Demand Forecasting On Supply Chain Using ML And NN

by

Naoshin Anzum Hridi
18301065
Md Sharior Hossain Farhan
18301266
Md. Junaed Abed
18101349
Mohammad Nafiz Fuad Rafsan
18101558

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science Engineering

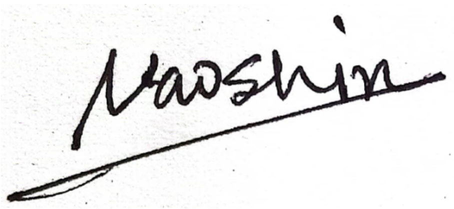
Department of Computer Science and Engineering
Brac University
May 2022

Declaration

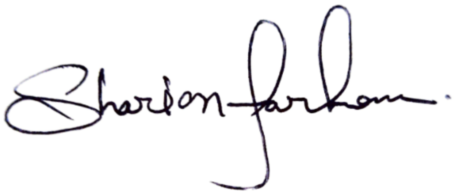
It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

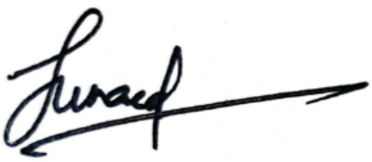
Student's Full Name & Signature:



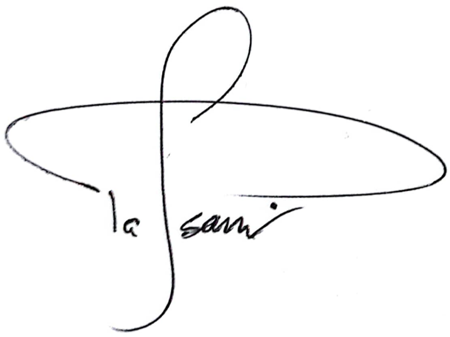
Naoshin Anzum Hridi
18301065



Md. Sharior Hossain Farhan
18301266



Md. Junaed Abed
18101349



Mohammad Nafiz Fuad Rafsan
18101558

Approval

The thesis titled "Demand Forecasting On Supply Chain Using ML And NN" submitted by

1. Naoshin Anzum Hridi (18301065)
2. Md Sharior Hossain Farhan (18301266)
3. Md. Junaed Abed (18101349)
4. Mohammad Nafiz Fuad Rafsan (18101558)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 26, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain

Assistant Professor
Department of Computer Science and Engineering
Brac University

Thesis Co-ordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Demand forecasting is mainly a process whereby analyzing historical sales data, strategic and operational strategies are devised in order to estimate customer demand. One of the most fundamental aspects of supply chain management is inventory management, its major goal is to cut expenses, boost sales and profits, optimize inventory, and most importantly, promote customer loyalty. The process of extrapolating relevant sales data may be separated into qualitative and quantitative forecasting, with each relying on multiple sources and data sets. When there is previous sales data on certain items and a predetermined demand, the quantitative forecasting approach is employed. It necessitates the application of mathematical formulas as well as data sets such as financial reports, sales, and income numbers, as well as website analytic. The qualitative technique, on the other hand, is based on new technologies, pricing and availability changes, product life cycles, product upgrades and most significantly, the forecasters' intuition and experience. Machine learning, clustering, time series analysis, neural networks, KNN, support vector regression, support vector machines, regression analysis, and deep learning are some of the approaches used to anticipate demand. A majority of study has gone into improving demand forecasting, which will enhance supply chain sales and profitability. To do that the researchers mainly focused on using machine learning or deep learning as its main methodology and others like support vector algorithm, time series analysis. However, to our best knowledge, only a handful of research is done using hybrid model consists of both deep learning and machine learning as its main methodology. That is why we want to concentrate on using hybrid models to develop dynamically configurable demand forecasting which eventually will give us promising results.

Keywords: Demand forecasting, supply chain sales, deep learning, machine learning, LSTM, DNN, Prophet, NeuralProphet, ARIMA, SARIMA, CNN, RNN

Dedication

We dedicate this work to our parents and teachers....

Acknowledgement

Our heartiest gratitude to the Great Almighty and our parents for enabling us to complete our undergraduate studies

We are thankful to our supervisor Dr. Muhammad Iqbal Hossain sir for his kind support and advice in our work. He helped us whenever we needed help. This thesis would not have been possible without his guidance and we are grateful to him for his selfless help.

Last but not the least, we thank our Brac University peers who helped us with information whenever we needed them.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ACF Autocorrelation Function

$ARIMA$ Autoregressive integrated moving average

CNN Convolutional Neural Network

DNN Deep Neural Network

GRU Gated Recurrent Unit

$LSTM$ Long Short-Term Memory

$MAPE$ Mean Absolute Percentage Error

MSE Mean Square Error

$OLS - SARIMA$ Ordinary least squares-Autoregressive integrated moving average

$PACF$ Partial Autocorrelation Function

$RMSE$ Root Mean Square Error

RNN Recurrent Neural Network

SGD Stochastic Gradient Descent

Chapter 1

Introduction

1.1 Motivation

Demand forecasting is known as the process of analyzing and estimating future demand for a product, outcome, or service utilizing predictive analysis of past data. It has evolved into one of the most critical supplies chain management issues. Even under the best conditions forecasting demand is tricky. Retailers, small grocery shops, and other organizations in a variety of industries are looking for big data and predictive analytic-based automated demand forecasting and inventory management solutions [1]. Demand forecasting also helps with important business activities including raw material procurement, financial planning, risk assessment, and production planning [2]. Most importantly, forecast accuracy helps firms to minimize stock-outs and overstocking, as well as save money, increase operational efficiency, and improve customer satisfaction [3]. Despite the fact that forecasting is a critical component of most retail businesses, there is still a lot of misconception regarding what a sales vs demand forecast is. To predict the future, all merchants employ some type of forecasting, and the rationale is straightforward[3]. It's difficult to create a profitable and long-lasting retail business without knowing how many products may sell tomorrow. The retailers may wind up buying too many of the incorrect goods and not enough of the proper inventory, causing clients to become enraged and lose a significant amount of money[4]. This is why, especially in terms of forecasting, merchants are becoming increasingly data-driven. The difference between being profitable and going bankrupt is a solid prediction.

Although many researchers already worked on demand forecasting using machine learning, time series analysis and few of them used deep learning to get the best possible outcome, no one focused on the improvement of small businesses as well as selectable features based on deep learning. We are planning to propose a framework where not only small businesses but also large ones can take advantage of the complex deep learning algorithm using this proposed framework. Using demand forecasting, we will handle product stocking (overstocking and out of stock) issues as well as market demand for items. For this purpose, we are going to use various methodologies of deep learning and machine learning like - ARIMA, OLS-SARIMAX, Prophet, Neural Prophet, CNN, LSTM methods for feature extractions. The suggested method will be put into action and evaluated using real-world data. Moreover, we are going to use these collections of data to train our algorithms. This improves the accuracy of our projections in terms of trend shifts and seasonality patterns.

We are now gathering real-world data and pre-processing it so that it may be trained with several algorithms to obtain a probability and compared to determine which one provides more accurate data and is relevant to our suggested model.

1.2 Research Problem

Despite the fact that 74% of retailers aspire to be data-driven, just 29% can connect analytic to action, according to Forrester Research(Forrester,1958). To put it another way, most merchants are still making educated guesses about their data. They have no idea how to utilize data to make accurate forecasts or how to use forecasting to make sound decisions. In 2001, Nike used a demand planning software which implementation caused them a $100 million loss in sales. Similarly, in 2014 Walgreens lost $1 billion dollars in forecasting blunders.

If we look at the current situation of the supply chain, market companies' retailers are focused on predictive analysis to boost productivity and profit while lowering expenses. They also aim to handle both "high stock" and "out-of-stock" issues. Customers' loyalty, as well as sales, are important to them. For this, effective operations management, businesses require accurate market forecasting and inventory and stock control systems[1]. In order to make strategic, tactical, and operational supply chain management decisions, businesses also require short- to long-term aggregate projections[3]. Forecast coherency is occasionally overlooked whereas, the emphasis in supply chain forecasting mostly depends on accuracy. The structure with a hierarchy of retail may be leveraged to produce more accurate and coherent predictions, which is a component of forecasting that is typically overlooked[3].

Demand for a specific item or brand is often accompanied by a number of unknown and uncertain factors, making it volatile and difficult, and challenging to predict. Unpredictability in demand is one of supply chain managers' key concerns as well as a challenge, since it may lead to significant forecasting errors, difficulties in the upstream supply chain, and unnecessary costs. In this paper[5], they analyze 843 genuine and measured time series with varying coefficients of variation (CoV) to see how promotion impacts volatility across the board. Demand volatility is caused by several intrinsic and extrinsic factors.

A range of factors, including promotions, weather, market trends, and season, can impact consumer behavior, all of which can contribute to demand volatility. Promotion is a common activity in the retailing sector, and it can generate demand fluctuations. The impact of promotion on-demand dynamics has been extensively researched in the literature[5]. In the research paper[6], we can observe that SPSS software was used to generate Multiple Linear Regression equations. The multiple linear regression model has less accuracy compared to the true value, according to the data. When random forest models are used to forecast the demand for bicycle rental, the accuracy and outcomes of multiple regression analysis are greatly enhanced. As a result, in order to strengthen the decision tree, this study presents a random forest model as well as a GBM packet.

Demand forecasting, as well as financial estimates, are two critical difficulties in supply chain management, according to the article [7]. Historical data or causal predictors are used in traditional forecasting approaches. Most are unable to account for the time-lag casualties between predictors and outcomes, as well as the supply-chain members' interacting dynamics. The following problems are highlighted in this study using a conceptual framework: (a) To account for product volatility and anticipate demand, time-series models are created. (b) In order to facilitate a financial analysis, vector autoregression is used to capture the interaction dynamics of the participants. Meanwhile, new factors such as ai, cloud computing, as well as IoT are driving the semiconductor industry ahead at lightning speed (technology push). By changing an influential predictor and evaluating it statistically and qualitatively to provide managerial insights, they look at how semiconductor firms estimate worldwide shipments of consumer electronics (demand forecasting) and the influence this has on a company's sales income.

Deep Learning has turned into a noteworthy component of the current generation of Time Series Forecasting models, leading to enhanced results, as a result of increased data and computational power offered in recent years. In the case of demand ML, forecasting, time series analysis, and Regression models can not always obtain accurate or most accurate results for the long-term trend, seasonality, stationarity, noise, and autocorrelation. Machine learning models have some limitations which can be overcome by deep learning methodologies. Mostly deep learning can work with long-term forecasts as it can handle missing values, complex pattern recognition, and model a sequence of data.

1.3 Research Aims and Objectives

While most of the work of demand forecasting is done in Machine learning and time series analysis there are a few works done in Deep learning. However, their works only focused on static features and how these features affect product location and time dimension for a short time. The main problem of demand forecasting regarding the supply chain is that if it is done with machine learning and time series analysis methodologies, it can not provide accurate results for long-term strategies and seasonalities due to mishandling of missing data and complex patterns. Although very few worked with deep learning in this field they also had some shortcomings regarding features and training the algorithms using a variety of datasets. Static features sometimes fail to give us accurate predictions for particular forecasting. In our case, we are proposing such frameworks which will choose features for a particular scenario that cover all the bases to give the most accurate results. Considering these perspectives our work progress would be like this:

1. We will first try to understand how demand forecasting works.

2. Next, we are going to collect real-life data of various supply chains or retail.

3. There is a possibility of raw data which will need to be pre-processed.

4. We'll go over the data briefly to make sure it's structured, accurate, and consistent with our test.

5. Then we will learn algorithms of machine learning and time series analysis to understand how to simplify forecasting.

6. Furthermore, we have to learn how to use data to get significant forecasting.

7. After that, we will learn suitable deep learning algorithms that will help us to develop models and apply them for demand forecasting.

8. After that, we are going to focus on utilizing our algorithm knowledge to find a way to make a selectable feature based on consumer demand.

9. A few data tests and pilots are required to work further.

10. We are going to train our models to extract features by using real-life datasets.

11. Lastly, we have to look through the statistical summary to know our progress.

Using deep learning methodologies it is possible to have faster and accurate demand forecasting for long-term trends. Most of the research work did not implement the new deep learning algorithms to make a more accurate prediction. We will implement these algorithms and check which one gives us faster and accurate predictions in demand forecasting. All of these works will give us a well-developed system for demand forecasting which will help to get the most accurate results in demand forecasting of the supply chain.

Chapter 2

Literature Review

2.1 Recurrent Neural Network(RNN)

In the research publication [8], the back-propagation through time training methodology was used to educate an RNN on a given training set. Every neuron in the hidden layer had recurrent connections that flowed back into all the neurons in the same layer in the process stage. The RNN was able to learn patterns over time as a consequence of this.

$$\begin{aligned}
&\boldsymbol{a}^{(t)}=\boldsymbol{b}+\boldsymbol{W}\boldsymbol{h}^{(t-1)}+\boldsymbol{U}\boldsymbol{x}^{(t)} \\
&h^{(t)}=\tanh\left(\boldsymbol{a}^{(t)}\right) \\
&o^{(t)}=c+\boldsymbol{V}\boldsymbol{h}^{(t)} \\
&\hat{y}^{(t)}=\operatorname{softmax}\left(\boldsymbol{o}^{(t)}\right)
\end{aligned} \tag{2.1}$$

2.2 ARIMA

They employed the ARIMA technique for time series analysis in their research article [1]. The conventional method for developing time series forecasting algorithms is to collect historical data, evaluate the data's underlying characteristics, and then apply the model to predict the outcome.

$$\begin{gathered}
Y_t=\beta_1 Y_{t-1}+\beta_2 Y_{t-2}+\ldots+\beta_0 Y_0+\epsilon_t \\
Y_{t-1}=\beta_1 Y_{t-2}+\mathring{A} Y_{t-3}+\ldots+\beta_0 Y_0+\epsilon_{t-1}
\end{gathered} \tag{2.2}$$

They used ARIMA to forecast baseline demand in the paper [5]. They employed dummy variables and fitted an ARIMA model to the prior non-promoted demand to capture promotional and post-promotional demands. In the absence of promotional stimuli, the baseline demand is calculated. The models are developed in R, and in Hyndman and Athanasopoulos' work, the 'auto.arima' function from the 'forecast' package is used to fit and define the parameters of the ARIMA model (2014). In contrast to an exact amount, baseline demand is an estimate based on facts. The ARIMAX system is a variation of the ARIMA model that incorporates an explanatory variable. It's typically a good idea to add extra covariate data as a predictor in the forecasting model when more covariate data is available. Price can be used as a covariate in the equation to get ARIMAX.

2.3 Exponential Smoothing

It is stated in the research paper[9], exponential smoothing is believed to be the most basic statistical approach used throughout forecasting electricity demand. The popularity of this program can be due to its accessibility of use, computational efficiency, and satisfactory accuracy. Smoothing settings are adjusted every five minutes in VSTLF in order to increase predicting accuracy. In each period, three parameters are adjusted and modified to better forecast the level, trend, and seasonality of the load series.

The simple exponential smoothing strategy adopts less processing and is utilized when data patterns in historical data do not exhibit a cyclic variation or trend. Holt's technique, widely known as the double exponential smoothing method, is used for a time series with a trend in this example. The Holt-Winters method works well for recurring time series because it can capture both tendency and periodicity in historical data.

2.4 Random Forest

The author mainly focuses on[6], the conventional strategy which was used to create the multiple linear regression model. Using SPSS software, a multiple linear regression equation was generated. When the data is compared to the true value, Apparently, the multiple linear regression model is less precise. As a result, the framework proposed in this study is a random forest. to improve the decision tree. Multiple regression analysis outcomes and accuracy are significantly enhanced when a random forest model is employed to anticipate bicycle rental demand. After looking through the data again, the author realized that the aspects contained in the season, such as the weather, are such non-recursive factors.

2.5 SVM

The author mainly focus in ths paper[1], regarding continuous variable prediction issues, they utilized the SVM model. SVR is considered in this research to anticipate sales demands based on the input factors. In the research paper[8], Each data item was charted as a point in n-dimensional space (where n is the number of features they have), with the value of each feature serving as the coordinate value. After that, the author classified the data by determining the hyper-plane that clearly separates the two classes.

2.6 LSTM and LSTM Autoencoder

2.6.1 LSTM

In paper [3], the LSTM (long short-term memory) network has been used by the author as a deep learning methodology. The authors express their opinion that a top-down approach is less preferable than bottom-up forecasting as it gives more accuracy when using a point-of-sale data approach for both offline and online supply chains. The estimates utilizing the suggested framework beat direct forecasts on all three metrics, ARMAPE, ARMSE, and ARMAE.

In this paper [10], the author offers a market forecasting method predicated on multi-layer LSTM networks. The recommended method utilizes a grid search methodology to dynamically select the highest accuracy model for a given time series by considering alternative combinations of LSTM hyper-parameters. It can find nonlinear patterns in time series data while taking into consideration non-stationary datas' essential characteristics and features. LSTM is a time-series data forecasting extension of RNN. An LSTM differentiates from an RNN in that it can store long-range temporal dependency information and map input and output data correctly.

$$
\begin{aligned}
& f_t = \sigma_g (W_f \times x_t + U_f \times h_{t-1} + b_f) \\
& i_t = \sigma_g (W_i \times x_t + U_i \times h_{t-1} + b_i) \\
& o_t = \sigma_g (W_o \times x_t + U_o \times h_{t-1} + b_o) \\
& c_t' = \sigma_c (W_c \times x_t + U_c \times h_{t-1} + b_c) \\
& c_t = f_t \cdot c_{t-1} + i_t \cdot c_t' \\
& h_t = o_t \cdot \sigma_c (c_t)
\end{aligned}
\tag{2.3}
$$

This paper [11] gives us how the LSTM algorithm works. Cells, or system and framework modules, constitute each LSTM, which collects and processes data streams. The cells resemble a transmission line, which joins one module to the next, transferring data from the past and collecting it for the future. Owing to the use of gates in each cell, data in each cell can be eliminated, censored, added, or restored for future cells. As a direct consequence, the gates, which are based on a sigmoidal neural network layer, allow cells to accept or reject the input.

2.6.2 LSTM Autoencoder

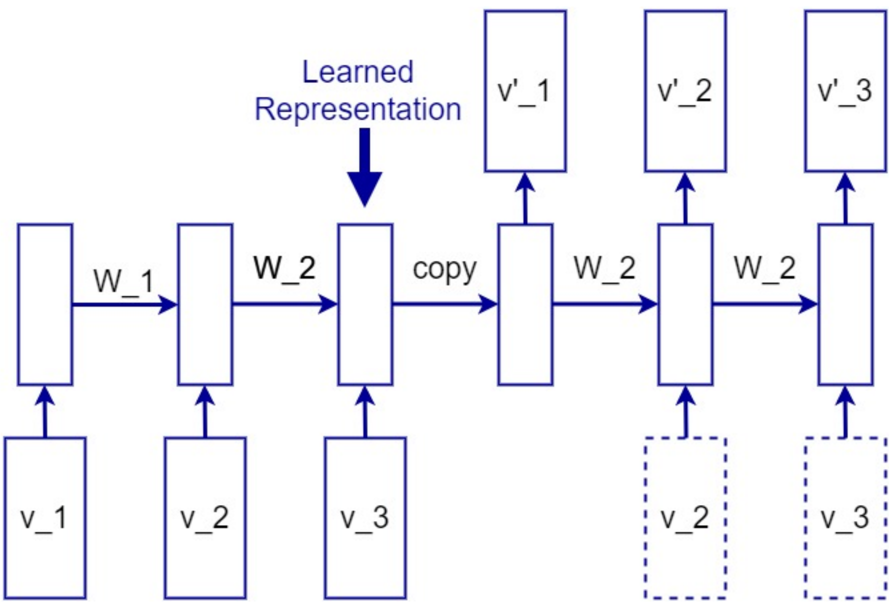


Figure 2.1: LSTM Autoencoder Model

In this paper[12] the author used Autoencoder which is a type of unsupervised neural network that uses data to train the optimum encoding-decoding technique. The inner layer contains a layer for input and output and encoder and decoder neural network along with latent space. The author uses the LSTM autoencoder so that the encoder compresses the data before sending it to the network, while the decoder is used to decompress the encoded form before sending it to the output layer. The output that was encoded is compared with the original data along with the mistake to adjust weights in back-propagation.

