Store Item Demand Forecasting Using Different Models

Hira Komal **40240336**

Gustavo Orna Villalta **40222999**

Romyna Cano Quevedo **40199777**

Mohamed Bouzid **40221466**

Saad Muhammad Syed **4024153**

Concordia Institute for Information Systems Engineering, Concordia University

***Abstract*— To make well-informed decisions during the planning stage of the supply chain management process, demand forecasting is essential. With the goal to maximize inventory management and production planning, we investigate and assess the effectiveness of advanced forecasting models in this research article, taking into account the dynamic nature of customer demand. Our research starts with a focus on data pretreatment, graphical analysis, and transformation to address non-stationarity and reveal trends and seasonality that includes sales data for 50 items across 10 retailers. The use of well-known forecasting models, such as XGBoost, ARIMA, Linear Regression, Random Forest, and LSTM, is then covered in detail. In the realm of supply chain demand prediction, the choice of a model is contingent on data characteristics. We discuss the suitability of each model, emphasizing ARIMA's versatility and simplicity, making it a preferred choice. Insights into the benefits of precise demand forecasting in supply chain management are provided in the paper's conclusion.**

Keywords: Risk mitigation, bullwhip effect, gradient boosting

I. INTRODUCTION

A. Background

Accurately anticipating and reacting to changes in customer demand is critical in the complex world of supply chain management. Demand forecasting becomes a crucial tool, providing a strategic viewpoint for well-informed decision-making in the planning stage. Understanding how customer demand is dynamic and changes over time, along with how to adapt and change with it, is crucial to managing an effective and flexible supply chain [1].

With this project, the complex problem of demand forecasting in the supply chain sector will be addressed in an engaging way. The fundamental assumption is that decision-making requires a sophisticated approach due to the dynamic and ever-changing nature of client demand. The crucial element, which is sometimes overlooked, is carefully choosing the appropriate forecasting model.

B. Project Objective

This project's main goal is to investigate and utilize the potential of advanced forecasting models. These models are complex and adaptive, making them well-suited to handle the intricacies of present-day supply chain dynamics. By carefully assessing their effectiveness, this research seeks to provide practical insights that transcend the conventional forecasting domain.

C. Project Scope and Approach

The project's scope includes a thorough examination of an extensive dataset covering a period of five years. Sales information for a wide range of 50 different products spread over 10 different locations is contained in this dataset. The period ranging from 2013, to 2017, offers a strong basis for dissecting the complex web of customer demand patterns.

D. Navigating the Project Landscape

This project is progressing in various stages. At the outset, the analysis acknowledges the non-stationary nature of the data and emphasizes the need for advanced forecasting methods beyond simple average calculations. Further research involves graphically breaking down the data set, discovering trends, seasonality and additional structures that form the basis of subsequent modeling. The core of the project lies in the careful evaluation of advanced forecasting models and the selection of those suitable to capture and respond to the characteristics of the identified dataset.

The ultimate goal is to create actionable insights that enable companies to optimize inventory management and production planning that promotes a proactive and responsive supply chain ecosystem. Essentially, this project represents a holistic approach to supply chain management, demand forecasting, data matching analyses, pattern research and strategic insights. As we delve into the complexity of customer demand, the results of this project promise to take supply chain efficiency to new heights, opening a paradigm where anticipation and adaptability come together to achieve operational excellence.

II. DATASET

The dataset spans five years, encompassing sales data for 50 different items across 10 distinct stores. The timeframe ranges from January 1, 2013, to December 31, 2017, providing a robust foundation for in-depth analysis. Figure 1 provides an overview of the dataset where different colors in the plot represent different stores. The dataset consists of a total of

913000 observations of daily sales data from different store and item combinations. There are 4 columns (features) in the dataset: data, store, item , sales which is the target variable. The data column follows the format YYYY-MM-DD. The store and item features are numerical identifiers ranging from 1 to 10 and 1 to 50 respectively. The sales column is the number of units sold for that item-store combination on that date. Before delving into analysis, outlier and missing data check was performed and it was shown that the dataset does not have any of them. We obtained the dataset from Kaggle [2].

