMSE	MAPE
Naïve Forest	0.129	0.155	395,346.336
Moving Average	0.068	0.083	208,000,000.000
EWMA	0.012	0.015	37,089.776
ARIMA	0.096	0.126	669,723.182
Multiple LR	0.015	0.019	27.977
Elastic Net	0.036	0.044	39.174
Random Forest	0.021	0.027	25.983
Xgboost	0.020	0.027	18.030
LSTM	0.247	0.335	142.788

Table 3.5: Original Data: Model Performance Metrics for Ships

Model	**MAE**	**RMSE**	**MAPE**
Naïve Forest	0.110	0.138	291,310.864
Moving Average	0.064	0.079	238,000,000.000
EWMA	0.010	0.013	29,667.299
ARIMA	0.103	0.114	917,593.590
Multiple LR	0.029	0.036	33.502
Elastic Net	0.036	0.050	38.164
Random Forest	0.032	0.040	32.590
Xgboost	0.028	0.042	23.427
LSTM	0.268	0.306	92.470

Table 3.6: Original Data: Model Performance Metrics for Motorcycles

Model	**MAE**	**RMSE**	**MAPE**
Naïve Forest	0.121	0.149	38,807.442
Moving Average	0.070	0.087	82,700,000.000
EWMA	0.011	0.014	5,410.018
ARIMA	0.126	0.140	120.042
Multiple LR	0.023	0.030	29.579
Elastic Net	0.049	0.060	51.759
Random Forest	0.034	0.043	27.937
Xgboost	0.040	0.049	31.709
LSTM	0.307	0.338	134.962

Table 3.7: Original Data: Model Performance Metrics for Planes

3.0.2 Comparison of the performance of forecasting techniques on the simulation data set

Model	MAE	RMSE	MAPE
Naïve Forest	0.222	0.286	72,049.100
Moving Average	0.131	0.164	429,161,700.000
EWMA	0.021	0.027	7,050.770
ARIMA	0.165	0.207	197,073.290
Multiple LR	0.151	0.183	78.970
Elastic Net	0.149	0.182	79.850
Random Forest	0.161	0.197	84.237
Xgboost	0.170	0.206	87.308
LSTM	0.168	0.206	228,254,200.000

Table 3.8: Simulation dataset:Model Performance Metrics for the simulated data of Classic Cars

Model	MAE	RMSE	MAPE
Naïve Forest	0.212	0.266	28,555.630
Moving Average	0.123	0.155	15,031,810.000
EWMA	0.020	0.025	2,767.030
ARIMA	0.164	0.205	682.200
Multiple LR	0.153	0.187	409.390
Elastic Net	0.154	0.188	476.094
Random Forest	0.155	0.193	382.889
Xgboost	0.165	0.204	396.479
LSTM	0.163	0.211	639.300

Table 3.9: Simulation dataset: Model Performance Metrics for Trucks and Buses

Model	MAE	RMSE	MAPE
Naïve Forest	0.217	0.278	44,056.720
Moving Average	0.126	0.161	25,296,130.000
EWMA	0.021	0.027	4,303.360
ARIMA	0.150	0.192	82.910
Multiple LR	0.161	0.203	124,360.166
Elastic Net	0.157	0.198	132,983.919
Random Forest	0.169	0.211	123,919.843
Xgboost	0.178	0.225	133,453.133
LSTM	0.143	0.195	73.080

Table 3.10: Simulation dataset: Model Performance Metrics for Vintage Cars

Model	MAE	RMSE	MAPE
Naïve Forest	0.232	0.297	35,158.370
Moving Average	0.135	0.171	40,967,100.000
EWMA	0.022	0.028	3,985.990
ARIMA	0.172	0.211	122.240
Multiple LR	0.169	0.210	122.304
Elastic Net	0.168	0.208	123.270
Random Forest	0.180	0.224	125.929
Xgboost	0.182	0.227	119.454
LSTM	0.171	0.211	118.480

Table 3.11: Simulation dataset: Model Performance Metrics for Trains

Model	**MAE**	**RMSE**	**MAPE**
Naïve Forest	0.221	0.280	12,673.620
Moving Average	0.127	0.161	13,639,840.000
EWMA	0.021	0.027	1,405.560
ARIMA	0.158	0.198	68.160
Multiple LR	0.168	0.214	134.274
Elastic Net	0.170	0.215	142.657
Random Forest	0.178	0.223	136.468
Xgboost	0.181	0.227	148.605
LSTM	0.158	0.199	67.030