The autoencoder's primary function is not to simply replicate the input to the output. The autoencoder is driven to study the prominent characteristics of the training data by encapsulating the concealed space to have a lower dimension than the input, i.e. "less than" m. To put it another way, a significant characteristic of the autoencoder's design is that it minimizes data dimensions while retaining the majority of the data structure's information. LSTMs are well suited to time series forecasting due to their capacity to discover trends in information over a long duration. The LSTM cell is utilized in multivariate data to extract correlation structure.

2.7 GRU

In this paper [13], The authors present GA-GRU which is a hybrid forecasting model that combines a Genetic Algorithm(GA) with GRU. They have used GA to determine there are five types of GRU model parameters, incorporating window size, the number of nodes in the hidden layer, batch size, epoch duration, and initial learning rate, because numerous hyperparameters affect its performance. They have created the GA-GRU hybrid model to obtain GRU hyper-parameters using GA. This study comprises three experiments to test the effectiveness of GA-GRU vs. other forecasting models comparison, k-fold cross-validation, and sensitivity assessment of the GA parameters. The results reveal that GA-GRU outperforms other forecasting models in terms of percent deviations, suggesting that the mutation factor be adjusted to 0.015 and the crossover probability to 0.70. The use of NN is likely to improve demand forecasting accuracy. However, this research proposes a novel forecasting approach for anticipating product demand in logistics that is based on GRU.

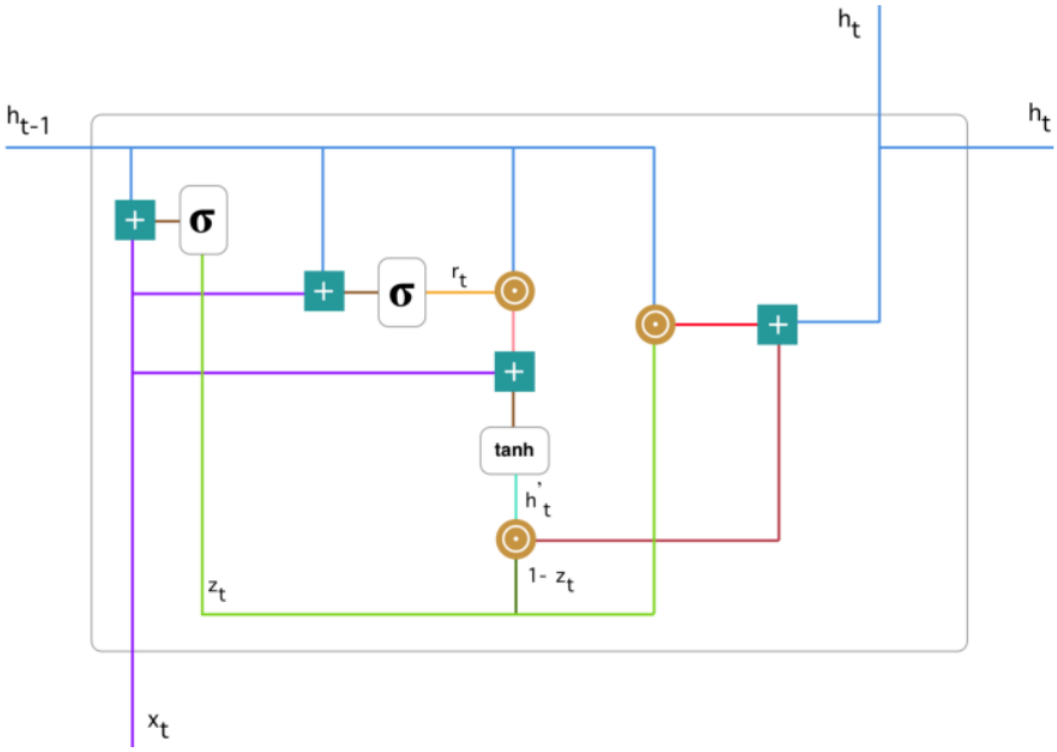


Figure 2.2: Single Gated Recurrent Unit

2.8 Facebook Prophet

The prophet is a Facebook Meta algorithm that can look at frequency, pattern, seasonal influence, and other time-series properties[14]. In terms of tendency, the Prophet method allows instantaneous points to be fitted piece-wise and linearly. Fourier series are used to create a cyclical model in the case of iterations. Users can submit holidays and weekends and crucial days in a structured format for assessing vacations and emergencies. The Prophet is a well-known time series integration system that follows this structure below:

$$y(t) = g(t) + s(t) + h(t) + \xi \tag{2.4}$$

Sometimes, in sales periodic data there are some unusual fluctuations that are caused by occasional sales and promotions. However, in the Prophet algorithm, g(t) is known as a trend function that takes non-periodic data in a time-series analysis whereas, in the equation s(t) is the periodic change in a week or a year. In the given function h(t) takes the statistical anomaly which occurs because of specific occasions and promotional sales. The function h(t) follows the normal distribution that mitigates unpredicted and random fluctuation.

i) Flexibility: Periodicity may be readily altered, and users can make a number of assumptions around trends.

ii) Scope of application: Missing values and the measures which are not taken at regular periods are acceptable.

iii) The fitting process is quick.

iv) Parameters that are simple to understand: this allows the researchers to modify the settings regarding the case.

2.9 Related Works

In the research paper [4], The findings reveal that their technique has a 96.04 percent evaluation accuracy, which is greater than the SVM method and the Logistic method and has superior logic. Adult, Wine, and Iris were chosen from the traditional data center and entered into the risk assessment model which used DBN as its main methodology. Under the constraint of Gaussian distribution, there are several parameters set up. Here, they set initialization to a random lower value, define the default development pace to 0.1, and update the rate to 0.01 as parameters. The following data sets are classified with an increased incidence of average classification accuracy, implying that this model may be utilized to resolve the classifier model using the framework.

In the research paper [1], they recommend an innovative approach to enrich demand forecasting, which is one of supply chains' most pressing challenges. The application of the boosting approach which acquires more weight to the demand forecasting model is another unique feature of this study. This improves the accuracy of our projections in terms of trend shifts and seasonality patterns. This study employed nine different algorithms to get the best results, including ML, DL, and EL (Ensemble learning). They also utilized ARIMA techniques as well as three distinct regression models. They employ a traditional technique to collect historical data, evaluate it, and apply it for model prediction in this model. They also employed time series and regression methods, such as Support Vector Regression, Exponential Smoothing, Holt-Winters, and Time Series and Regression Methods, among others. Nine distinct time series techniques are used in our proposed system, including Moving Average (MA), Exponential Smoothing, and Holt-Winters. In addition, because data takes a long time to train, we employed PCAmultiple methods of feature identification.

In the research paper[8], the naïve forecast is used as a benchmark against the other methods of the evaluation process as it is known as one of the most basic forecasting methodologies. To extrapolate future variables, it takes the most current value of the variable of relevance. Using a number of historical changes in demand measurements as independent variables, the multiple linear regression model attempts to predict the change in demand. Linear regression can only utilize two independent variables that seem to be concurrent - (a) There is an independent variable and (b) there's also a dependent variable. The independent variable is the parameter that is used to calculate the dependent variable or result.

In this research paper[5], They looked at 843 genuine demand time series, some of which had non-identical coefficients of variation, and they concentrated on marketing and advertising, which is the major cause of volatility across the board. The CoV is computed by dividing the sample normal deviations as just a factor of the mean of the sample data. The coefficient of variation is a magnitude metric for comparing the relative variance of several series. It is utilized as a demand variability criterion in the supply chain and indicates data ambiguity. They presented that forecasting demand for various coefficients of variation implies the usage of multiple models capable of capturing the underlying behavior of demand series, which represents major problems owing to a broad spectrum of demand patterns They offer a hybrid approach to anticipate demand by dividing demand into two categories: baseline and promotional. They observed that ARIMA with covariate (ARIMAX) is excellent at forecasting erratic series of demand, whereas ETSX was much worse. For a range of demand categories with varying CoV values, SVR and DLR models produce credible forecasts. The ARIMAX and combination models outperform the other models provided in terms of inventory performance. The hybrid algorithm also has minimal inventory costs and displays consistent performance across different series with varied CoVs.

In this research paper [15], they offered a system for creating and configuring a supply chain dynamically in response to order parameters. The orders are automatically routed by the framework to the appropriate suppliers which helps reconfigure the supply chain in real-time.

In this research paper [16], the researchers presented a comprehensive skeleton to show how data and information affect decision-making, especially in management and economics. They analyzed the relation between big data and forecasting methods. They came to the conclusion that enterprises can use big data analytics for strategic demand plans.

In supply chain management, demand forecasting, as well as financial estimation, are two crucial aspects [7]. Traditional forecasting approaches mainly rely on either past statistical information (time-series) or causal factors to make a prediction (regression). Though many strategies have been presented, the majority of them are still unable to account for the time-lag causalities between forecasters and consequences, as well as the supply-chain members' interaction with other individuals. Interactive dynamics can be aggressive or collaborative for the chipmakers for packaging and testing (PT) businesses.

In paper [13], data analytics are required to translate data into relevant information given a huge quantity of data. Machine learning, or the automation of knowledge works, is one of the interruptive predictive analytics that is anticipated for market growth, employment, and inequality. The majority of applications for machine learning (65%) have been implemented in the industry, with the agriculture sector being debated in a modest number of papers. The practical impact of this article is to expose the current learning challenges of machinery to enable stakeholders and policymakers to plan transformation measures effectively. This disruption offers a range of opportunities and difficulties for companies and supply chain manufacturers.

In the research paper [17], Artificial intelligence techniques employing artificial neural networks are used to predict demand forecasting techniques. The prediction methodology is also implemented in order to improve future time series prediction.

This is due to the influence of several factors in the retail trading system's on-demand function. It was also discovered that as the forecasting time gets short, the ANN gives better prediction accuracy.

The paper [18] shows a rundown of the predictive analytic-enabled supply chain management functions (SCMF). Logistics, Procurement, Order Fulfillment, Transportation, and also purely technical elements are all addressed. are included in supply chain management functions. Among the most common uses of BDA-enabled predictive models are demand management and source risk assessment. This is a well-detailed piece of relevant literature that was gathered and reviewed.

In this paper[14], the author used the Attention Mechanism. Furthermore, The attention mechanism is a system for allocating resources that improve the precision of the model. Focusing only on information that is valuable to us is a key principle. The significance of bits of information is determined in part by their probability distribution.

In this paper[13], The Genetic Algorithm (GA) is a very well-known meta-heuristic algorithm that solves problems by using a stochastic optimization technique. The fundamental benefit of GA is that it helps you to quickly identify an optimum and near-optimal solution to a large-scale problem. GA is derived from natural evolution and was invented by Holland. The set population, or the original set of random solutions, is the starting point for GA. Then GA chooses chromosomes that have a high fitness value. GA comprises three operators for this process: selection, crossover, and mutation. As a result, the efficacy of GA is determined by the original population as well as the operators. The following are the three GA operators and fitness functions: Selection, Crossover, Mutation, and Fitness Function.

Chapter 3

Methodology

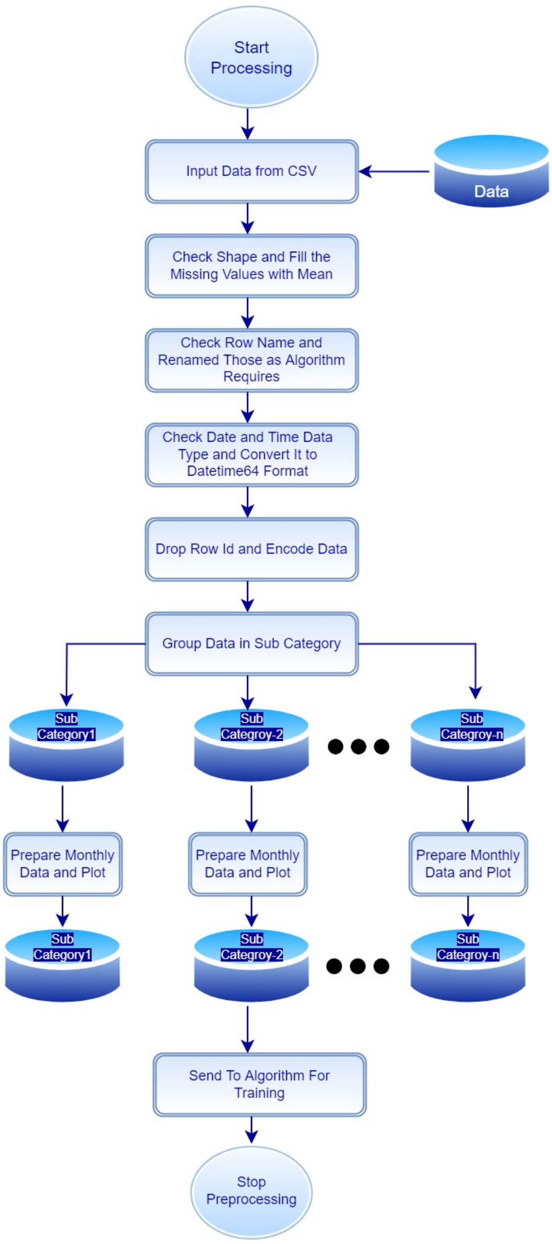


Figure 3.1: Data Pre-processing

The primary goal is to see whether forecasting approaches provide the best predictions in terms of reduced forecast failures and greater forecast accuracy. In time series forecasting, there are a variety of probabilistic models to choose from.

Our work technique has been broken into four stages in order to construct our proposed system. It entails integrating ARIMA, RNN, and Prophet into action. The technique is depicted in Figure.1 as a block diagram: Dataset Loading and Data Pre-processing. The following is a representation of our methodology:
Dataset Loading:
1. Dataset Loading
2. Data Pre-processing
3. Training the train data of the dataset with ARIMA, RNN, GRU, and Prophet
4. Evaluate the Performances

3.1 Dataset Loading:

Importing the dataset is the initial step in the data processing. This was accomplished by the use of a drive to pull data from the cloud. Because most datasets are in Comma Separated Value (CSV) format, we must go through various processes to get the values from them. We utilized the pandas' library, which is based on the NumPy library. Panda is an important tool for data structuring and analysis. As a result, we utilized the panda method to read and alter our CSV file after importing the NumPy library.

3.2 Data Pre-processing:

3.2.1 Sources of data

The goal of data processing is to find the best feature combinations feasible. Data mining relies heavily on feature combinations. When an algorithm's bottleneck is hard to overcome, good feature pairings can often result in positive outcomes. In this section, Superstore Sales Dataset was taken as an example. We can evaluate the data distribution and identify useful characteristics from this data set by performing visual analysis. The data is then processed, and the feature set for the training model is created.

	Order_ID	Order_Date	Ship_Date	Ship_Mode	Customer_ID	Customer_Name	Segment	Country	City	State	Postal_Code	Region	Product_ID	Category	Sub_Category	Product_Name	Sales	Quantity	Discount	Profit
0	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.9600	2	0.00	41.9136
1	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3	0.00	219.5820
2	CA-2016-138688	2016-06-12	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036	West	OFF-LA-10000240	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...	14.6200	2	0.00	6.8714
3	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	FUR-TA-10000577	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	0.45	-383.0310
4	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	OFF-ST-10000760	Office Supplies	Storage	Eldon Fold 'N Roll Cart System	22.3680	2	0.20	2.5164

Figure 3.2: Input Data Before Pre-processing

In our dataset, we have 9994 rows and 21 columns. As we are going to forecast the sales of a particular product, we have to exclude unnecessary attributes such as Ship Mode, Customer ID, Row ID, etc. In figure - 3.2. the raw data before pre-processing is shown. In figure - 3.3 all the data types are shown. We had to change the data type of Sales to int.

```
Order_ID                 object
Order_Date       datetime64[ns]
Ship_Date                object
Ship_Mode                object
Customer_ID              object
Customer_Name            object
Segment                  object
Country                  object
City                     object
State                    object
Postal_Code               int64
Region                   object
Product_ID               object
Category                 object
Sub_Category             object
Product_Name             object
Sales                     int64
Quantity                  int64
Discount                float64
Profit                  float64
dtype: object
```

Figure 3.3: Data Types

For the pre-processing, we have to select only 2 columns. In our research, we have chosen the Binders sub-category to forecast. Therefore, we have grouped the sales data of that particular subcategory as we can see in figure 3.4.

order_date	sales
2014-01-04	3
2014-01-06	609
2014-01-07	10
2014-01-13	8
2014-01-16	18
2014-01-19	32
2014-01-20	115
2014-01-26	10
2014-01-28	3
2014-02-02	18
2014-02-03	96
2014-02-04	99
2014-02-06	8
2014-02-11	77
2014-02-15	21
2014-02-16	1
2014-02-21	8
2014-02-23	4
2014-03-03	150
2014-03-05	49

Figure 3.4: Input Data After Pre-processing

3.2.2 Data visualization analysis and effective feature selection

We display the original data and choose an appropriate feature set in order to gain the basic data qualities that impact sales.