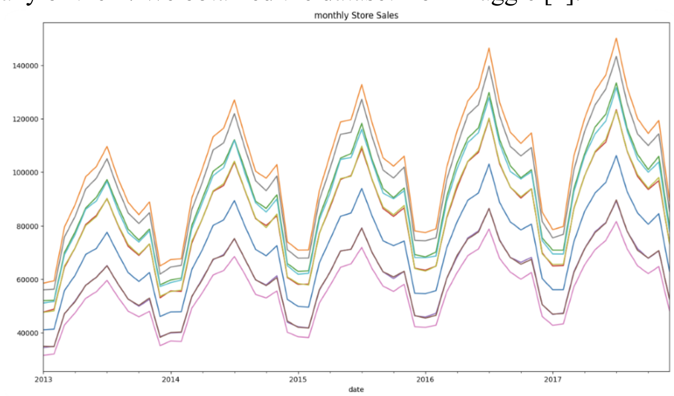

Figure 1. Yearly sales from 2013 to 2017.

.

III. ANALYSIS

In figure 2, we analyze the dataset by checking the trend and seasonality of the dataset. We conclude the following from the graphs:

A. Data Preprocessing

1. *Sales Dataset (Stationarity Assessment) – 1st Graph*
   The examination of the sales dataset reveals a non-stationary nature. Acknowledging this characteristic is crucial, as it implies that relying on mean values alone for future predictions may lead to inaccuracies. This insight emphasizes the need for advanced forecasting methods to address non-stationarity effectively.

B. Graphical Analysis

1. *Trend Analysis – 2nd Graph*
   The second graph illustrates a discernible increasing trend over time in the sales data. This systematic upward movement highlights the importance of considering and incorporating this trend into our forecasting models. Advanced models capable of capturing and accommodating trends will likely result in more accurate predictions.

2. *Seasonality Identification – 3rd Graph*
   The third graph unveils a repeating pattern, particularly with an upward movement in sales observed during July. This recurring pattern proves that there is seasonality in the data. Recognizing and incorporating seasonality into our forecasting models is crucial for enhancing accuracy, allowing adjustments based on recurring patterns in client demand.

3. *Residuals Analysis – 4th Graph*
   In the fourth graph, the observation that residuals are decomposing randomly around 0 indicates an additive series structure. This characteristic provides valuable insights when selecting an appropriate modeling approach. An additive series suggests that the components (trend, seasonality, and residuals) can be combined linearly, facilitating the development of a more interpretable forecasting model.

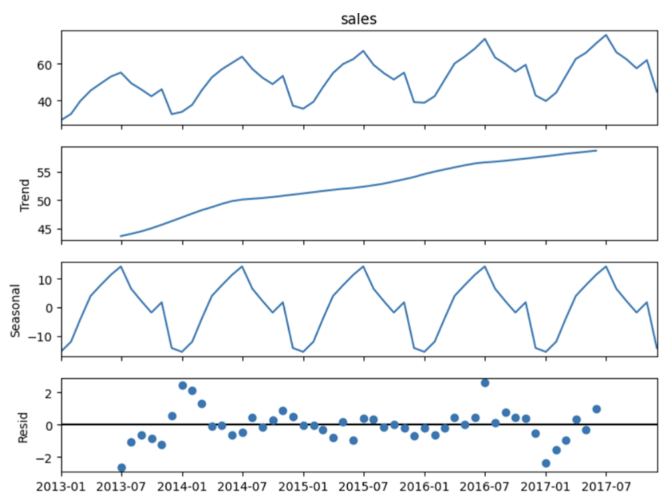

Figure 2. Top to bottom. Sales dataset, trend analysis, seasonal identification graph and residual analysis.

C. Transformation

Monthly Sales Before Differencing Transformation

The initial state of the monthly sales data exhibits non-stationary behavior, likely characterized by varying means over time. This non-stationarity introduces complexity in the forecasting process, prompting the need for a transformation to enhance the suitability of the data for advanced modeling.