Table 3.12: Simulation dataset: Model Performance Metrics for Ships

Model	**MAE**	**RMSE**	**MAPE**
Naïve Forest	0.231	0.292	44,958.390
Moving Average	0.136	0.169	25,364,230.000
EWMA	0.022	0.028	4,340.750
ARIMA	0.147	0.189	164.790
Multiple LR	0.163	0.207	133.289
Elastic Net	0.162	0.206	144.075
Random Forest	0.172	0.215	142.828
Xgboost	0.180	0.228	132.196
LSTM	0.149	0.192	175.110

Table 3.13: Simulation dataset: Model Performance Metrics for Motorcycles

Model	**MAE**	**RMSE**	**MAPE**
Naïve Forest	0.227	0.286	36,173.040
Moving Average	0.123	0.155	15,031,810.000
EWMA	0.022	0.027	3,549.170
ARIMA	0.159	0.197	83.850
Multiple LR	0.166	0.200	95.346
Elastic Net	0.164	0.196	93.459
Random Forest	0.166	0.204	91.473
Xgboost	0.171	0.209	93.620
LSTM	0.161	0.199	88.400

Table 3.14: Simulation dataset: Model Performance Metrics for Planes

3.1 Conclusion

The aim of this project was to explore how effective advanced machine learning techniques are for forecasting distorted demand signals in an extended supply chain.

For the simulated supply chain dataset, the advanced machine learning models showed only a slight improvement over traditional forecasting methods. Techniques like Naive Forecast, Moving Average, Exponentially Weighted Moving Average (EWMA), and ARIMA performed relatively well in this controlled setting. The improvements offered by machine learning models were not significantly better in this case.

However, for the real transactional sales dataset, advanced machine learning techniques such as Multiple Linear Regression (MLR), Elastic Net, Random Forest, XGBoost, and LSTM showed larger improvements in forecasting accuracy. Among these, Random Forest and XGBoost consistently provided the best results, delivering more accurate predictions for each product across all evaluation metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

On the other hand, Naive Forecast and Moving Average performed the worst in both datasets, producing the highest error levels. These simpler methods struggled to effectively process the demand signals, making them less reliable for supply chain forecasting.

In conclusion, this project highlights that while traditional techniques can work reasonably well in simpler or simulated settings, machine learning methods are far superior when dealing with real-world complexities. By leveraging advanced algorithms, supply chain managers can achieve more accurate demand forecasts, reduce errors, and improve decision-making. This study emphasizes the importance of choosing the right forecasting method based on the nature of the dataset and the desired level of accuracy.

Bibliography

[1] Various Authors. Sustainability practices in the industry. *Sustainability*, 15(9):7399, 2023. Accessed: 2024-11-29.

[2] Various Authors. An analysis of advanced forecasting techniques, 2024. Accessed: 2024-11-29.

[3] Various Authors. Economic forecasting using advanced arima models. *Economic Modelling*, 93:307–324, 2024. Accessed: 2024-11-29.

[4] Paperspace Blog. Time series forecasting: Regression and lstm, 2024. Accessed: 2024-11-29.

[5] CoinMonks. Naive persistence model: A baseline forecasting technique in quantitative finance, 2024. Accessed: 2024-11-29.

[6] SFU CSPMP. Xgboost: A deep dive into boosting, 2024. Accessed: 2024-11-28.

[7] Investopedia. Autoregressive integrated moving average (arima), 2024. Accessed: 2024-11-29.

[8] Kerem Kargın. Elasticnet regression: Fundamentals and modeling in python, 2024. Accessed: 2024-11-28.

[9] AI Monks. Understanding exponentially weighted moving averages (ewma), 2024. Accessed: 2024-11-28.

[10] Harshdeep Singh. Understanding random forests, 2024. Accessed: 2024-11-28.

[11] Prathamesh Sonawane. Xgboost: How does this work?, 2024. Accessed: 2024-11-28.

[12] Analytics Vidhya. Introduction to long short-term memory (lstm), 2024. Accessed: 2024-11-29.

[13] Analytics Vidhya. A thorough introduction to arima models, 2024. Accessed: 2024-11-28.

[14] Analytics Vidhya. Understanding the linear regression, 2024. Accessed: 2024-11-28.

[9] [13] [14] [8] [10] [6] [11] [?] [1] [2] [3] [5] [12] [4] [7]