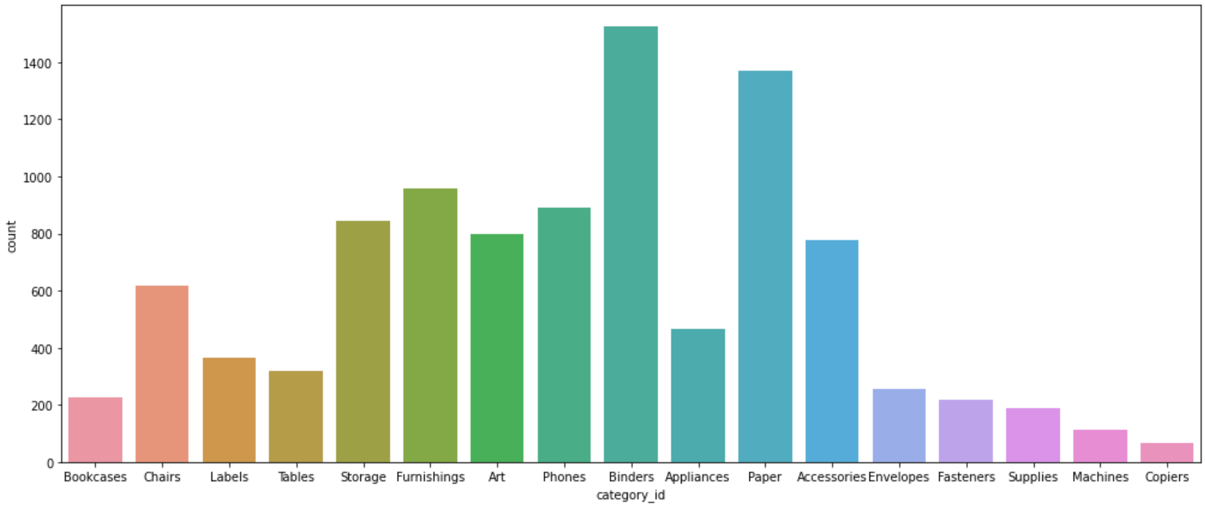


Figure 3.5: Each Sub-Category Count plot

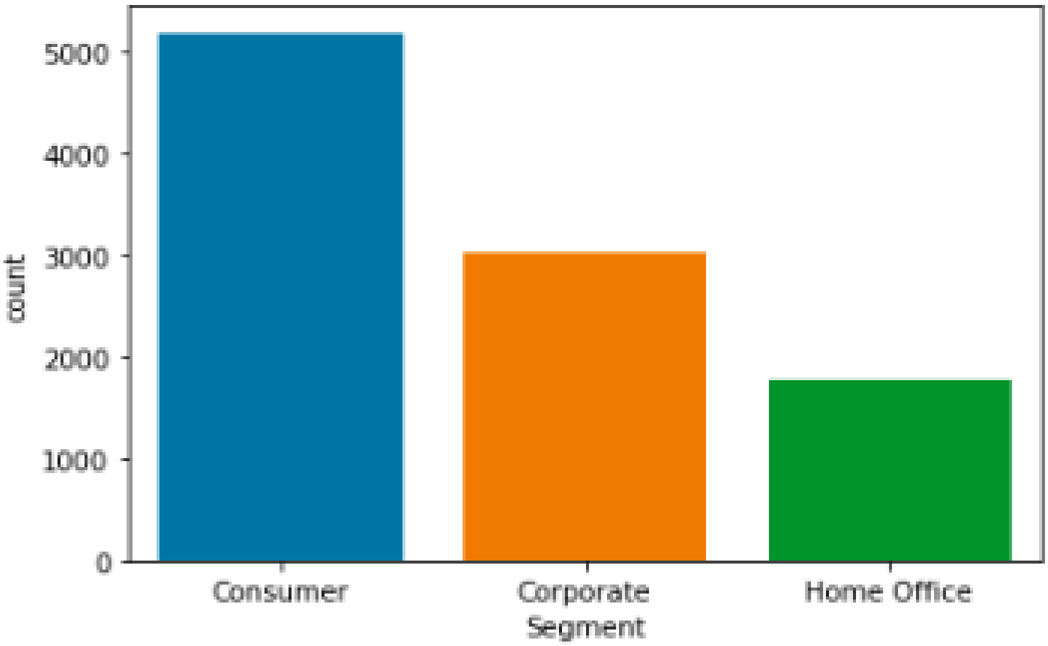


Figure 3.6: Each Segment Count Plot

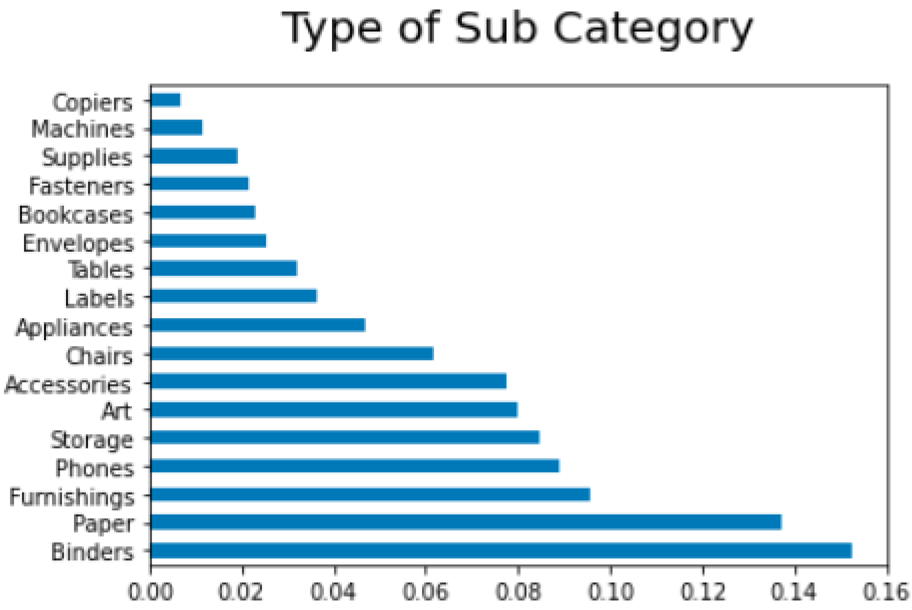


Figure 3.7: Types of Sub Category

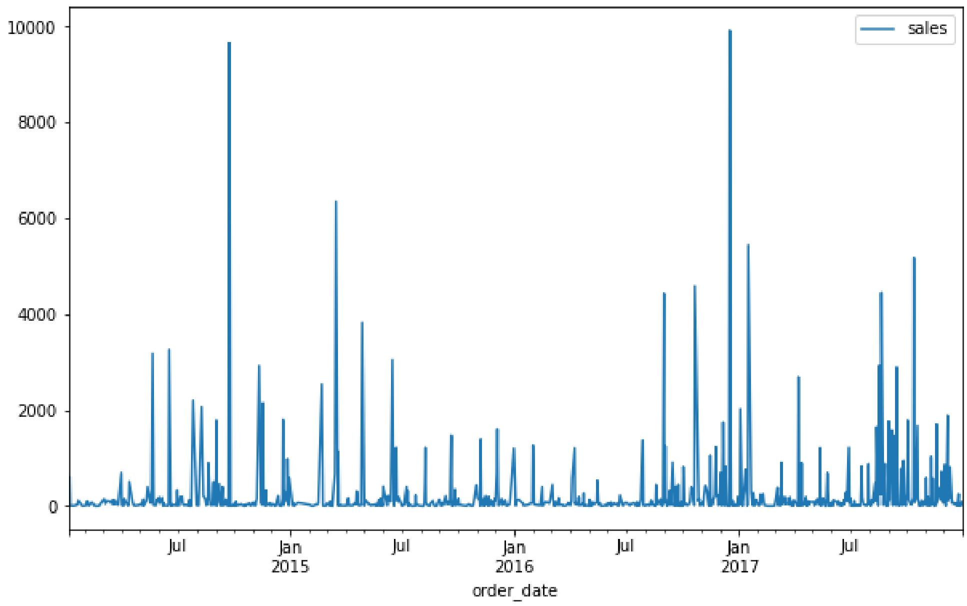


Figure 3.8: Order Date vs Sales Data Frame Plot

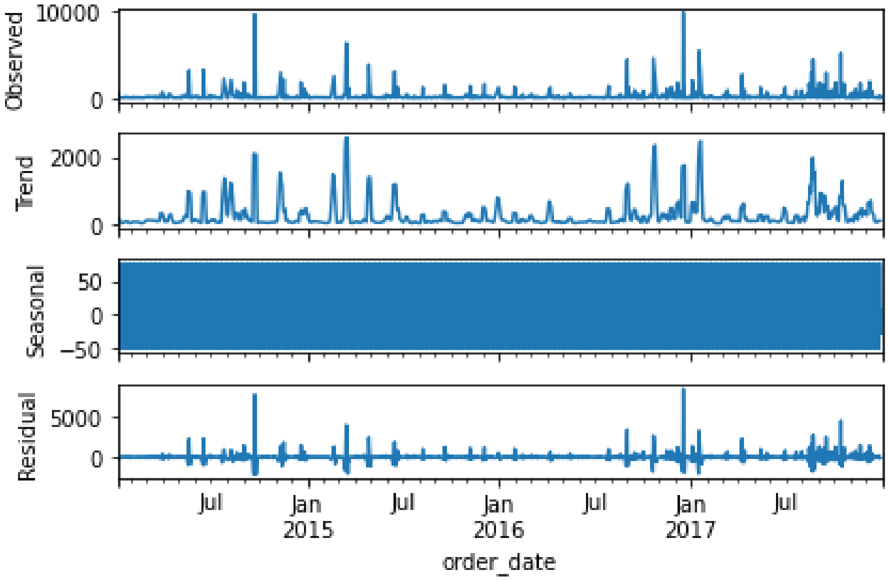


Figure 3.9: Seasonal Decomposed

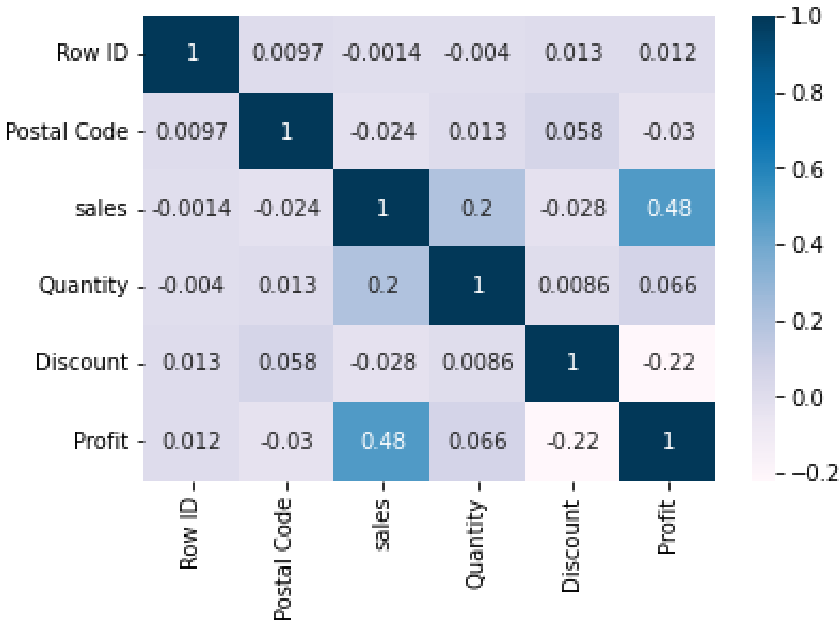


Figure 3.10: Heat-map of Dataset

3.2.3 Feature engineering

Following the procedures below, upon the collecting of the appropriate feature set, we clean up the dataset:

a. Fill in the documents - The mean value of total sales can be used to replace the missing value of sales data. Also, in some cases, we interpolate some values using the linear method "time" which Works on daily and higher resolution data to interpolate given length of interval. The interpolation method will fill NaN values using interpolation method.

b. Process outlier value - In our data sets, some of the values were outliers as there seems to be promotional sales data that causes it. For this, we are either going to drop the value or we are going to replace the value with the mean value of total sales.

c. Remove irrelevant features - To decrease the model's interference and complexity, remove features that are superfluous, redundant, or unrelated to the goal.

d. Encoding features - The purpose of encoding is to convert float or string values to integers for better results.

e. Stationarity Check - To get the optimal result we have to check whether the data is stationary or not. Stationary data means that the variance and autocorrelation do not change over time.

To check the stationarity, we have used Augmented Dickey-Fuller Test.

i. Augmented Dickey-Fuller Test: The Dickey-fuller test is a method to check the stationarity. It finds the root of the data to check whether the data passes the null hypothesis.

If the test statistic(p) is less than 0.05 the data is stationary. On the other hand if data is greater than 0.05 the data is non-stationary.

ii. Making Data Stationary: To make the data stationary there are many ways. However we follow square root procedure to make the data stationary. After this we drop the NaN values and proceed with the stationary data set. The figure 3.11 shows us the before and after of data stationarity.

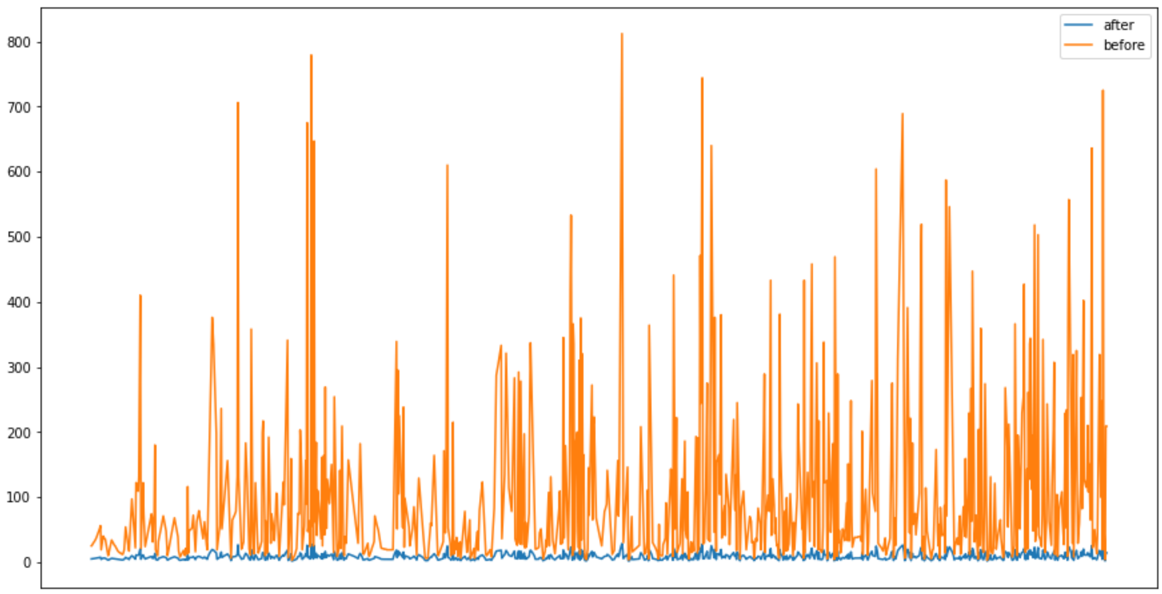


Figure 3.11: Stationary(Before VS After).

3.2.4 Training the dataset with ARIMA, OLS-SARIMA, Prophet, Neural Prophet, DNN, CNN-RNN-DNN

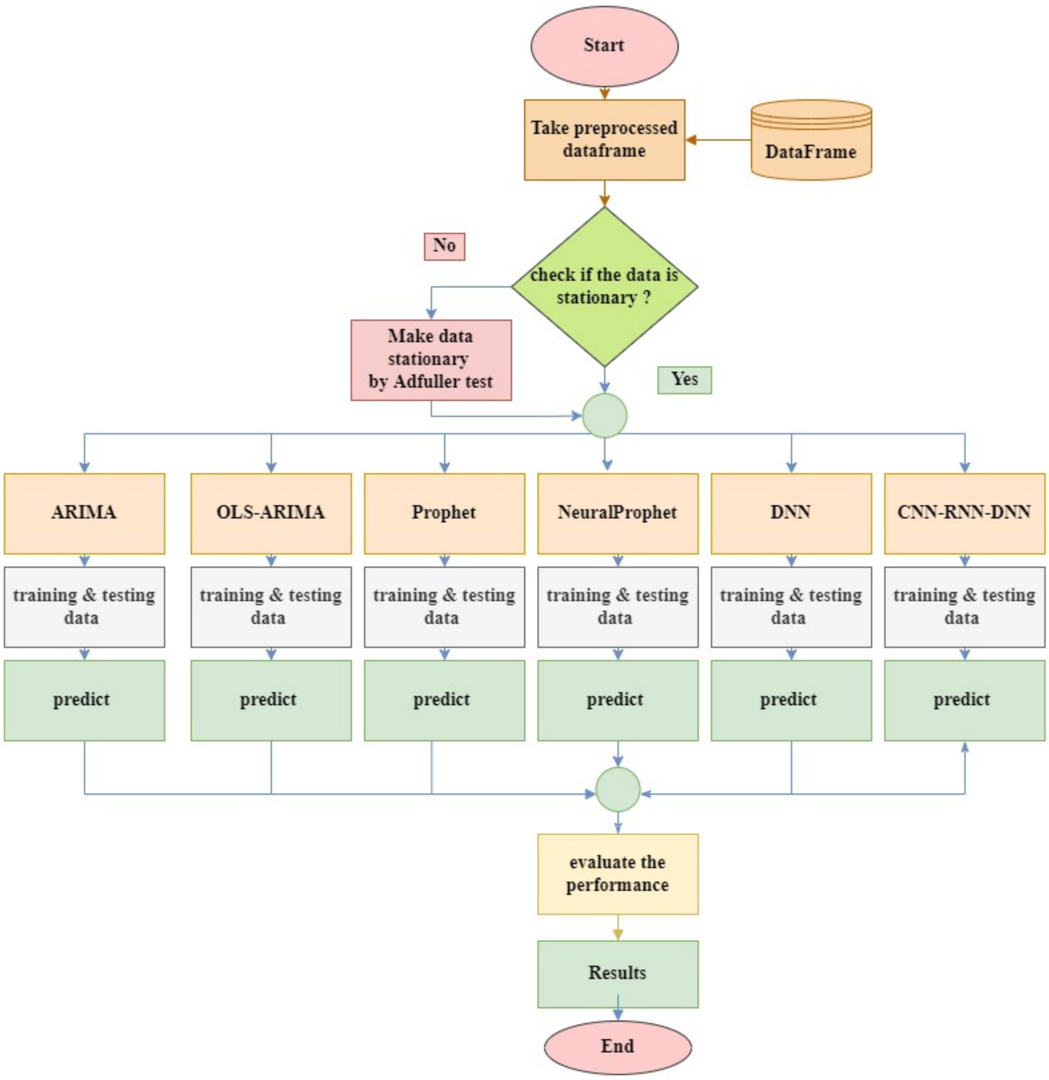


Figure 3.12: Work Plan

After pre-processing the data we are going to follow work plan(figure 3.12) to run our algorithms and training the data. And now we are going to implement the algorithm and predict sales for the next 24 months. To predict sales we are going to use machine learning algorithms like ARIMA, deep learning algorithms like RNN, GRU, and auto-regressive algorithm Prophet and RNN-CNN and RNN-Prophet model. We are going to get result for all the model mentioned and compare them to see which one of them gives a better prediction.

A. ARIMA

For the dataflow, we are going to
1. Visualize the Time Series Data
2. Make the time series data stationary

3. Plot the Correlation and Autocorrelation Charts
4. Construct the ARIMA Model based on the data
5. Use the model to make predictions

After pre-processing to stationarity check we use the “dickey fuller test” it gives us a “p” value which checks whether the value is ≤ 0.05. If it is then it rejects the null hypothesis and the data is stationary. For the ARIMA model, we plot auto-correlation and partial auto-correlation as they have a relation with the ARIMA model.

ARIMA is imported from ”statsmodels” library. Auto regressive Component AR(p) has been set to 1. We take into consideration the previous timestamp modified by a multiplier and then add white noise with the p parameter set to 1. Moving Average MA(q) determined the number of lagged forecasting errors in the prediction. The parameter ”d” depends on heavily stationarity parameter p and we set the value d=0 because of datas’ stationarity.

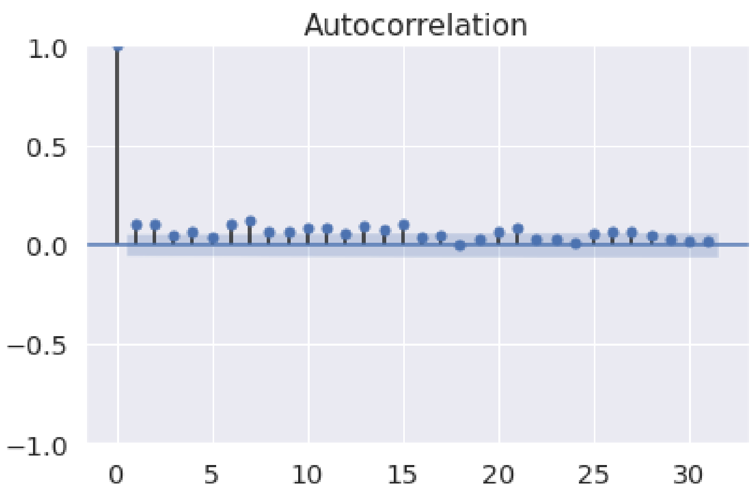


Figure 3.13: Auto-correlation plot of Sales

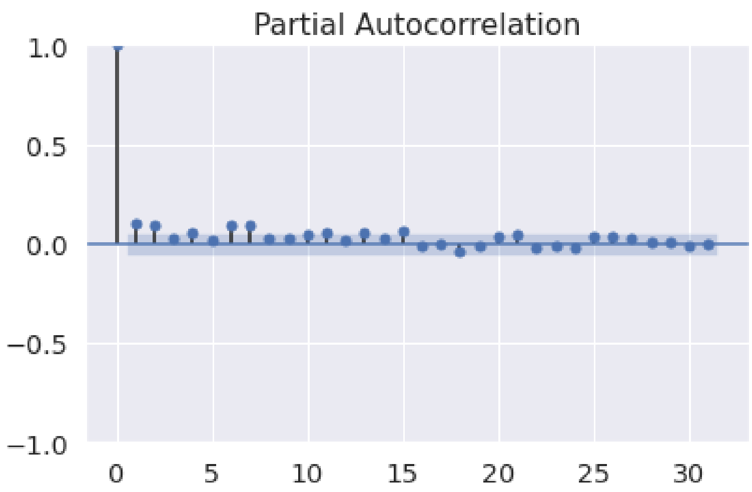


Figure 3.14: Partial Auto-correlation Plot

The PACF is arguably the most accurate method for identifying an AR model ever invented. After the model's order, the theoretical PACF "shuts down." The phrase "shuts down" implies that partial autocorrelations are mathematically equal to zero beyond a certain degree. In other words, the proportion of non-partial autocorrelations generates a sequence of the AR model. The "order of the model" refers to the most severe delay of x that is used as a predictor.