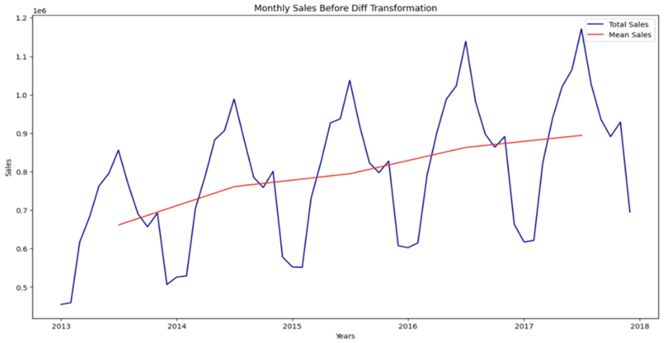

Fig 3 Shows monthly sales before differencing transformation

2. *Differencing Transformation*

Differencing is applied to the monthly sales data by computing the difference between two consecutive terms in the time series dataset..

3. *Monthly Sales After Differencing Transformation (Stationary)*

The resulting series after the differencing transformation is expected to exhibit stationarity. Stationarity is a desirable property in time series analysis, signifying that the statistical properties, mean and variance, are constant over time. Achieving stationarity enhances the suitability of the data for advanced forecasting models, contributing to improved accuracy in predicting future sales.

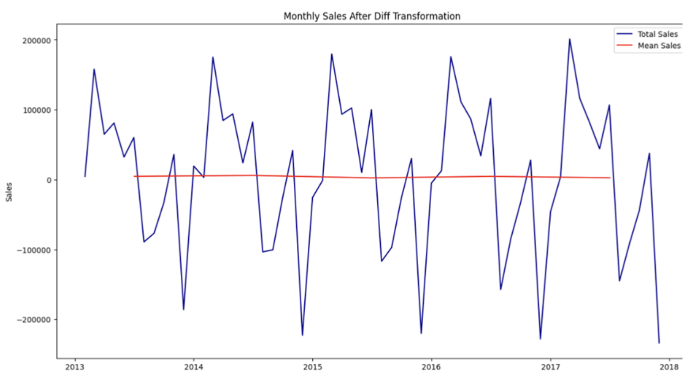

Figure 3. Monthly sales after differencing transformation.

D. Next Steps

This transformation lays the groundwork for the next steps in the analysis, where the stationary series will be utilized in the application of advanced forecasting models like ARIMA.. These models, coupled with the insights gained from the previous analyses, are poised to provide accurate predictions for optimizing inventory management and production planning.

IV. AUTOCORRELATION AND PARTIAL CORRELATION

An important component of our time series forecasting method is taking historical sales data into account. We use partial correlation and autocorrelation techniques to capture the temporal relationships in the data. These strategies are especially crucial in light of our feature representation, in which every feature corresponds with sales from the preceding month. The autocorrelation coefficient assesses the degree of association between a month's sales and its historical levels. The correlation and partial correlation plots for our dataset are shown in Figure 4. By looking into the graphs, we decide a **twelve month lookback period**, and we create a data frame with thirteen columns, one for each of the previous twelve months and one for our dependent variable.

It is necessary to comprehend the influence of other aspects in addition to the direct impact of previous sales in order to effectively express our features. By accounting for the impact of other months, partial correlation allows us to figure out the precise relationship between sales from a given month and current sales.

i. Partial Correlation Coefficients:

Measuring the direction and degree of the direct association between sales in the current month and sales in each lag month.

ii. Managing Confounding Factors:

Resolving the possible impact of further delayed months, guaranteeing a more lucid comprehension of the causal relationship between previous and current sales.

iii. Increasing Model Robustness:

By taking into consideration the interdependencies of lag features, partial correlation insights help to create forecasting models that are more robust.

Autocorrelation and partial correlation investigations provide valuable information that impact our forecasting models.The identified lags and their correlations guide the inclusion of relevant features, ensuring our models capture the nuances of the sales patterns observed over the past year. Our estimates are more accurate because of the incorporation of temporal insights, which also provide us a more sophisticated picture of how past sales have influenced present performance.

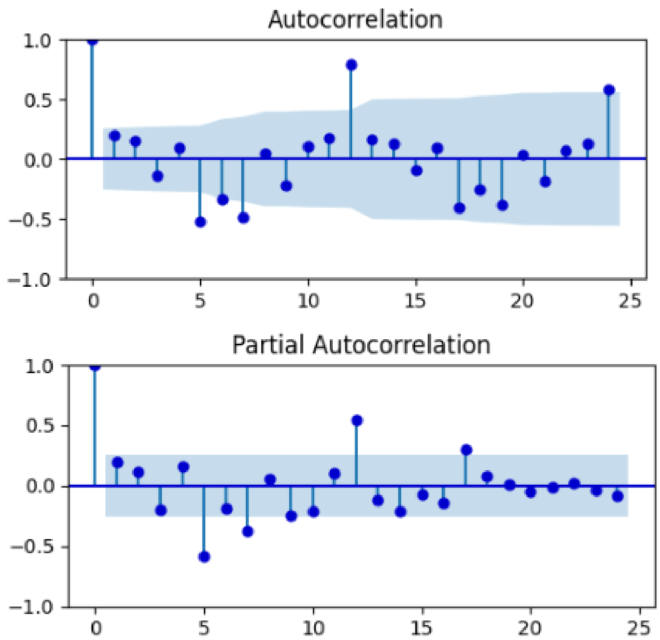

Figure 4. Autocorrelation and partial correlation plots .

V. MODELS

A. ARIMA

The statistical model known as ARIMA[3], or AutoRegressive Integrated Moving Average, is used for time series forecasting. Sales, temperature, stock prices, and other data that are gathered over time are called time series. The process of predicting a time series' future values based on its historical values is known as forecasting. Three elements form the foundation of ARIMA models: moving average, integration, and autoregression. There is a parameter for each part that shows how much it influences the model. The letters p, d, and q, respectively, stand for these parameters. As a result, ARIMA(p,d,q) is the typical notation for an ARIMA model.

i. Autoregression (AR) is the mathematical process by which the present value of a time series is calculated by its past values, plus a random error. The number of prior values that are used is indicated by the parameter p. An AR(2) model, on the other hand, employs the two most recent values, whereas an AR(1) model simply uses the most recent one.

ii. Integration (I) is the process of making a time series stationary by calculating the difference between successive values. It is simpler to model a stationary time series since its mean and variance remain constant over time. The number of times the differencing is applied is indicated by the parameter d. An I(1) model, for instance, takes the first difference, whereas an I(2) model takes the second.

iii. A moving average (MA) indicates that there is some random error and that the present value of the time series depends on the model's historical errors. The number of previous errors is indicated by the parameter q. An MA(2) model uses the two most recent mistakes, whereas a MA(1) model just uses the most recent error.

There are several ways to fit an ARIMA model to time series data, including cross-validation, Bayesian inference, and maximum likelihood estimation. Using the historical values and time series errors, the fitted model can then be used to create single-step or multi-step forecasts.

B. LINEAR REGRESSION

A regression is a statistical method used in different areas like finance to try to determine what is the relationship between two variables. The regression analysis predicts a dependent variable based on one or multiple independent variables [4]. In this case, the linear regression was used as a prediction model with only one independent variable. The dependent variable is defined by the linear regression as it shown in the formula:

$$y = \beta 0 + \beta 1 x + \epsilon$$

Where y is the dependent variable that the model is trying to predict, $\beta 0$ is the intercept in the y-axis, $\beta 1$ is the slope of the independent variable, x is the independent variable and ϵ is the regression residual in the calculation. The model of linear regression uses this equation to estimate the values of the dependent variable using all the information provided of the independent variable. This model is frequently used in machine learning for its reliable prediction calculations.