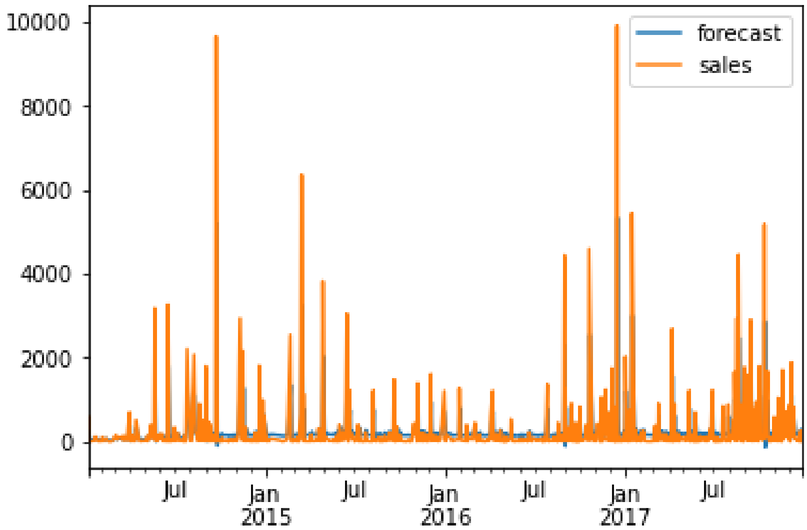


Figure 3.15: Algorithm Plot VS Dataset Plot

The ACF is more commonly used to distinguish MA models than the PACF. The hypothetical PACF for an MA model does not terminate, but rather fades toward zero in some fashion. The ACF contains a more distinct pattern for an MA model. Only the lags included in the model p,d,q p AR prototype lags d differencing q MA lags will have autocorrelations in the ACF that are greater than zero.

From the data, we fit an ARIMA model and we get Figure 3.13. For training data we split the dataset maintaining 80/20 rules and split 1157 days to train our model and give forecast on next 300 days.

B. OLS-SARIMA

OLS-SARIMA is a hybrid time series forecasting approach that uses univariate data with trends and seasonality. Seasonal differencing is a key player while making a prediction in this case. We'll start with stationary seasonal models. The seasonality's period "s" are known.

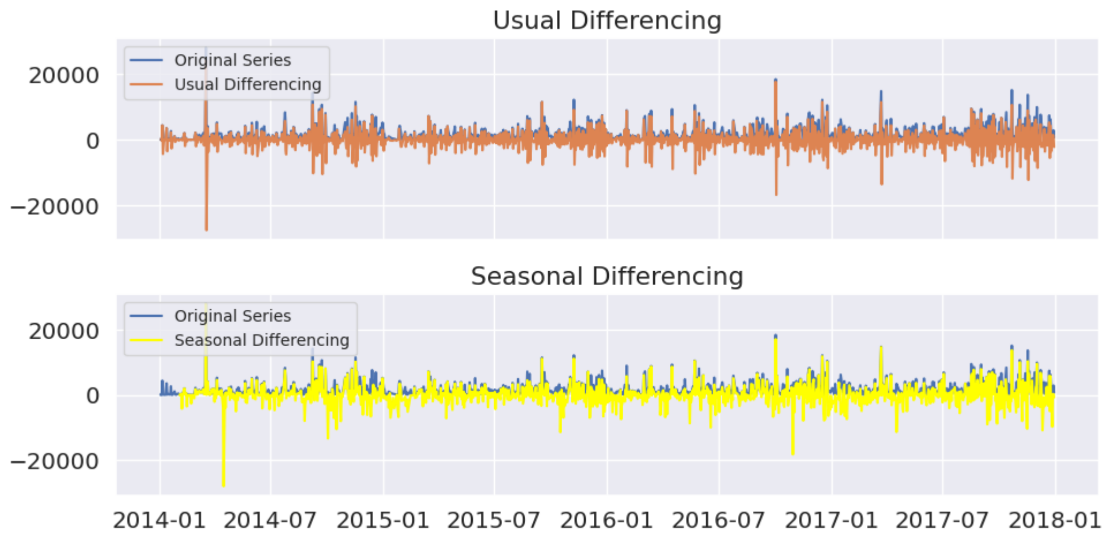


Figure 3.16: Seasonal Differencing VS Usual Differencing.

There are three processes involved in using OLS-SARIMA:

1. Establish the model. To support exogenous variables we implement SARIMAX model instead of SARIMA. To the "order" and "seasonal order" inputs, the trend and seasonal hyper-parameters are supplied as 3 and 4 element tuples, respectively. To get best parameters value of Order and Seasonal Order we are using auto_arima which will give us best fit parameter of SARIMAX. OLS also have to be fit to get prediction with Order and Seasonal Order parameters. After this we made a hybrid model called "Stacked" where both the model merged to get a better prediction.

2. Comply with the model's specifications. The implemented model will fit the dataset and it will return instance of the SARIMAX Results class which contains the details such as data and coefficients of the fit. Before predicting data we need to plot auto-correlation as well as partial auto-correlation charts.
3. Use the fit model to make a prediction. We partitioned the dataset into 1157 days for training and forecasting the next 300 days using 80/20 principles.

C. Prophet

i. Generating a Forecast: -

1. Fbprophet library is used by Prophet. We construct a Prophet object and then use the fit(Prophet.fit) method to generate model and predict(Prophet.make_future_dataframe) methods.

2. Prophet always takes a data frame with two columns: df and y. The df (date stamp) column should have a Pandas-friendly format, such as YYYY-MM-DD for dates and YYYY-MM-DD HH:MM:SS for time stamps. The y column, which represents the measurement we want to forecast, must be numeric.

3. We'll utilize the first 24 months' target values as training data and anticipate the prediction for the next 24 months.

ii. Plotting the Forecast : -

1. By invoking the Prophet.plot Prophet.plot_components methods and passing it the forecast data frame, we can plot the forecast trends and its components as seen in figure.

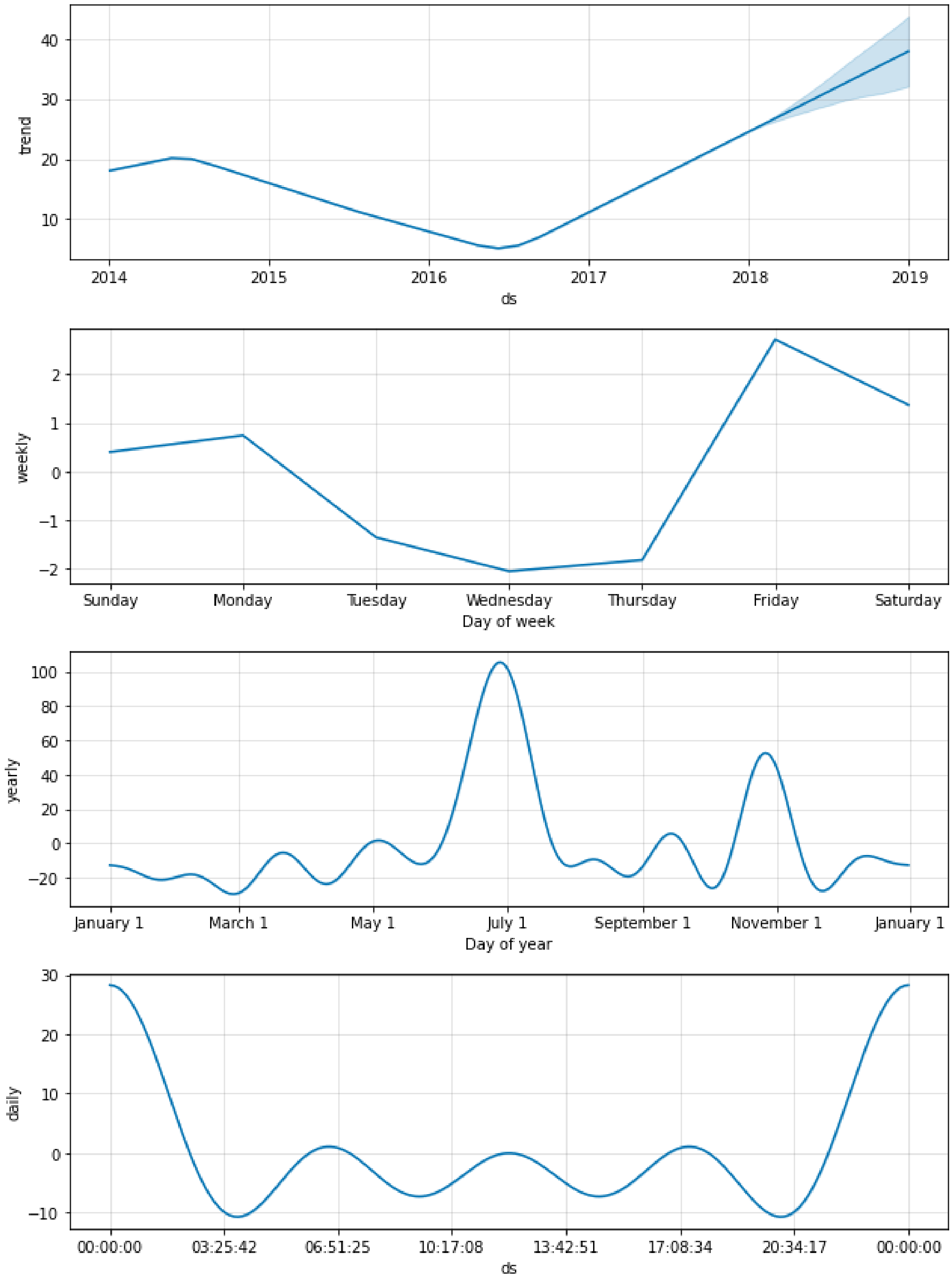


Figure 3.17: Trends for binder sub-category

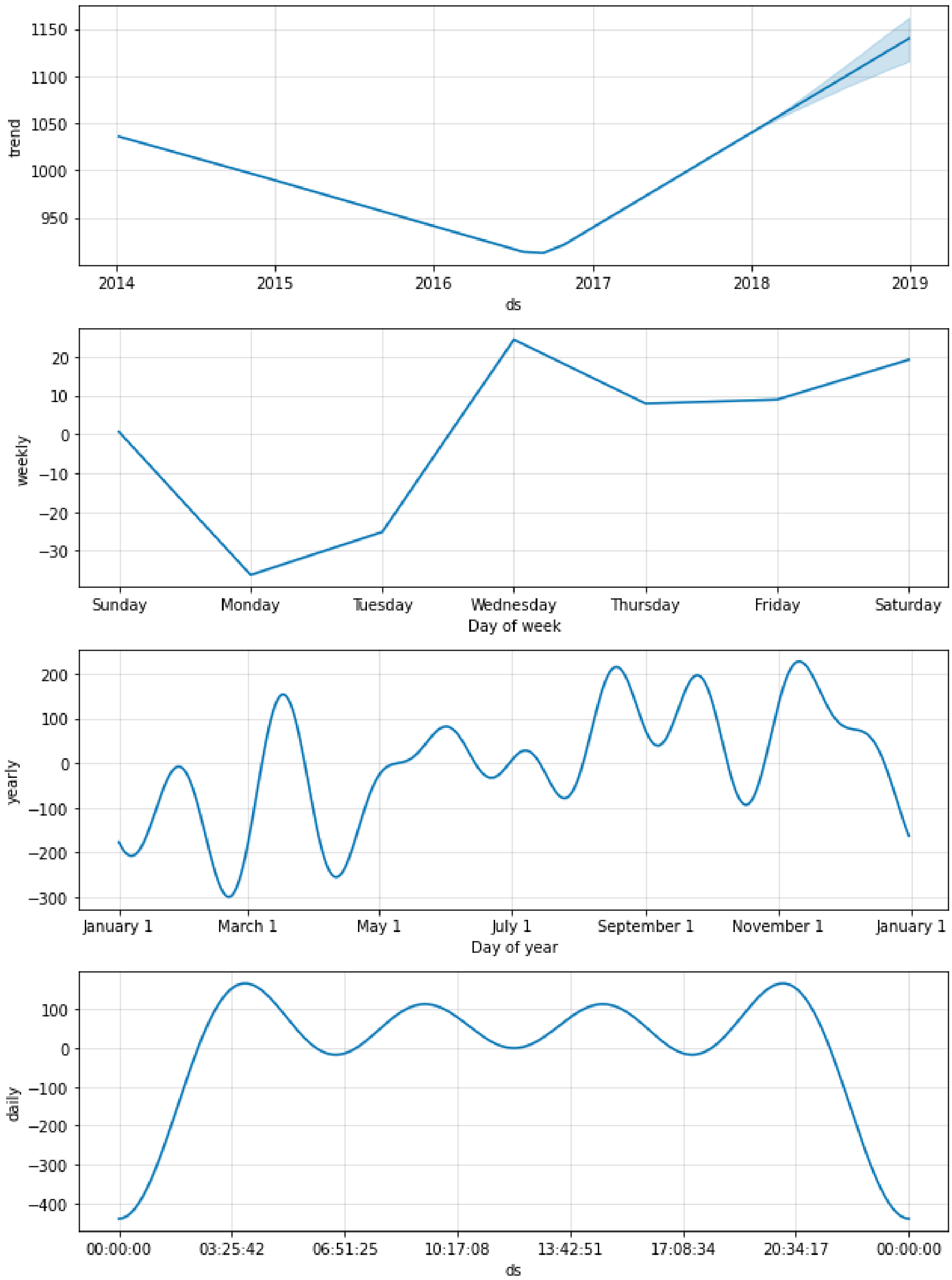


Figure 3.18: Trends for furnishings sub-category

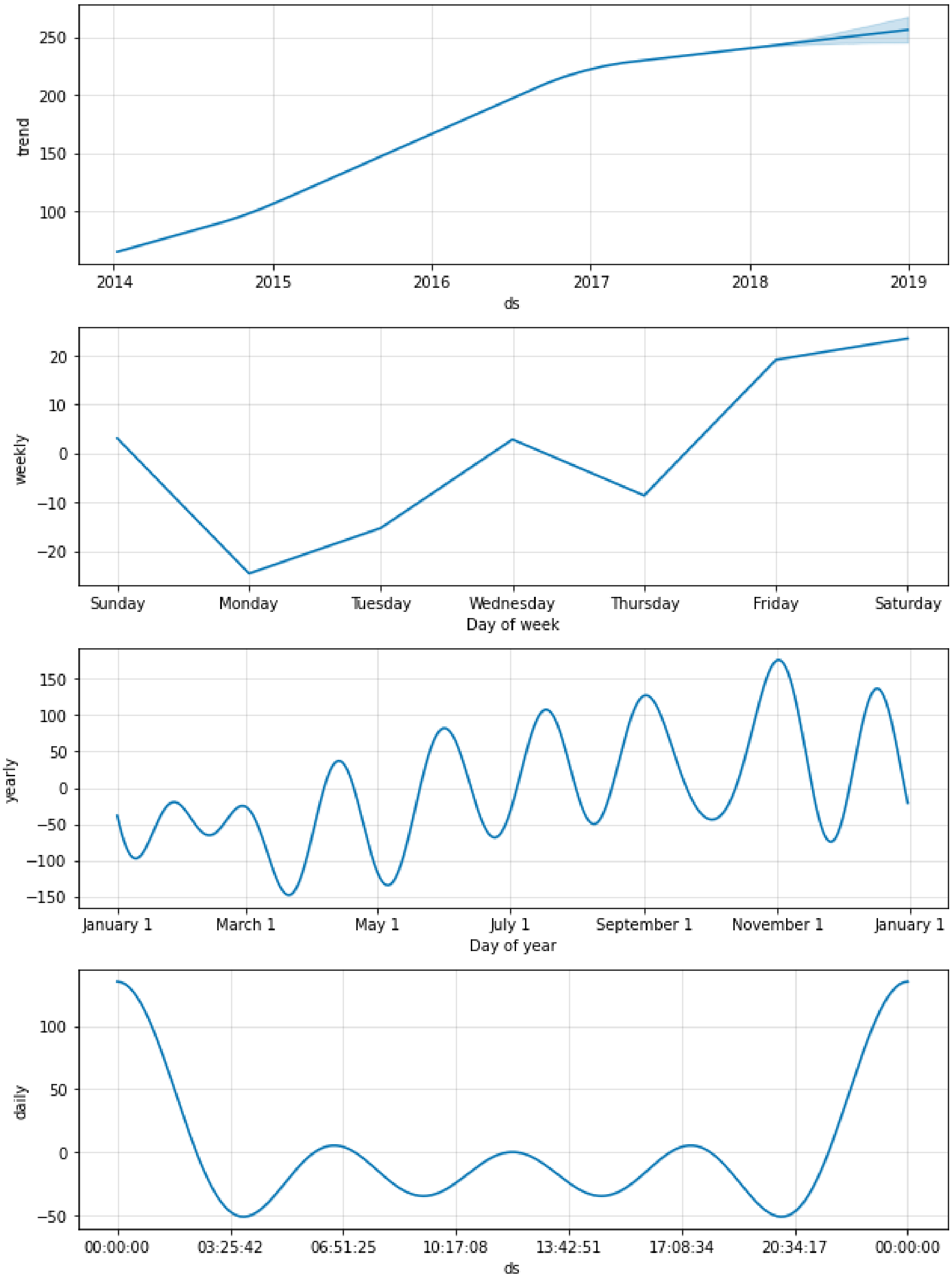


Figure 3.19: Trends for paper sub-category

2. The components plot is a collection of charts that represent different time series components (trend, seasoanilities) and external impacts.

iii. Forecast Component Plot: -

The date values (ds) for both past and future dates are represented on the X-axis.
The targeted values (y, yhat) for both historical and future dates are represented on the Y-axis.
The black dotted dots in the graph reflect past training data points.
The projections for the past and future are shown by the blue line.
There is also a light blue area that depicts the uncertainty bands.

Graph 1: all-time trend value (history and future).
Graph 2: daily seasonality a daily profile based on the training data for each day.
Graph 3: weekly seasonality a weekly profile based on the training data for each day of the week.
Graph 3:yearly seasonality a yearly profile based on the training data for each month of the year.

We use cross-validation to evaluate prediction performance across a 360-day horizon, starting with 1157 days of training data in the first cutoff and making predictions every 300 days after that.

iv. Uncertainty in the trend: - The possibility of future trend alterations is the main source of uncertainty in the prediction. We anticipate similar trend shifts in the future as in the past. Allowing more rate flexibility by raising "changepoint_prior_scale" increases forecast uncertainty, which is one characteristic of this method of evaluating uncertainty. Because we predict greater rate changes in the future if we model more rate changes in the past, the uncertainty intervals are a valuable sign of overfitting. The "interval_width" argument controls the width of the uncertainty intervals (by default, 80 percent).