C. RANDOM FOREST

A random forest regression is a technique that uses regression and classification by using multiple tree predictors and each tree is defined by a random vector that was sampled separately from the vector that is the main input of the decision tree[5]. A single decision tree has a high variance but when multiple trees are used in parallel, the variance lowers as each tree is trained on a particular sample, so the output of the technique is based on multiple trees rather than a single decision tree[6] This model regression works with two processes: bootstrap and aggregation. Bootstrap is when a random row and feature sampling is done from the dataset to have individual datasets for each model. Aggregation is the process to obtain the final output of the decision trees, which is to calculate the mean of all the outputs from all the trees. Combining these two processes along with the multiple decision trees, the prediction for this model is done. Random forest regression offers these advantages to the user: It is easy to use and implement as a model on a variety of datasets. It has better accuracy of prediction than a single decision tree model. It is perfectly capable of handling very large datasets without issues.

D. XGBOOST

XGBoost[7] is a popular and powerful package that uses the well-known gradient boosting algorithm for decision trees. XGBoost, standing for eXtreme Gradient Boosting is a portable, highly adaptable, and efficient system. Gradient boosting is a technique that builds a powerful learner by successively adding new decision trees that fix the mistakes of the old ones, thereby integrating several weak learners (usually shallow decision trees). This is not like traditional ensemble methods that combine weak learners sequentially. The term "gradient" describes how a loss function, which measures the effectiveness of the model, is fitted to the negative gradient for every new tree. By doing this, the model's accuracy and capacity for generalization will progressively increase. Compared to other gradient boosting implementations, XGBoost offers a number of benefits, including:

i. It performs better because it controls over-fitting with a more regularized model formalization.

ii. It supports tree-based and linear models, both of which can be optimized by the application of the gradient descent technique.

iii. It is scalable and quick since it supports distributed and parallel processing.

iv. It offers a number of features and functionalities, including early stopping, feature importance, cross-validation, pruning, and handling missing values.

E. LSTM

Before delving into LSTMs[8], it is important to understand recurrent neural networks (RNNs), the basis upon which LSTMs are based. RNNs are a type of Neural Networks (NN) used to handle sequential data by preserving hidden states that hold data from earlier time steps. Despite their strength in detecting sequential patterns, RNNs have issues with vanishing and exploding gradients. These problems occur when training gradients become abnormally small or big, making it difficult for long-term dependencies to be learned effectively.

LSTMs were developed to address the difficulties involved in learning and remembering knowledge over extended periods of time. LSTMs were created in 1997 by Hochreiter and Schmidhuber with the express purpose of addressing the shortcomings of conventional RNNs in terms of capturing long-range relationships. The cell state, input gate, forget gate, and output gate are the four parts of an LSTM cell. The long-term data is stored in the cell state, which is the central component of the LSTM unit. The amount of new input that is added to the cell state is determined by the input gate. How much of the prior cell state is retained is determined by the forget gate. How much of the current cell state is output is determined by the output gate. The LSTM cell's functions at each time step t are represented by the following equations:

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

Where xt is the input vector, ht is the hidden state, ct is the cell state vector, f_t , i_t , o_t are the forget, input, and output gate vectors, respectively, σ is the sigmoid function, *tanh* is the hyperbolic tangent function, W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c , are the weight matrices, and b_f , b_I, b_O , b_C are the bias vectors. Because LSTMs can learn from the temporal dependencies in the data and create predictions based on previous patterns, they are ideally suited for time series forecasting. The process of predicting a variable's future values from historical observations is known as time series forecasting. A time series forecasting problem for long short-term memory (LSTM) can be formulated in a variety of ways. These include the use of one or more input/output variables, one or more time steps, and a single-step or multi-step prediction horizon. LSTMs can be trained and programmed to find the best mapping function from the input sequence to the output sequence, depending on the task. We can see the architecture of the LSTM in the figure below.

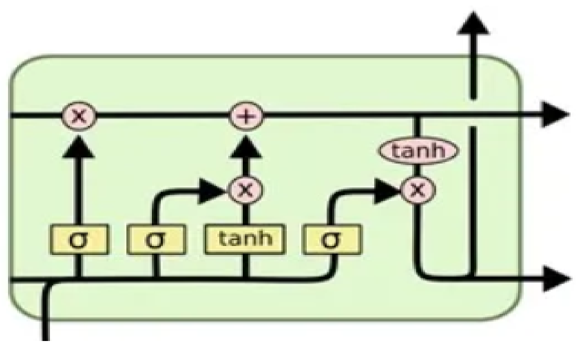

Figure 5. LSTM architecture

VI. MODEL SETTINGS

Each of the 5 models has its own set of parameters and hyperparameters. In this section, we will present the parameters used for each of the models:

i. ARIMA(p,d,q):
The following function was used to create the model SARIMAX(data.sales_diff, order=(12, 0, 0), seasonal_order=(0, 1, 0, 12),trend='c'). We used sarima instead of arima to fit the seasonality. order=(12, 0, 0) specifies a non-seasonal ARIMA model with an autoregressive component of order 12 and no differencing.
seasonal_order=(0, 1, 0, 12): This specifies a seasonal ARIMA model with no seasonal autoregressive or moving average components, a seasonal differencing of order 1, and a seasonal period of 12 (months). trend='c': This specifies a constant trend component.

ii. Linear Regression:
For this model, we just used the default function linearregression() without any parameters.

iii. Random Forest regressor: For this model, we specified the n_estimators to be 100 and the maximum depth of the tree to be 20.

iv. XGBoost: the regressor configuration is as follows: n_estimators were set to 100, the maximum depth is 3, a learning rate of 0.2, and squared error loss function.

v. LSTM: For the LSTM, we used the following architecture: an LSTM layer with 4 units, followed by two dense layers with 1 unit each. The two Dense layers are used to map the output of the LSTM layer to a single output value. The model is evaluated by the mean squared error loss function and the Adam optimizer..

VII. PERFORMANCE EVALUATION

As part of the performance evaluation of the models, the scores of the Root Mean Square Error, Mean Absolute Error and R-squared were obtained to evaluate the overall performance. The R-squared can range from 0 to 1. The closer to a value of 0 means that the model does not have any variability in the target variable, the closer to 1, the model explains perfectly the variability in the dependent variable.