D. Neural Prophet

NeuralProphet is a modular framework made up of interpretable, scalable, and independently adjustable components. Mini-batch stochastic gradient descent is used to train all modules together (SGD). Any model component that can be trained using SGD may be incorporated as a module, making it simple to include cutting-edge forecasting approaches into the framework.

The procedures are as follows:

1. Reading and Pre-processing Sales Meta Data After pre-processing the data when we pass our data to neuralprophet, it only expects two columns: a column called 'ds' that stands for dates, and another column called 'y' that represents the value that we're trying to predict, so we'll set our date column to equal ds and our sales column to y, and then remove all the other columns.

2. Modeling a NeuralProphet
We start with an untrained instance of our neuralprophet model and train it with the fit() function. The variable'm' will be used to hold the final model.
The fit technique requires three parameters: the data frame, the data frequency (days), and the epochs (number of passes that the algorithm has to complete during training).

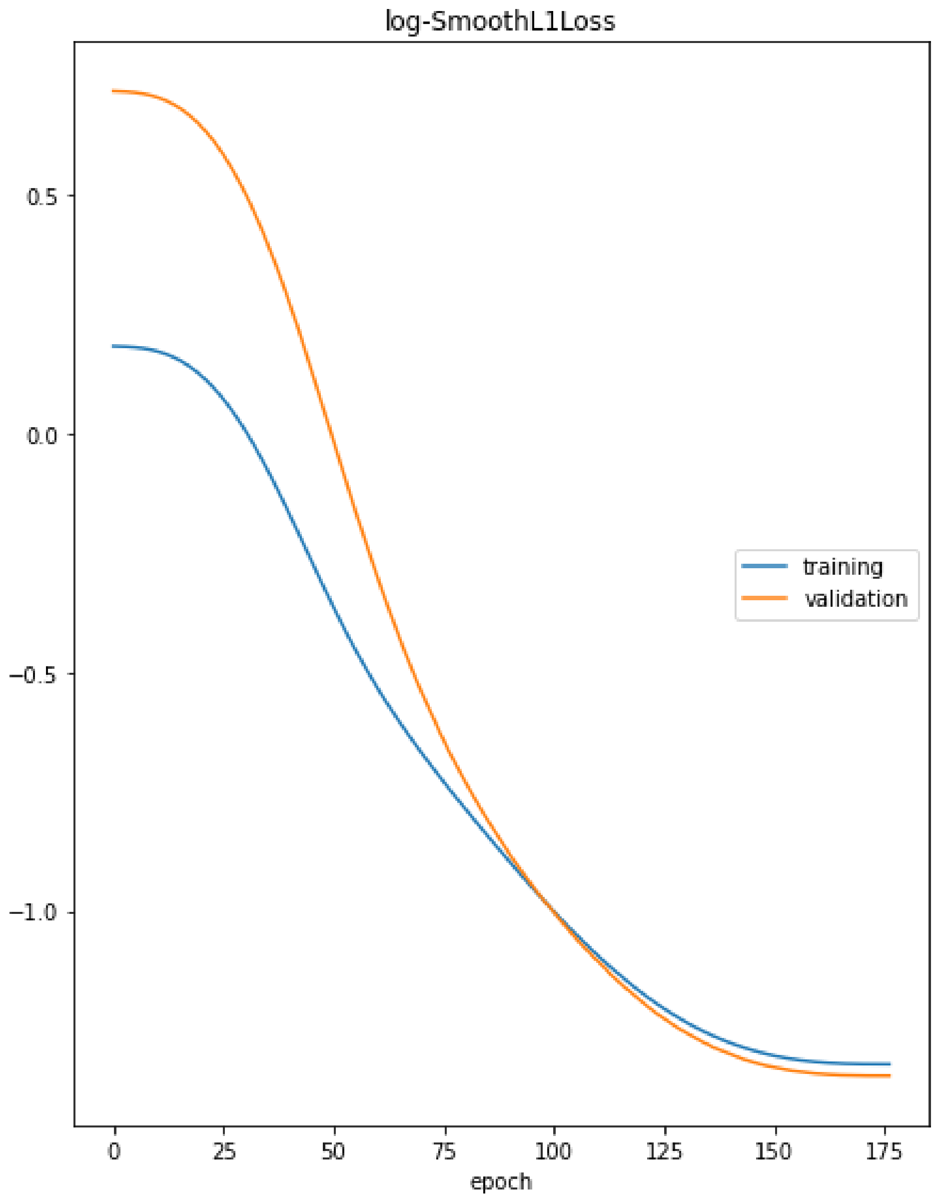


Figure 3.20: Training VS Validation of Neural Prophet

3. Forecasting using NeuralProphet
After tuning the model with weekly-seasonality = 8, daily seasonality = 7 with a learning rate = .0009, we tried both overfitting and under fitting. We got the best result with epoch ranging in 177-250. We use cross-validation to evaluate prediction performance across a 365 days, starting with 1157 days of training data in the first cutoff and making predictions every 300 days after that.

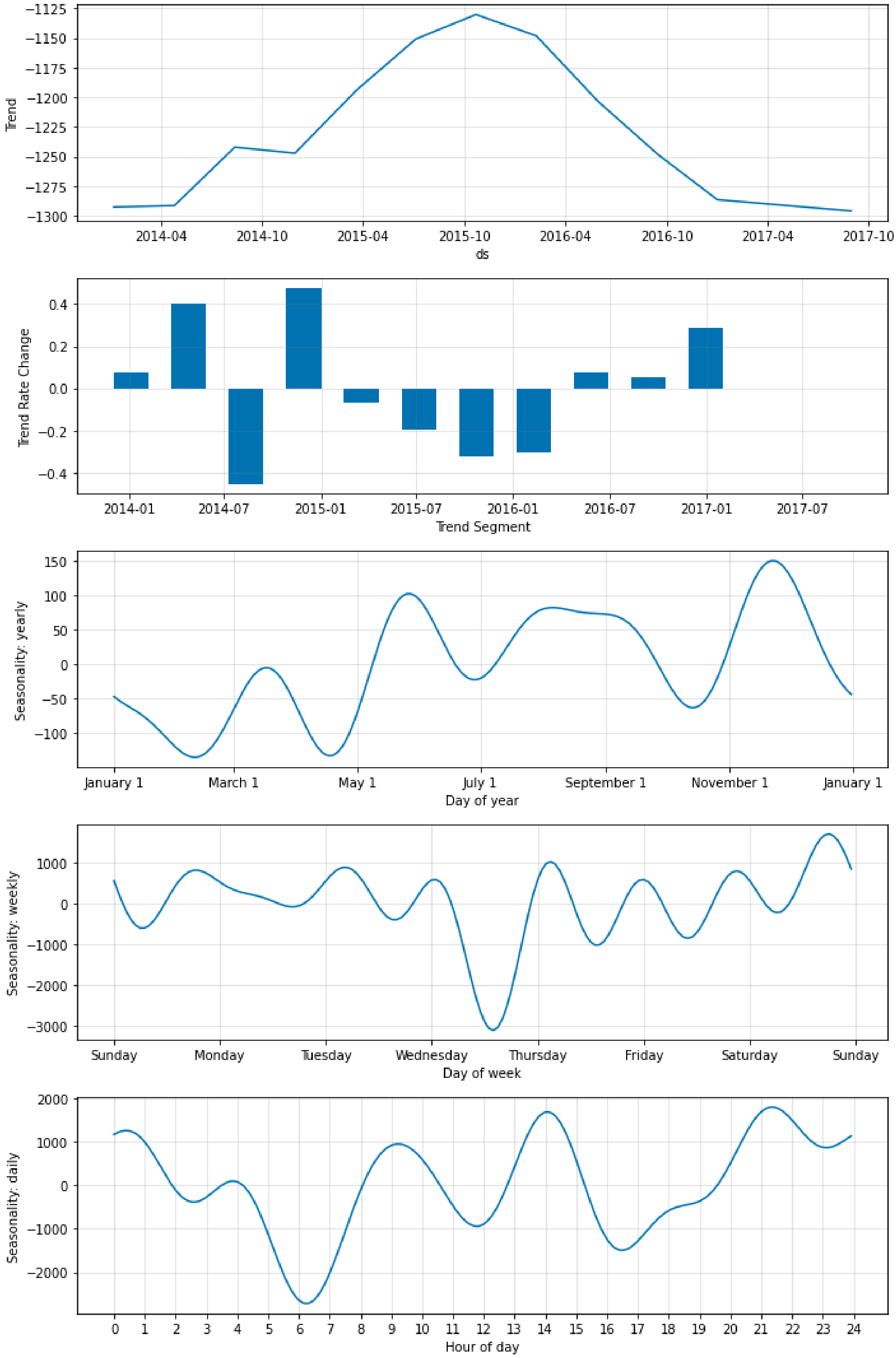


Figure 3.21: Trends for Binder Sub-category

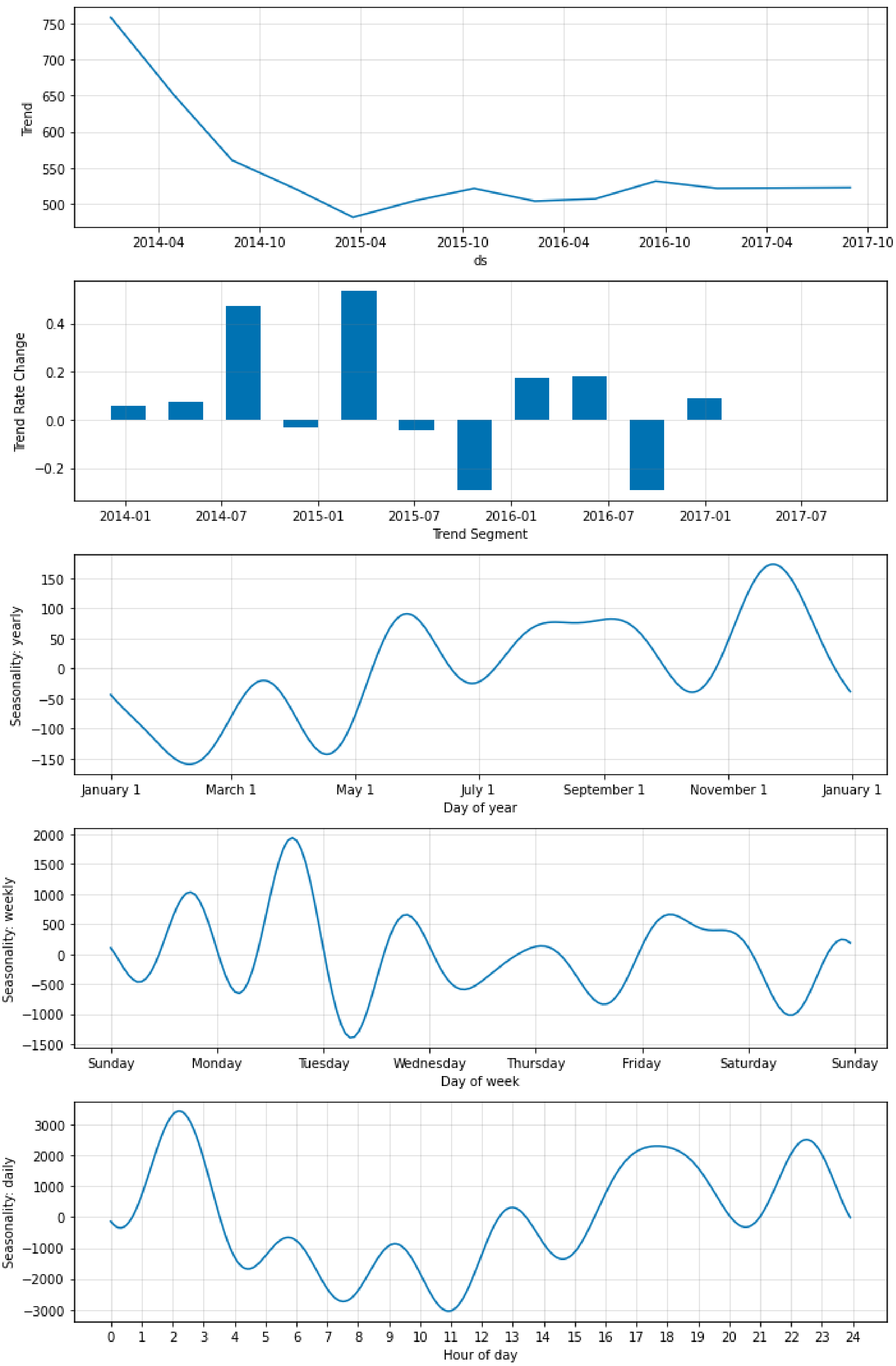


Figure 3.22: Trends for Furnishing Sub-category

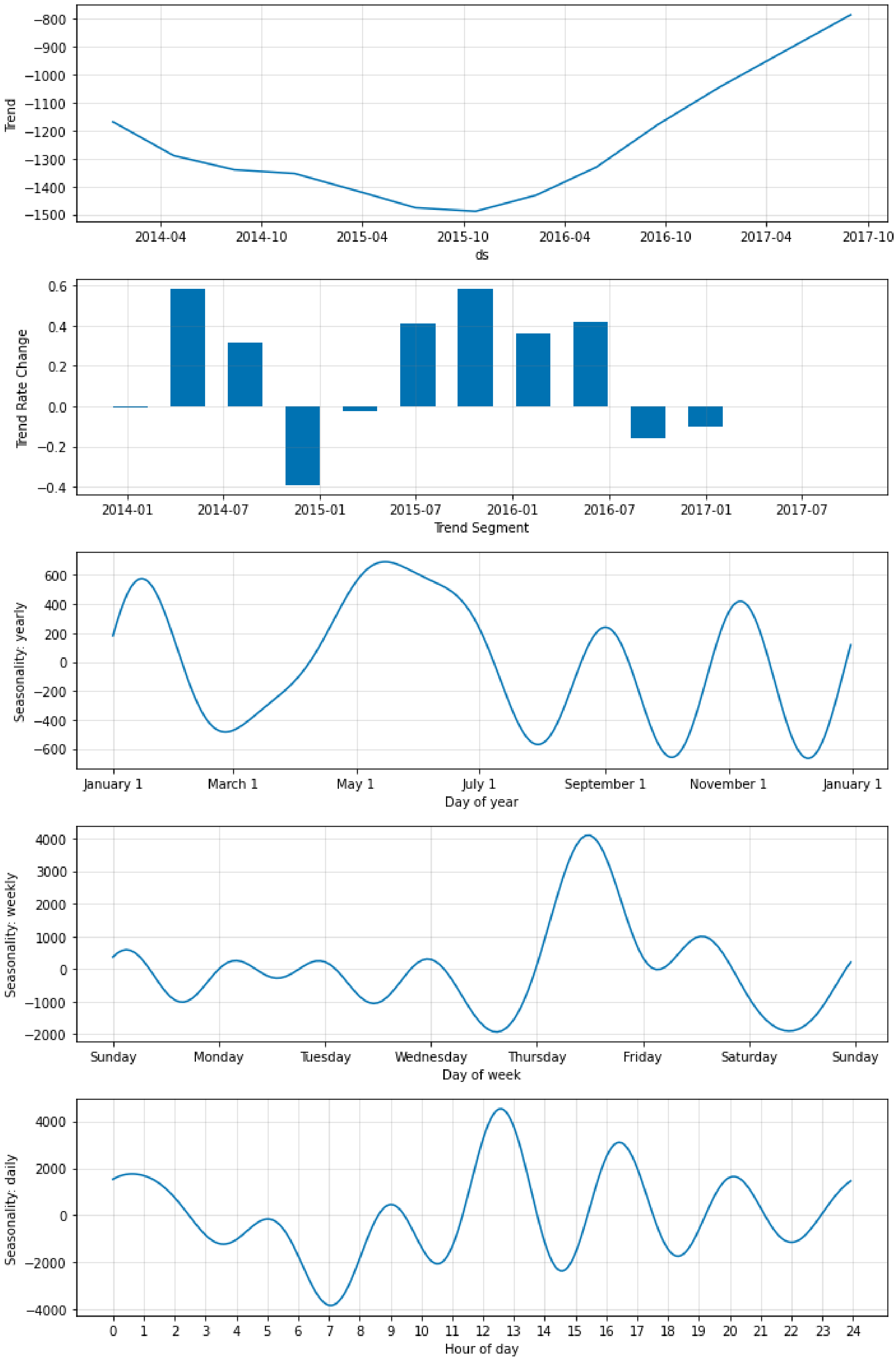


Figure 3.23: Trends for Paper Sub-category(np)

E. DNN

i. Model Preparation: -

Firstly we have loaded our sales data as a series and generated a time array for forecasting. We have split our dataset by following 80/20 rules. We use windowed_dataset function to set our dataset in a way that it will be divided in sub_section to increase the dimension of the dataset.

ii. Construct Multiple Dense Layer: -

In this section we are using ReLu activation function in the neurons of the hidden dense layer. For our dataset we have used 6 dense layer to train the model. AS we can see a set of hyper parameters are used as input shape for the dense layer. Our window size is 365 days.

iii. Tuning the learning Rate : -

The learning rate is one of the first and most important parameters of a model. It controls how the model learns. Our Model is trained by executing gradient descent through a network of nodes on our dataset. We have used learning rate scheduler to dynamically configure the learning rate in each epoch. Then we have used this lr_scheduler function as a callback to fit our model. We have select a spot for the learning rate past the edge of the curve we get noise. We did this to enhance the ability of the learner to more easily escape saddle points on the surface: places where the gradients are relatively flat in all directions.

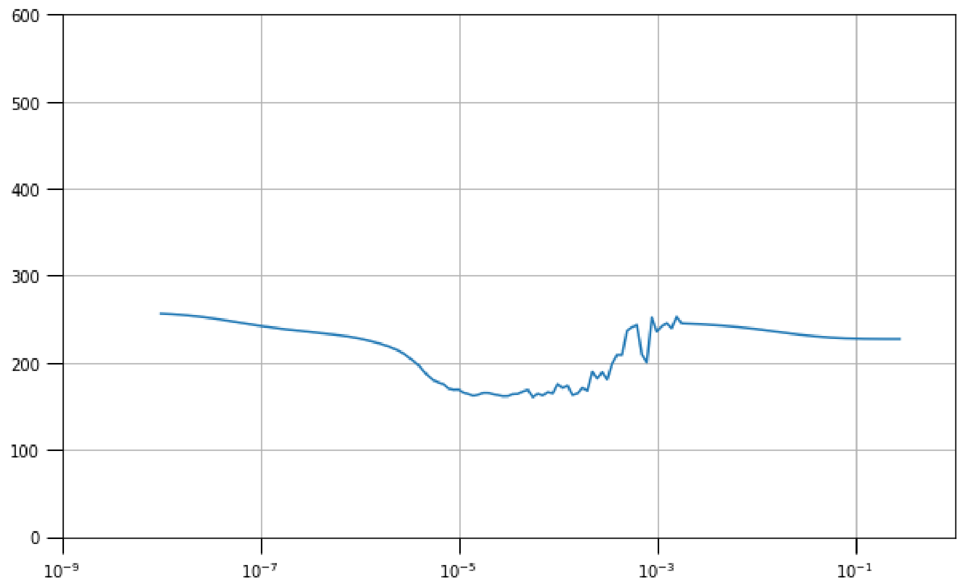


Figure 3.24: Tuning The Learning Rate

iv. Training The Model : -

After tuning the learning rate we train the model using the appropriate learning rate.

F. CNN-RNN-DNN : -

CNN layers for feature extraction on input data are paired with RNNs and DNN to allow sequence prediction in the CNN-RNN-DNN architecture. Our CNN-RNN-DNN model can be defined by adding CNN layers on the front end followed by RNN layers with multiple dense layers on the output. This architecture is defining three sub models: The CNN model for feature extraction, RNN model for interpreting the features across time steps and DNN model for changing the dimension of the output.

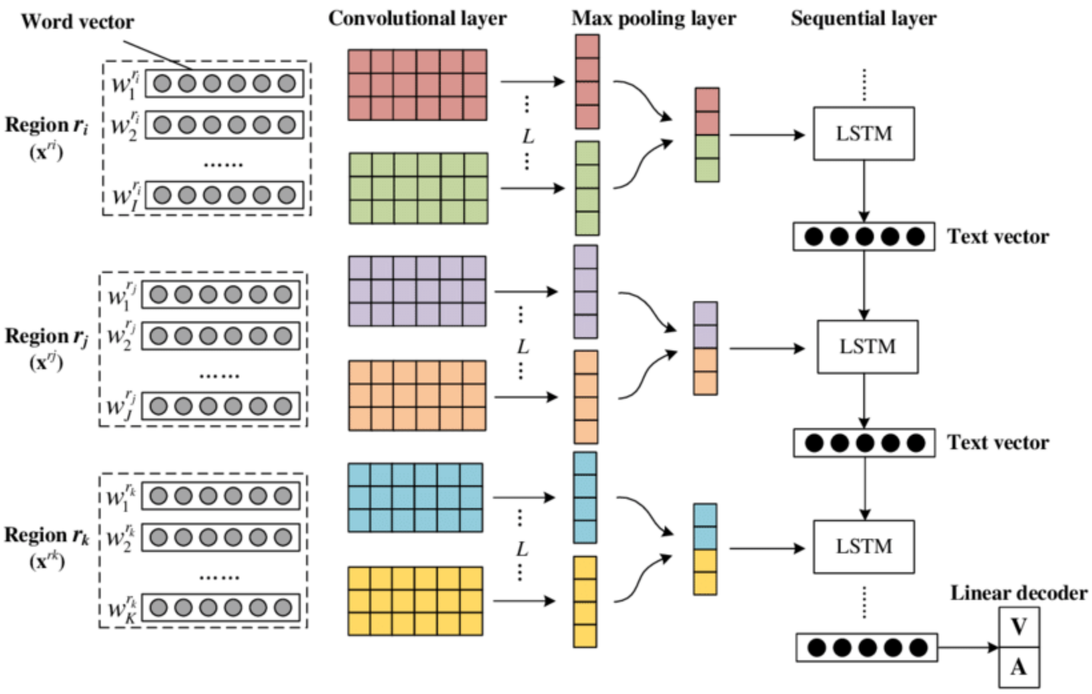


Figure 3.25: CNN-LSTM Architecture

i. Model Preparation: -

Firstly we have loaded our sales data as a series and generated a time array for forecasting. We have split our dataset by following 80/20 rules. We use windowed_dataset function to set our dataset in a way that it will be divided in sub_section to increase the dimension of the dataset.

ii. Construct CNN-RNN-DNN Layer: -

In this section we are using ReLu activation function in the neurons of the hidden dense layer. We use Conv1D layer from keras for CNN. We use 64 filters in Conv1D. Next, we use 2 RNN layer of 64 size as well as 3 CNN layer of same size. Furthermore, we use three dense layers to shape our output.

iii. Tuning the learning Rate : -

The learning rate is one of the first and most important parameters of a model. It controls how the model learns. Our Model is trained by executing gradient descent through a network of nodes on our dataset. We have used learning rate scheduler to dynamically configure the learning rate in each epoch. Then we have used this lr_scheduler function as a callback to fit our model. We have select a spot for the learning rate past the edge of the curve we get noise. We did this to enhance the ability of the learner to more easily escape saddle points on the surface: places where the gradients are relatively flat in all directions.

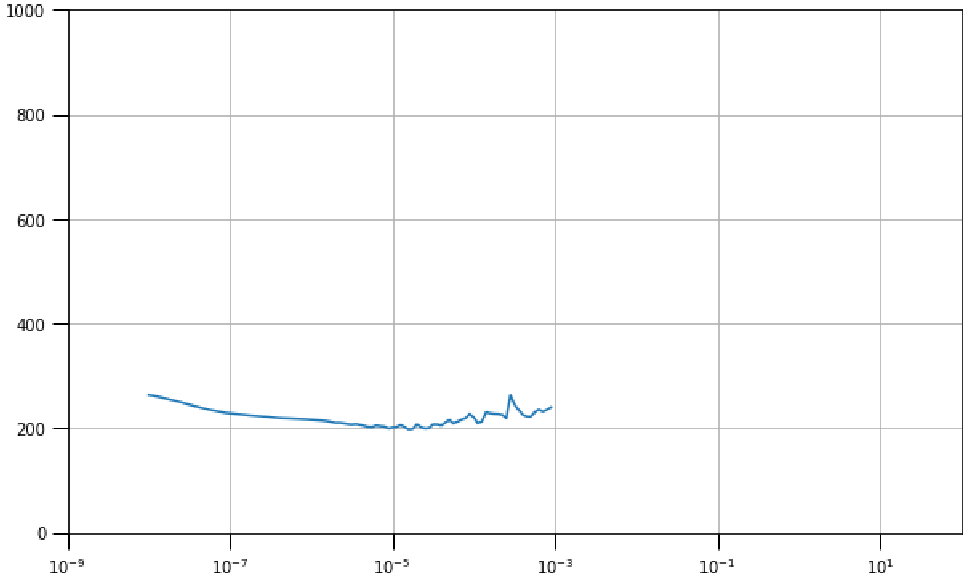


Figure 3.26: Tuning The Learning Rate

iv. Training The Model : -

After tuning the learning rate we train the model using the appropriate learning rate.

Chapter 4

Evaluate the Model Performances

4.1 Model Performance

Throughout the research we tried to find solutions of existing forecasting method and their errors.To solve this we break down the categories into sub categories to get a better result of supply chain forecasting. As ours is the recommendation system, we recommend a daily forecast of each most selling sub-category to the vendors against the existing data. To do so we are running some machine learning, neural and hybrid models and suggest the model with the best accuracy.

4.1.1 ARIMA

ARIMA uses forecast() functions to forecast future time steps against the existing data by using ARIMAResults object to make predictions. To get the accurate results we use forecast() function after training 1157 days data and predict the next 300 days forecasting. We choose 3 categories - Binders, Furnishing and papers as our main sub-categories and with each sub-category has 1157 days data to train and 300 days data to test.
The figures show time vs sales forecast of each sub-categories. Here in the graph the blue line shows the result of training data and orange line shows the future prediction of next 1 year.

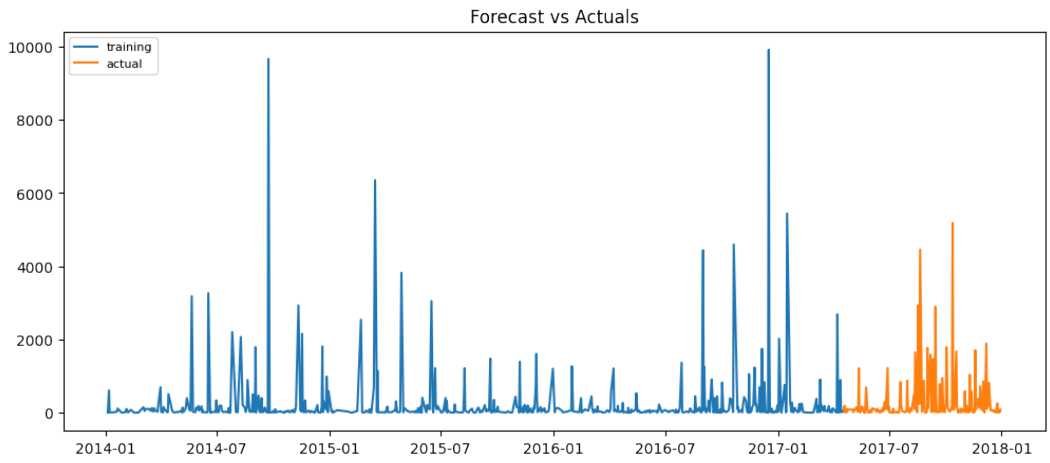


Figure 4.1: ARIMA forecasting against existing data for ”Binders” sub-category

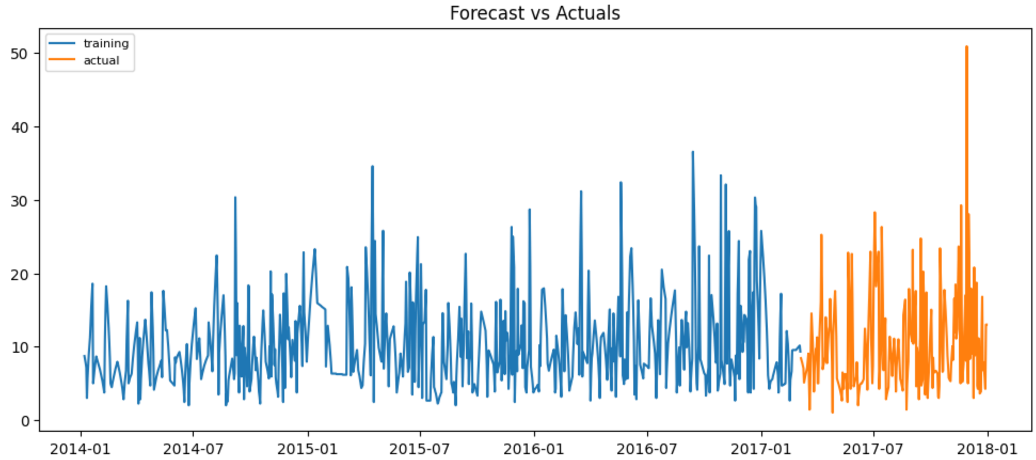


Figure 4.2: ARIMA forecasting against existing data for "Furnishings" sub-category

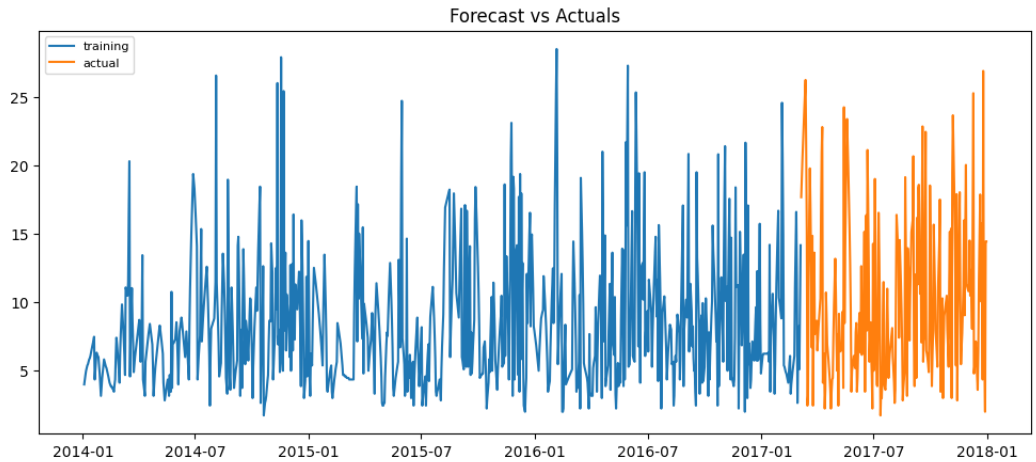


Figure 4.3: ARIMA forecasting against existing data for "paper" sub-category

4.1.2 OLS-SARIMA

Forecasting future time steps against existing data is done by SARIMA utilizing the forecast() methods and the SARIMAResults object. In the hybrid OLS-SARIMA model we took the forecast result of SARIMA and use predict() function to get the future prediction against existing data. We can divide the training dataset into train and test sets, fit the model with the train set, then create predictions for each element on the test set using the train set.After training 1157 days of data and forecasting the following 300 days, we utilize the forecast() method to acquire correct results. We chose three primary sub-categories: binders, furnishings, and papers, with each sub-category having 1157 days of data to train and 300 days of data to test.
Here the figures show time vs sales forecast of each sub-categories. It shows the sales data of 15 days time period. The red line shows the t+1 predicted data whereas the blue line shows the actual existing data.

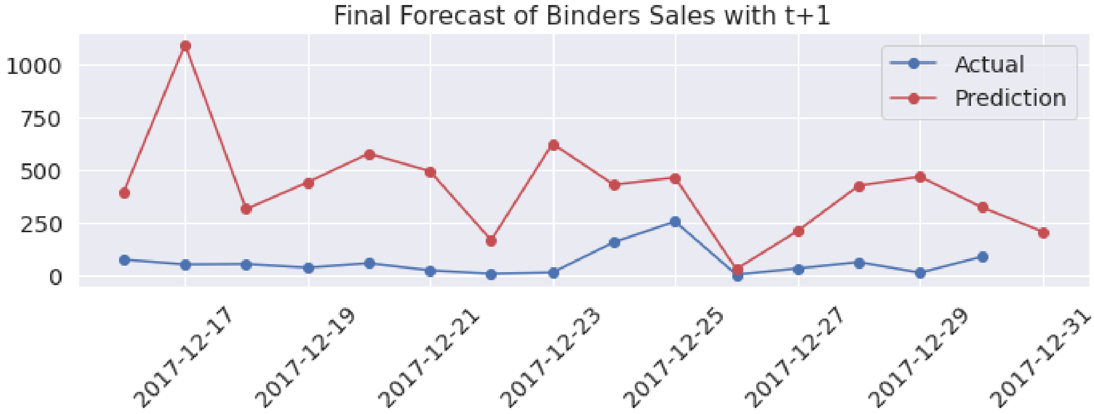


Figure 4.4: SARIMA forecasting against existing data for "Binders" sub-category

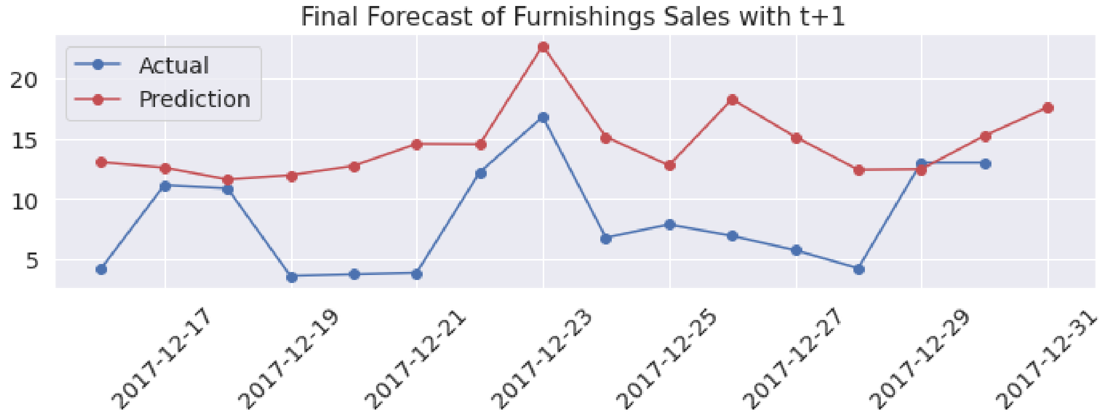


Figure 4.5: SARIMA forecasting against existing data for "Furnishings" sub-category

Figure 4.6: SARIMA forecasting against existing data for "Papers" sub-category

4.1.3 Prophet

Prophet has the ability to assess forecast inaccuracy using past data using time series cross validation. This is accomplished by identifying cutoff points in the history and fitting the model solely utilizing data up to those cutoff points for each of them. The projected and actual numbers may then be compared.

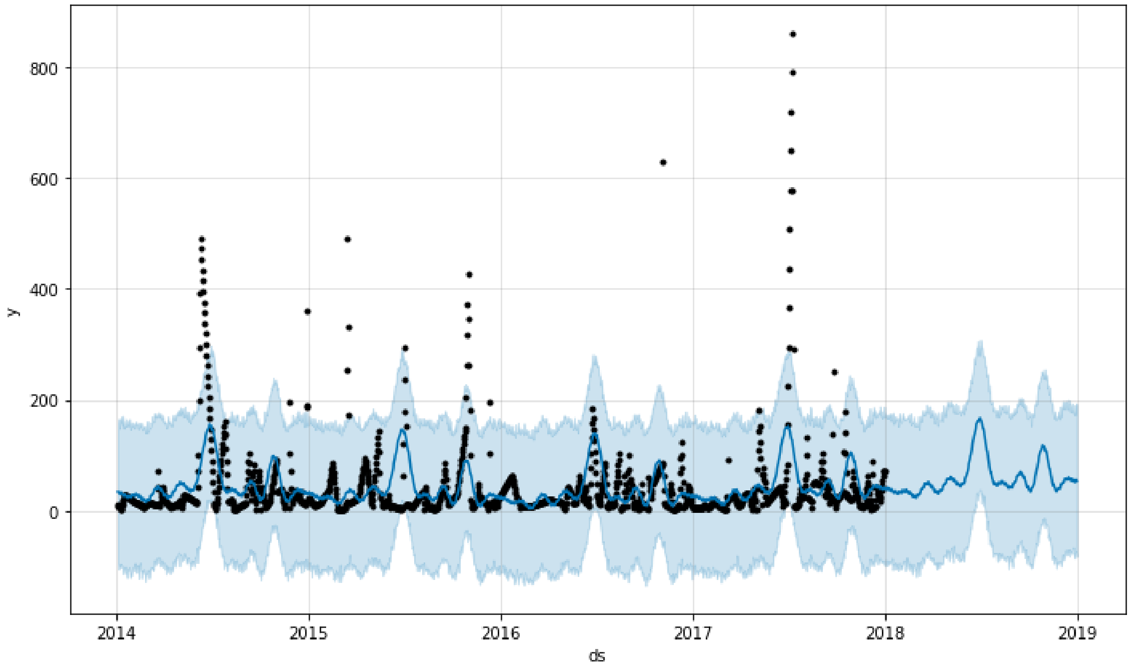


Figure 4.7: Prophet forecasting against existing data for "Binders" sub-category

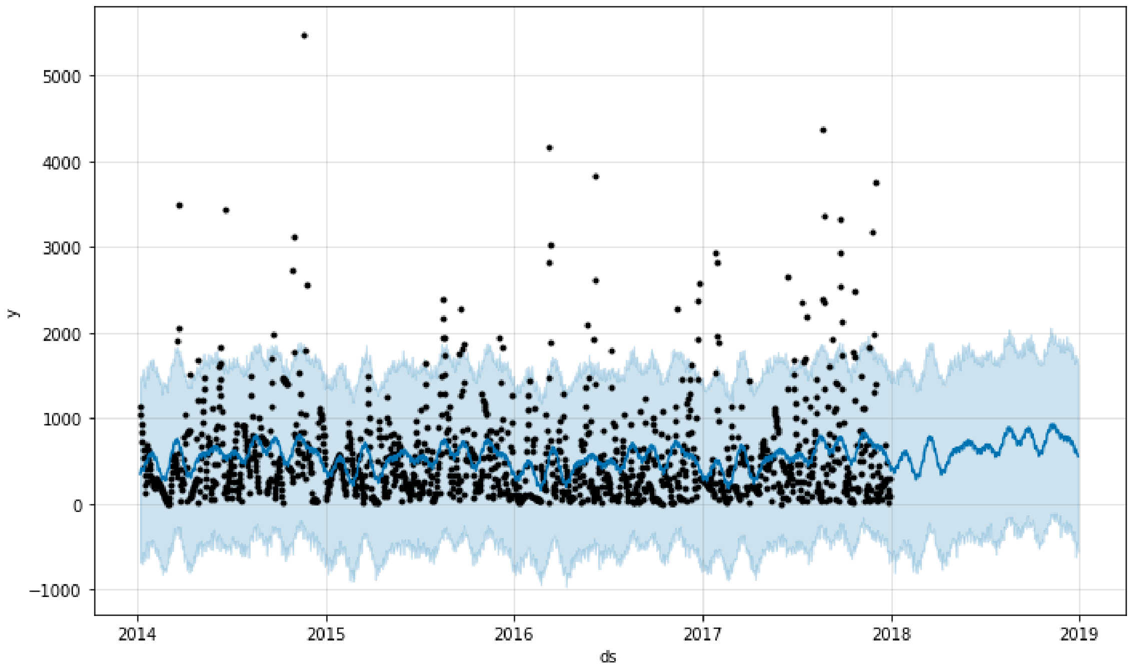


Figure 4.8: Prophet forecasting against existing data for "Furnishings" sub-category