a) Linear Regression

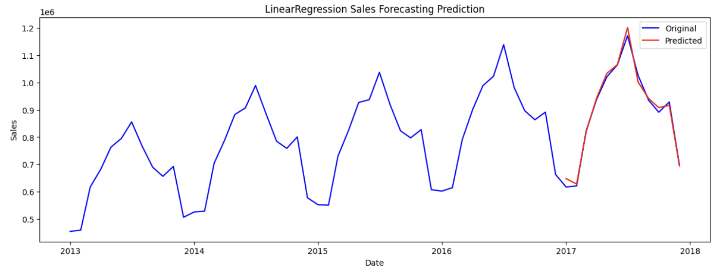

Figure 6. Linear regression forecasting

RMSE: 16221.040790693221
MAE: 12433.0
R2 Score: 0.9907155879704752

b) Random Forest regressor

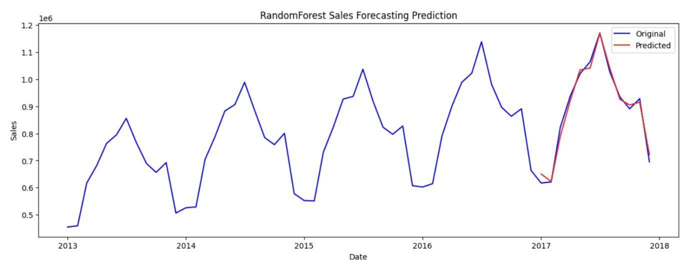

Figure 7. Random forest forecasting

RMSE: 19432.876826399122
MAE: 16575.083333333332
R2 Score: 0.9866748787757915

c) XGBoost

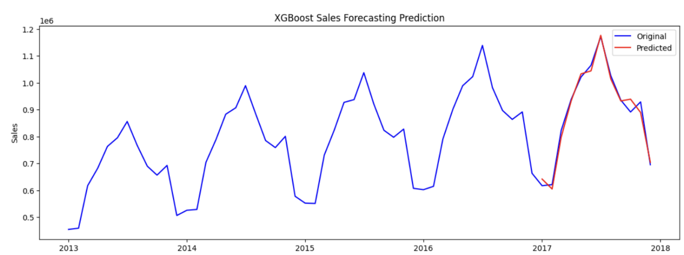

Figure 8. XGBoost forecasting

RMSE: 22730.366245766185
MAE: 18379.166666666668
R2 Score: 0.9817690273475503

d) LSTM

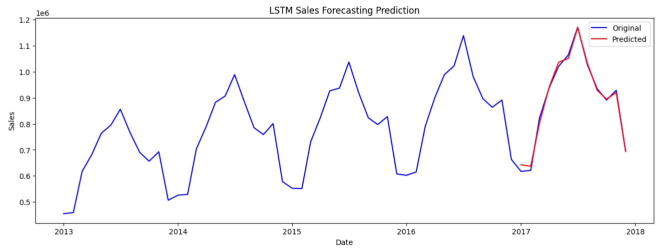

Figure 9. LSTM forecasting

RMSE: 12465.964008718565
MAE: 9987.0
R2 Score: 0.9945166153728947

e) ARIMA: in this model we plotthe difference between sales instead of the sales themselves.

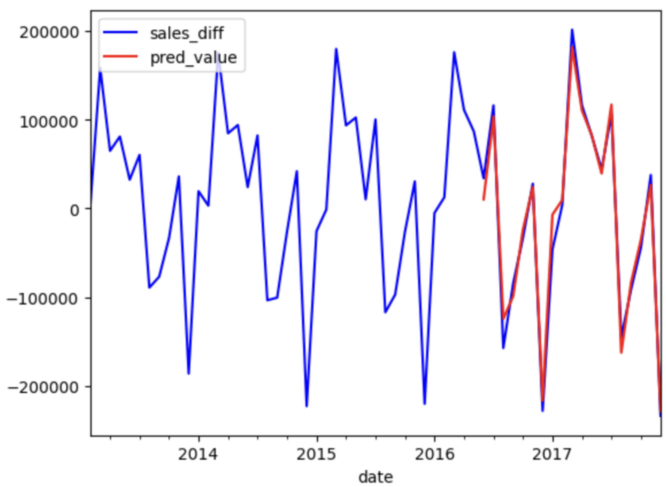

Figure 10. ARIMA forecasting

RMSE: 14959.893464767041
MAE: 11265.335750106897
R2 Score: 0.9835642876266992

	index	RMSE	MAE	R2
0	XGBoost	22730.366246	18379.166667	0.981769
1	RandomForest	19432.876826	16575.083333	0.986675
2	LinearRegression	16221.040791	12433.000000	0.990716
3	ARIMA	14959.893465	11265.335750	0.983564
4	LSTM	12465.964009	9987.000000	0.994517

Figure 11. Performance comparative analysis.

Considering the results, the best models with the best performance are LSTM followed by ARIMA. The results reflect that LSTM has the highest R-squared score than all the models because it is closer to 1 and the RMSE and MAE scores are the lowest of all the 5 models compared. As for the ARIMA model, the scores of the RMSE and MAE are the second lowest with 14959.893465 and 11265.335750 respectively. With XG Boost the prediction is within 1.12% of the actual.

VIiI. RESULTS AND COMPARISON

The forecast error is the difference between the prediction and the actual observed values, and it must be analyzed to determine the accuracy and reliability of each model. The error is expressed as:

$$E_t = F_t - D_t$$

The metrics used to evaluate and compare the prediction are RMSE (Root Mean Square Error) and the MAE (Mean Absolute Error). RMSE measures the average size of the error between the prediction and the actual values, assigning more weight to large errors. It is mostly used if large errors are critical since it is sensitive to outliers and puts more weight on them. RMSE is expressed as:

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

The Mean Absolute Error (MAE) is used to measure the average of absolute errors between the prediction and the actual values without the consideration of their direction. Is expressed as:

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

where,

n: *number of observation*

y_i: *the actual value of the i^{th} observation*

$\widehat{y}_i$: *the predicted value of the i^{th} observation*

To compare the results and choose the most suitable model, the results were analyzed according to the metrics mentioned.