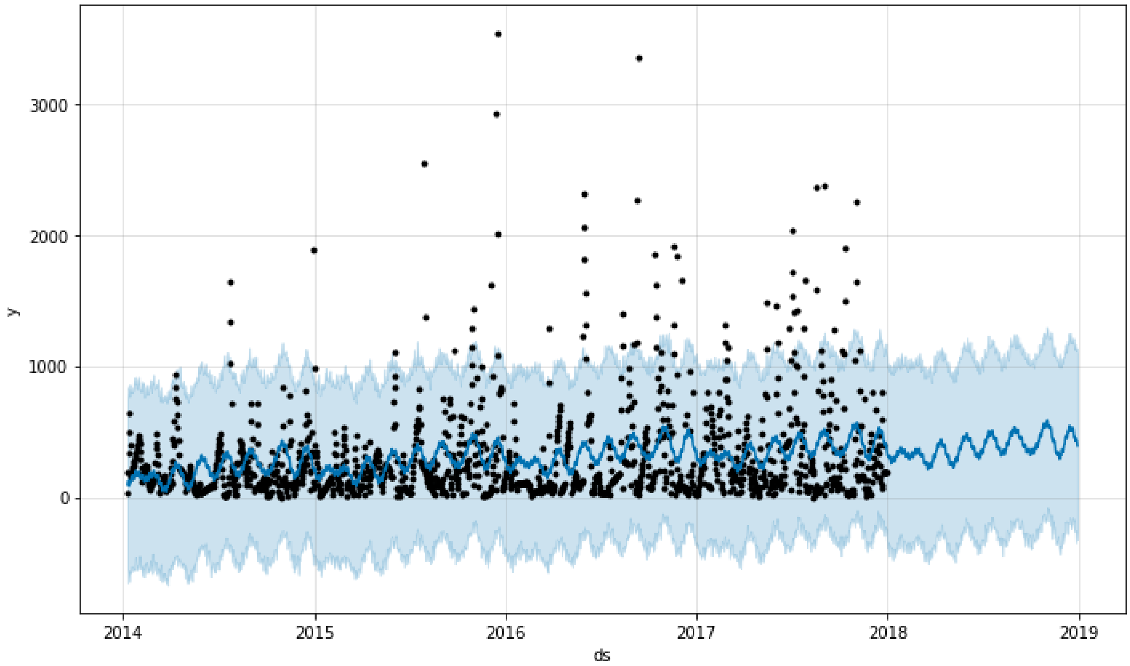


Figure 4.9: Prophet forecasting against existing data for "Papers" sub-category

Using the cross validation function, we perform this cross validation technique automatically for a variety of historical cutoffs. We define the forecast horizon (horizon), as well as the size of the first training period (initial) and the cutoff date spacing (period).
We can divide the training dataset into train and test sets, fit the model with the train set, then create predictions for each element on the test set using the train set. After training 1157 days of data and forecasting the following 300 days, we utilize the forecast() method to acquire correct results. We chose three primary sub-categories: binders, furnishings, and papers, with each sub-category having 1157 days of data to train and 300 days of data to test. At each simulated forecast date and for each cutoff date, cross validation produces a data frame containing the real values y and the out-of-sample forecast values "yhat".

Here in the figures, the dark blue line shows the future prediction of 1 year whereas the black dotted line shows the forecast of 1157 days. The upper and lower bound light blue color area shows the uncertainty which can be created due to various reasons like promotional sales or decline in sales etc.

4.1.4 Neural Prophet

As part of the training process outputs, a NeuralProphet model self-validates and offers validation metrics. Split the data into training and testing sets using the model's split df() method if we installed the [live] version of NeuralProphet and want to keep track of metrics during the training process().

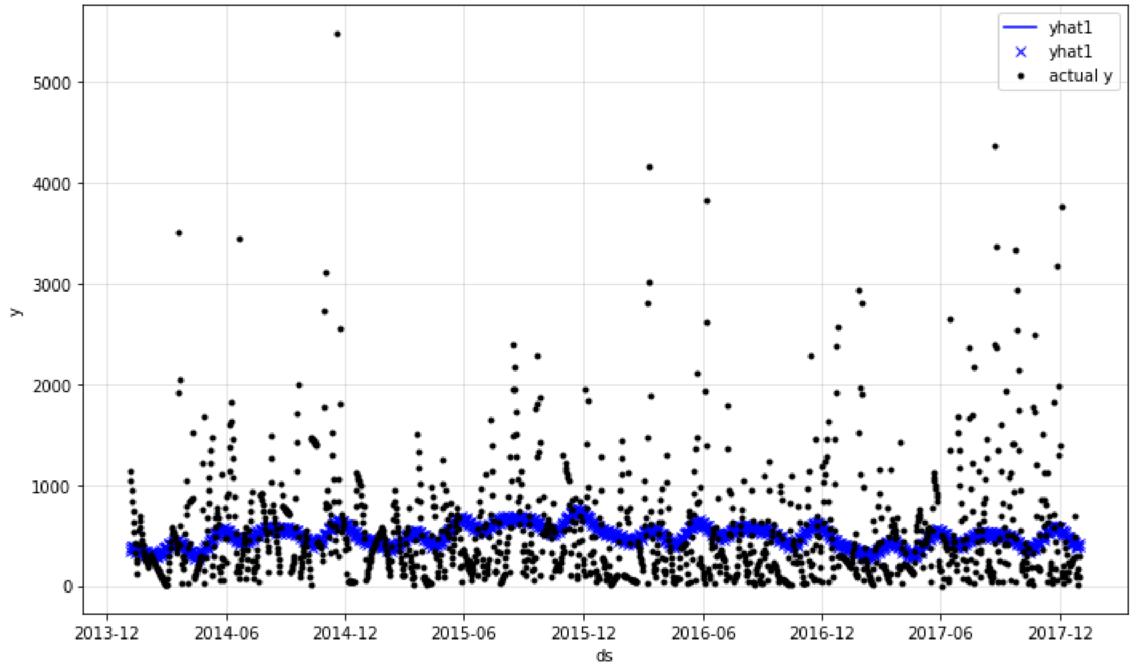


Figure 4.10: NeuralProphet forecasting against existing data for ”Binders” sub-category

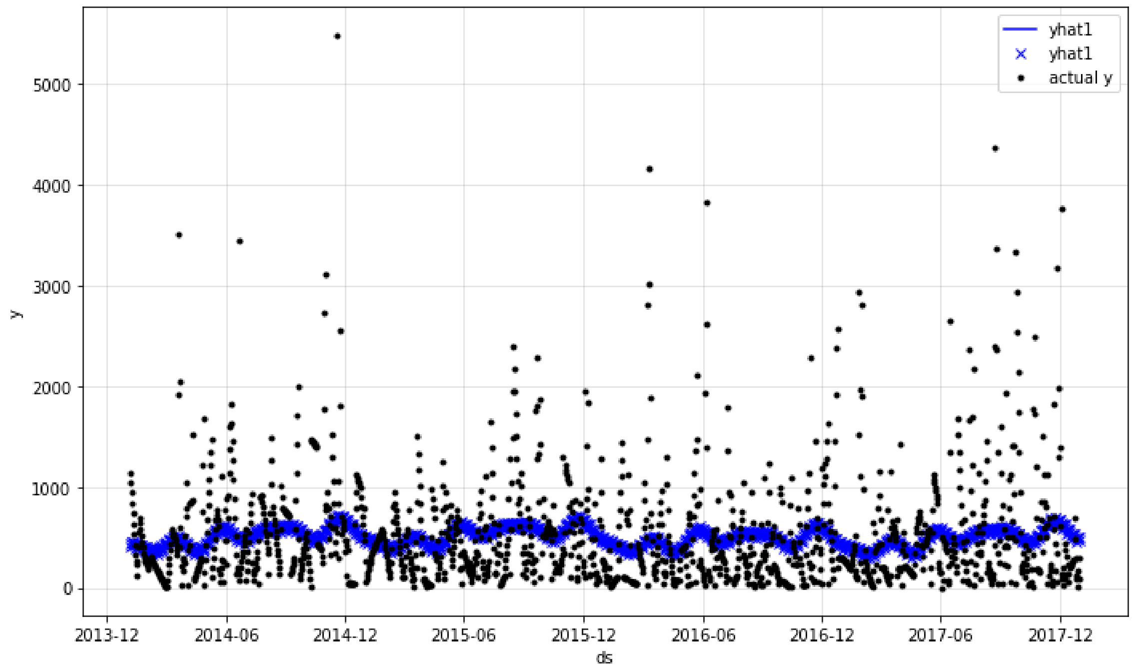


Figure 4.11: NeuralProphet forecasting against existing data for ”Furnishings” sub-category

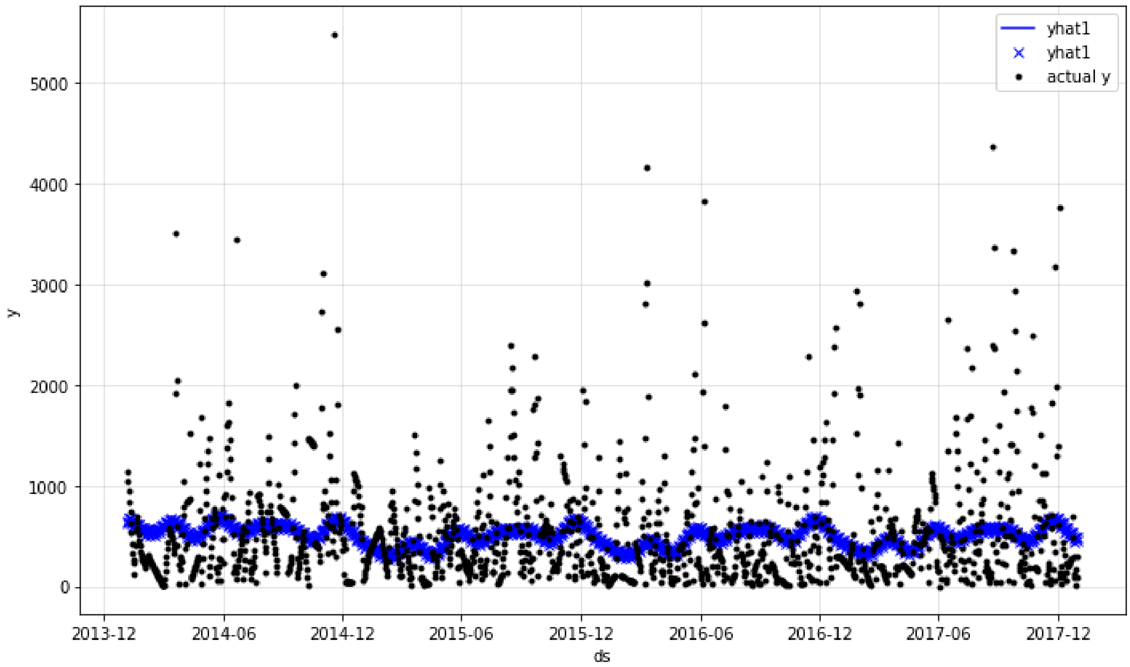


Figure 4.12: NeuralProphet forecasting against existing data for "Papers" sub-category

The make future forecast() function may be used to forecast future trends by giving data and the number of periods (days) to it. It look ahead a year. To execute the forecast, we will use the predict() function to forecast. The yhat1 column in the forecast dataframe represents the projections. We can divide the training dataset into train and test sets, fit the model with the train set, then create predictions for each element on the test set using the train set.After training 1157 days of data and forecasting the following 300 days, we utilize the forecast() method to acquire correct results.
The figures of each sub-categories represent actual sales prediction as well as uncertainty of the data. The black dot represent the actual forecast whee as the "x" shows the predicted result of the next 1 year data and the dark blue line shows the uncertainty of the trend.

4.1.5 DNN

In DNN we tune the hyper-parameter to build the model with create_model() function. We use this model to tune the learning rate with adjust_learning_rate() function. After finding the best learning rate we train our model and forecast to cross validate our dataset.

We split the training dataset into train and test sets, fit the model with the train set, and then use the train set to make predictions for each element on the test set. We use the forecast() technique to get proper results after training 1157 days of data and forecasting the next 300 days. Binders, furniture, and papers were chosen as the three principal sub-categories, each with 1157 days of data to train and 300 days of data to test.

Figure 4.13 - 4.15 shows the forecasting of three sub-categories - Binders, Furnishings, and Papers. Each graph shows time vs sales validation. The blue line defines the actual set and the orange line defines as forecasting set.

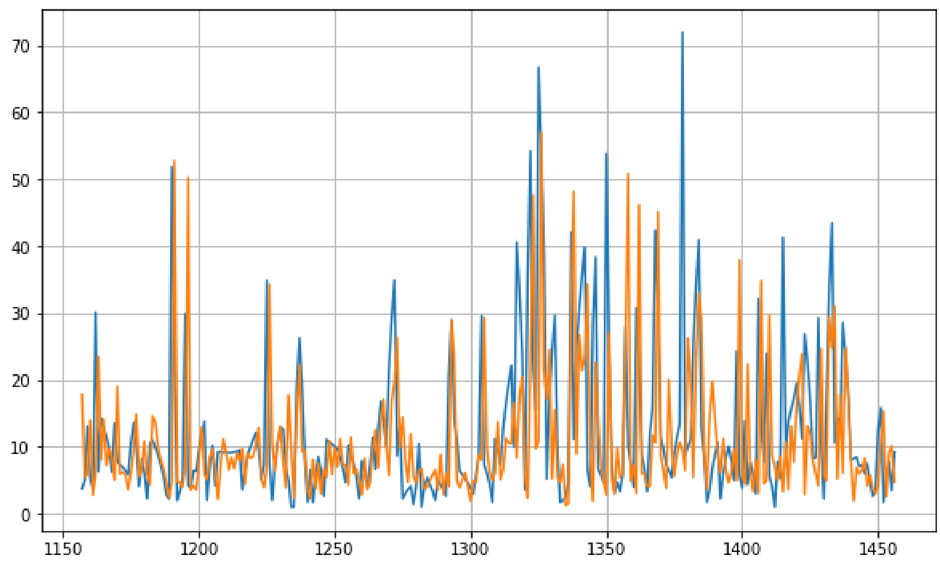


Figure 4.13: DNN forecasting against existing data for "Binders" sub-category

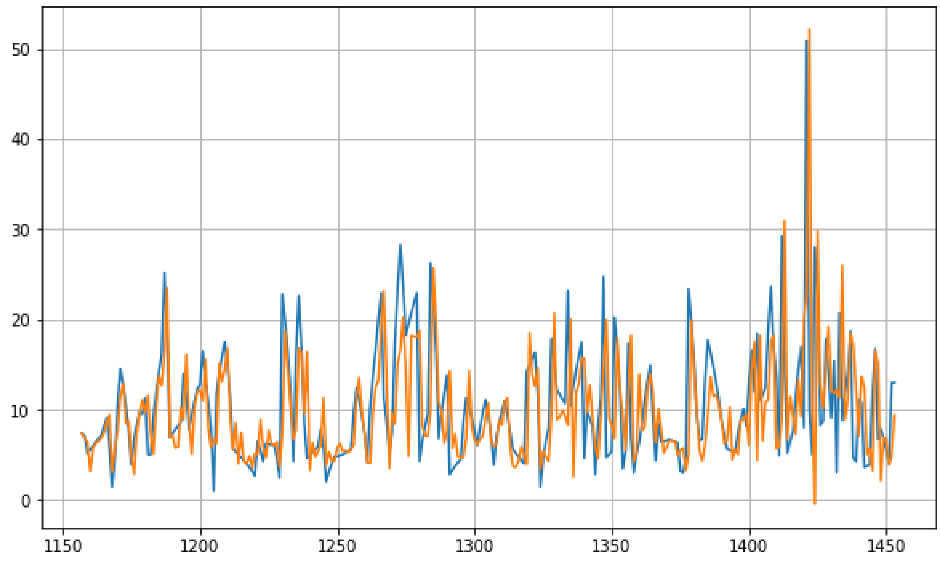


Figure 4.14: DNN forecasting against existing data for "Furnishings" sub-category

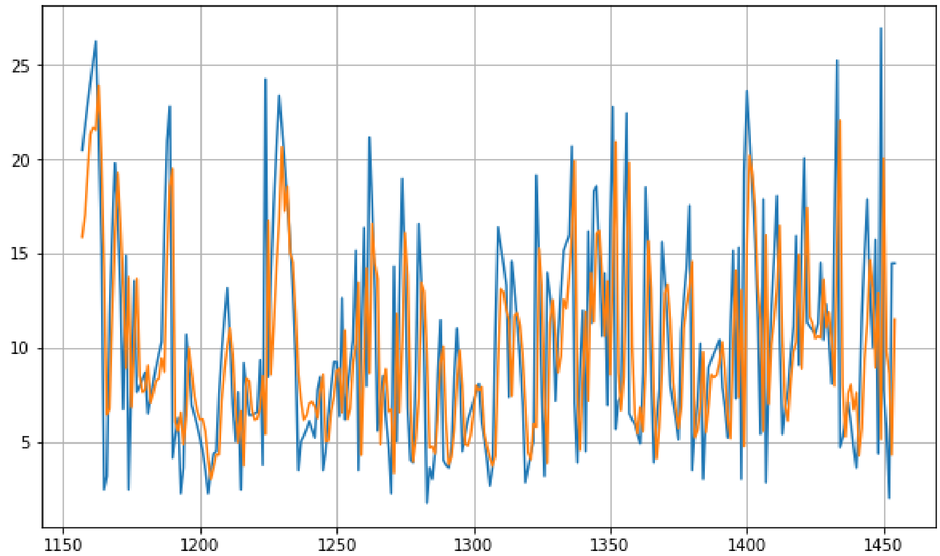


Figure 4.15: DNN forecasting against existing data for ”Papers” sub-category

4.1.6 CNN-RNN-DNN

The hyper-parameter is tuned in CNN-RNN-DNN to generate the model with the create model() method. The adjust learning rate() method is used to control the learning rate using this model. We train our model and predict to cross-validate our dataset after obtaining the optimal learning rate.

We can split the training dataset into train and test sets, fit the model with the train set, and then use the train set to make predictions for each element on the test set. We use the forecast() technique to get proper results after training 1157 days of data and forecasting the next 300 days. Binders, furniture, and papers were chosen as the three principal sub-categories, each with 1157 days of data to train and 300 days of data to test.