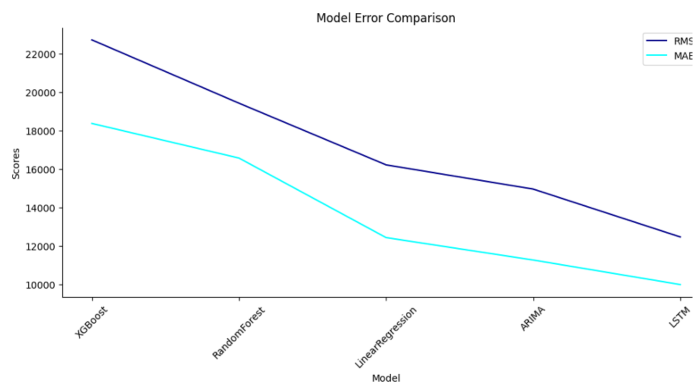

Figure 12. RMSE (Root Mean Square Error) and the MAE (Mean Absolute Error) comparison can be observed in the XG Boost, Random Forest, Linear Regression, Arima and LSTM models.

Considering the two metrics, the results show that the most accurate models are LSTM and ARIMA since they obtained the smallest error score in comparison with the rest of the models.

IX. CONCLUSION

In supply chain demand prediction, the choice of a model relies on the nature of the data and the characteristics within it.

a. XG Boost: Mainly used for data with categorical features. It predicts considering different variables such as product types, promotions, demographics, etc.
b. Random Forest: this method is useful for different prediction scenarios due to the combination of decision trees which allows us to make predictions with both numerical and categorical data.
c. Linear regression: Involves the assumption of relationships between the features and demand.
d. ARIMA: Best used for data with patterns, such as trends or seasonality since it can detect temporal dependencies in data.
e. LSTM: Method that is mostly effective with data that shows temporal dependencies.

In conclusion, the ARIMA model is a model that is preferred because of its versatility, showing high accuracy of results and attributes that are straightforward. Among the models reviewed through the course, ARIMA involves the moving average tool and was chosen due to its effectiveness and simple application. The use of it simplifies the prediction process and yields an accurate result, which is very important in supply chain management because forecasts are the basis for decisions and the benefits include:

1. Accuracy in forecast has an impact on optimization of inventory levels, maintaining optimum levels of inventory leading to cost efficiency.
2. Manufacturing planning can be scheduled according to the forecast, reducing the risk of underproduction or overproduction.
3. Enables good coordination between suppliers, manufacturers, retailers, improving the flows in the supply chain ensuring a reduction in lead times and improving responsiveness.
4. Accurate forecasting reduces the bullwhip effect by providing reliable demand signals on the supply chain.
5. Understanding future demand provides information to mitigate risks effectively regarding new products, possible market expansion

X. REFERENCES

[1] Mentzer, J. T., & Moon, M. A. (2004). *Sales forecasting management: a demand management approach*. Sage Publications.
[2]Store item demand forecasting challenge https://www.kaggle.com/competitions/demand-forecasting-kernels-only/data
[3] Autoregressive integrated moving average https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
[4] Dastan Maulud, A. M. (2020). A Review on Linear Regression Comprehensive in Machine Learning. Journal of Applied Science and Technology Trends, Vol. 1 No. 4.
[5] Balraj Singh, P. S. (2017). Modeling of impact of water quality on infiltration rate of soil by random forest regression. Springer, 2.
[6] Dutta, A. (2023, December 06). Random Forest Regression in Python. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/random-forest-regression-in-python/
[7] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. https://doi.org/10.1145/2939672.2939785
[8]Hochreiter, S., & Schmidhuber, J"urgen. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.