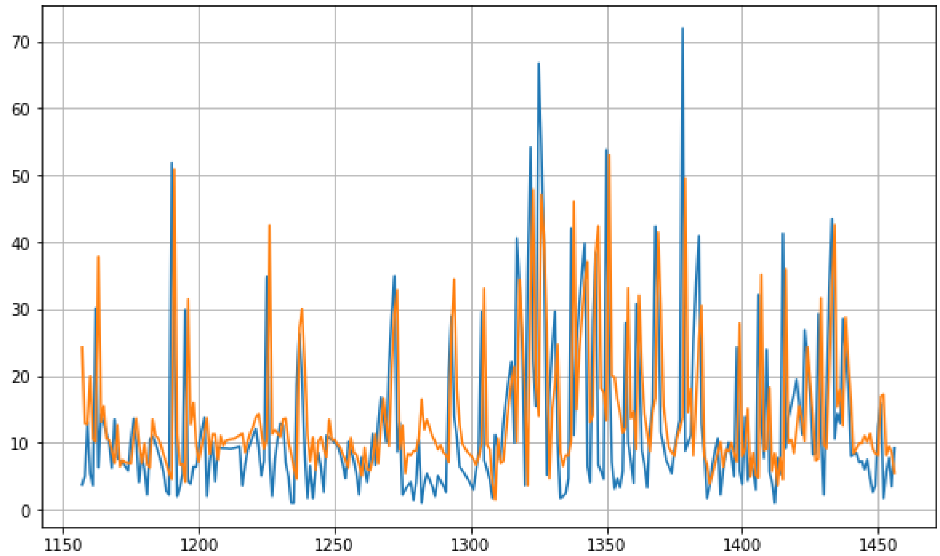


Figure 4.16: CNN-RNN-DNN forecasting against existing data for ”Binders” sub-category

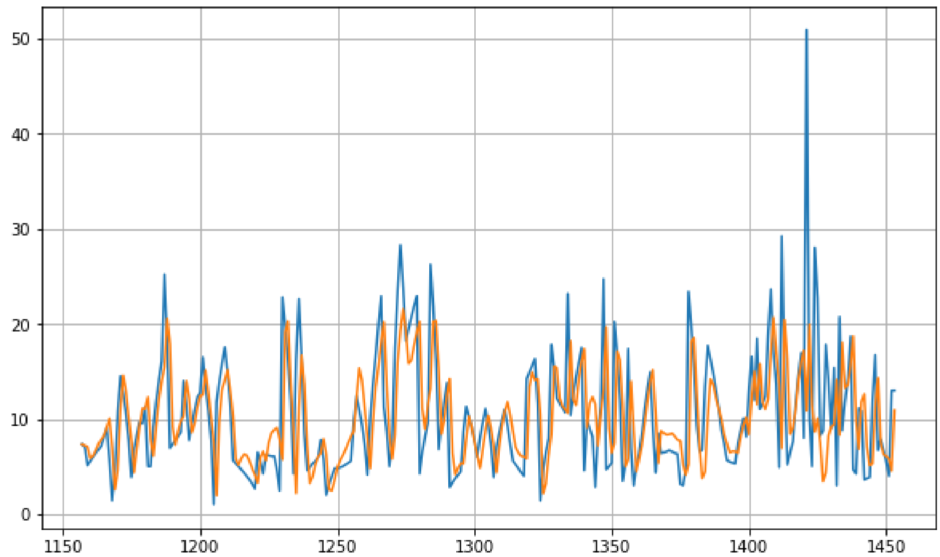


Figure 4.17: CNN-RNN-DNN forecasting against existing data for ”Furnishings” sub-category

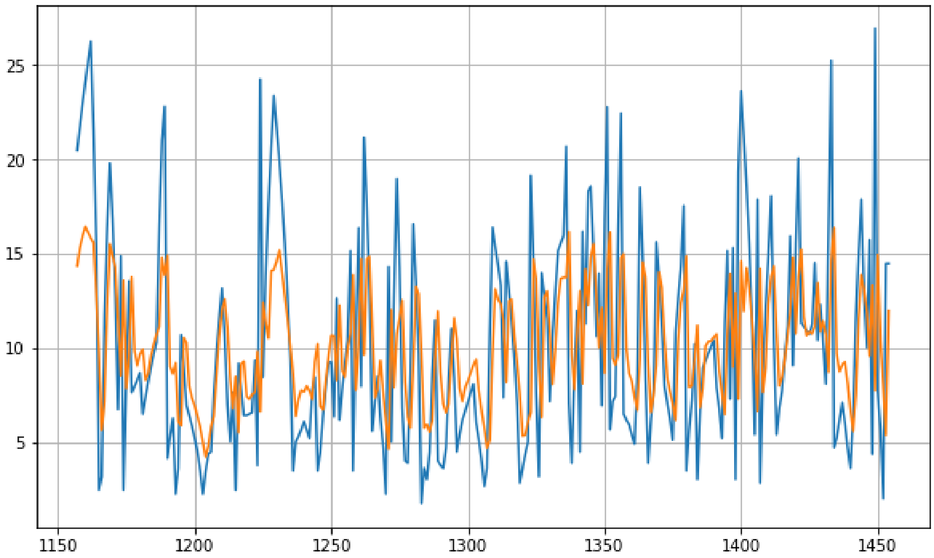


Figure 4.18: CNN-RNN-DNN forecasting against existing data for "Papers" sub-category

The predicting of three sub-categories - Binders, Furnishings, and Papers - is shown in Figures 4.16 - 4.18. Each graph depicts the relationship between time and sales validation. The real set is defined by the blue line, while the forecasted set is defined by the orange line.

4.2 Evaluation

After implementing all the models and training and testing the data we are comparing the models based on three parameters - MSE, RMSE, and MAPE.In the given table we have shown the results of three categories out of seventeen.

CNN-RNN-DNN > DNN > Prophet > Nural prophet > ARIMA > SARIMAX

Figure 4.19: Model Comparison

Model	Sub-category	MSE	RMSE	MAPE
DNN	Binders	158.78	12.6	0.84
	Paper	27.23	5.22	0.51
	Furnishings	37.74	6.14	0.44
CNN-RNN-DNN	Binders	160.03	12.65	1.16
	Paper	24.43	4.94	0.53
	Furnishings	33.13	5.760	0.5
ARIMA	Binders	368218.38	606.81	11.64
	Paper	35.05	5.92	0.53
	Furnishings	37.95	6.16	0.67
SARIMAX	Binders	188946.71	434.68	13.52
	Paper	41.2164	6.42	0.79
	Furnishings	51.1225	7.15	1.17
Prophet	Binders	5444101.64	737.63	25.18
	Paper	25.18	286.97	3.89
	Furnishings	34457.06	185.63	4.74
Neural Prophet	Binders	223654.85	502.03	4.24
	Paper	233858.88	505.01	4.46
	Furnishings	203759.08	500.05	4.06

Table 4.1: Results

Chapter 5

Result and Limitations

5.1 Performance Analysis

We will measure RMSE value to compare the performance of the algorithms based on the accuracy.

From Table 4.1 we can see CNN-RNN-DNN performed better than any other models on the three subcategories. DNN came in second place, and Prophet functioned well as well.

This hybrid model performed better because CNN and RNN cooperate together. CNN always learns to recognize patterns across the dataset. Conv1D layer use conV(), maxPool(), and flattened() function to generate an output by recognizing pattern which later used as RNN input. In terms of our dataset, CNN architecture just has to recognize pattern as we are using supervised-learning dataset. After this we are using LSTM layer architecture as an RNN model which takes CNN architecture output as its input values to provide sequence prediction.

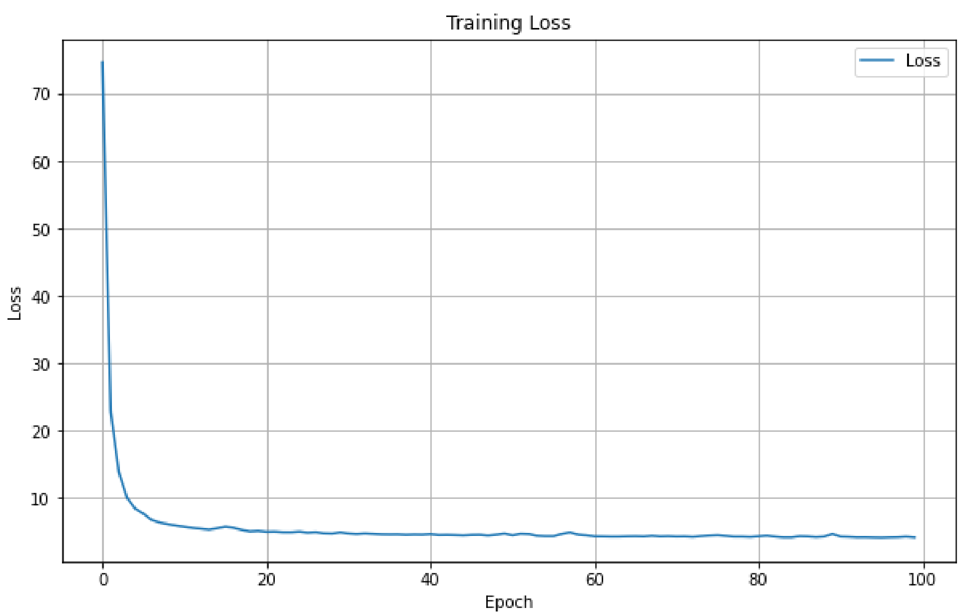


Figure 5.1: Training Loss

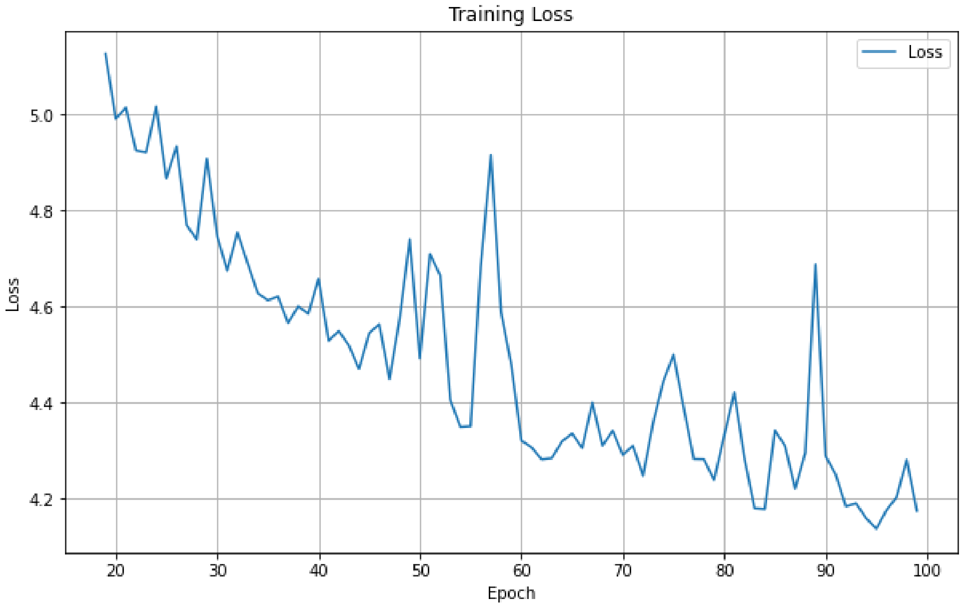


Figure 5.2: Training Loss(Zoom)

Loss value indicates how well or poorly a certain model behave after each iteration of optimization. The lower the loss the better the model performs. From figure 5.1 we can see that the larger the epoch is the lower the loss is. From figure 5.2 we can see that epoch vs loss graph is downward trending. As a result we can say that it does not have any overfitting issue.

Recommendation System:

After comparing all the algorithm we can see the best one is CNN-RNN-DNN hybrid model. However, Prophet also performs well in this genre because it keeps out the outliers like promotional sales, holiday offers, etc. So, we have design a system which will divide the sub-categories and train those sub category data. Based on all categories it will recommend the sales of future months without outliers.

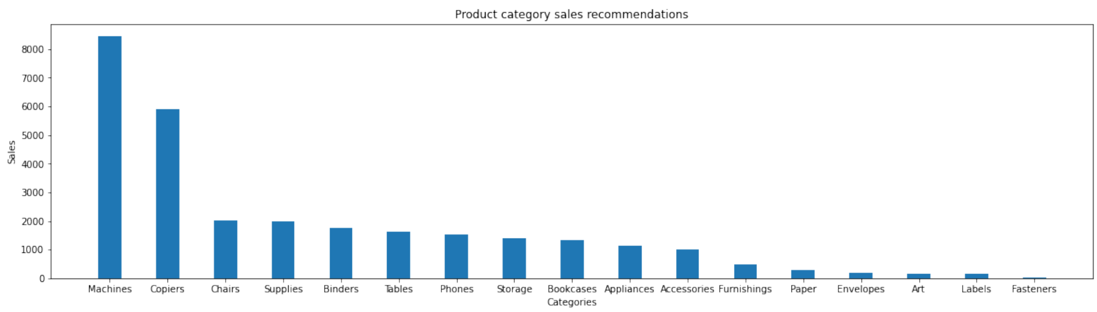


Figure 5.3: Sales Recommendation(January-2018)

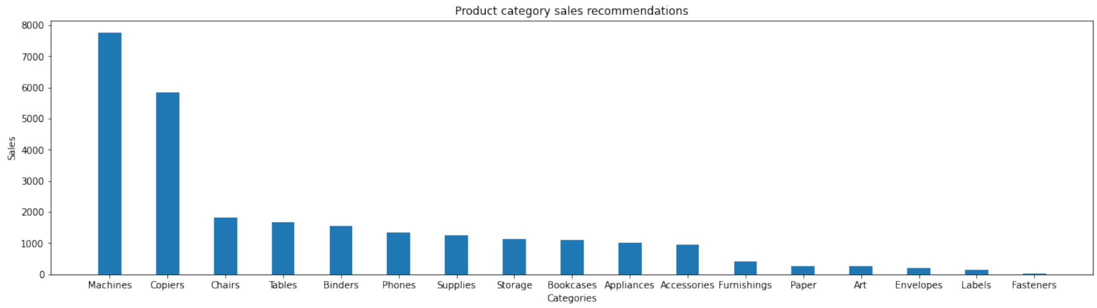


Figure 5.4: Sales Recommendation(February-2018)

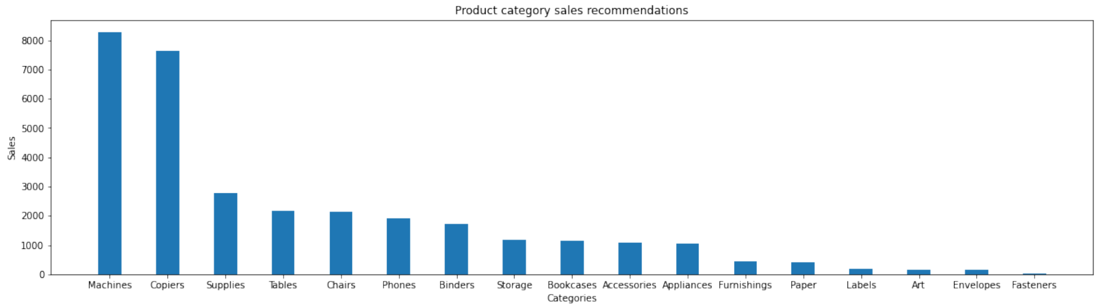


Figure 5.5: Sales Recommendation(March-2018)

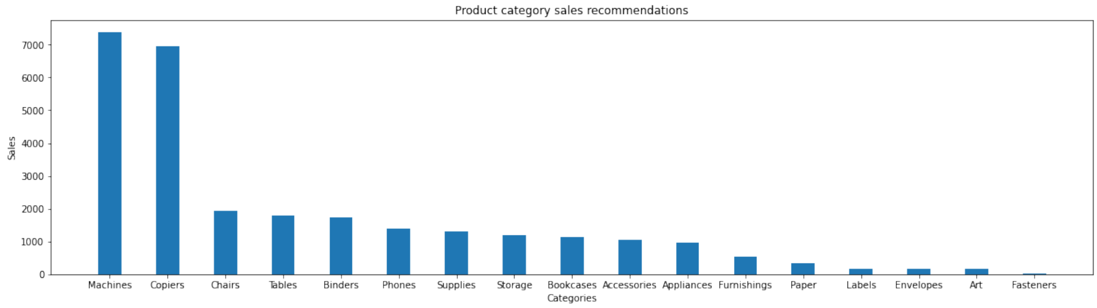


Figure 5.6: Sales Recommendation(April-2018)

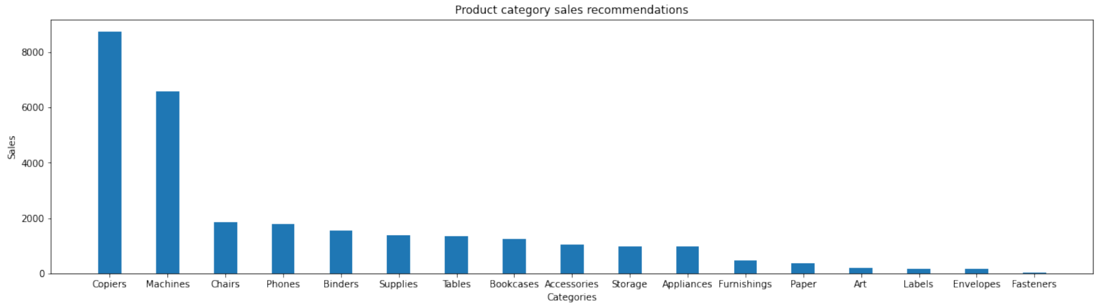


Figure 5.7: Sales Recommendation(May-2018)

5.2 Limitations

While completing the research, we have faced some limitations and complications which are listed below -

1. Scarcity of Sales Data

Our topic was data specific. For this very reason we have hard time finding supply chain and warehouse data as our initial plan was to find product stocking as well as sub-category wise prediction.

2. Dataset size

As our main target was to give a prediction on sub-category, we need a large scale data. However, it was hard to find appropriate data in the internet and also private companies refused to share there datas.

From the result, we can see every models RMSE and MAPE KPI results. However, Prophet or Neural Prophet did not perform better than our expectation.

2. Data was not stationary

Our dataset was not stationary and we have to make data stationary.

2. No Continuous Data

In our datasets, there was no continuous data in any sub-category or single product. For this we have a hard time to find future prediction.

Chapter 6

Conclusion and Future Work

The key challenge in supply chains in the retail business is demand forecasting, as it helps to optimize supplies, decrease costs, and boost sales, profit, and customer loyalty[2]. Demand forecasting helps with financial planning, production planning, risk assessment, and raw material acquisition, among other things. Most importantly, forecast accuracy helps firms to minimize stock-outs and overstocking, as well as save money, increase operational efficiency, and boost customer satisfaction[2]. Despite the fact that forecasting is an important part of most retail operations, many people are still confused about what a sales vs demand prediction is. All merchants use some form of forecasting to anticipate the future, and the reasoning is simple. Without predicting how many goods will sell tomorrow, it's impossible to build a lucrative and long-lasting retail firm[15]. Despite the fact that many academics have worked on demand forecasting using machine learning and a few have utilized deep learning to get the best results, no one has concentrated on small business improvement or selectable characteristics based on deep learning. We intend to provide a system that will allow not only small businesses but also huge enterprises to benefit from the complicated deep learning algorithms. We use six different algorithms - ARIMA, OLS-ARIMA, Prophet, NeuralProphet, DNN, CNN-RNN-DNN to see their accuracy. Among them CNN-RNN-DNN gives the best forecast.

We want to work furthermore in the future. We wish to work on building a framework consists of CNN-RNN-DNN hybrid model. We will handle product stocking (overstocking and out of stock) concerns as well as market demand for products using demand forecasts. The proposed technique will be implemented and assessed with real-world data. Prophet can take promotion sales or holiday effect data called outlier. We will implement this feature in our proposed framework.

Bibliography

[1] Z. H. Kilimci, A. O. Akyuz, M. Uysal, *et al.*, "An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain," *Complexity*, vol. 2019, 2019.

[2] B. Angerhofer and M. Angelides, "System dynamics modelling in supply chain management: Research review," in *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, vol. 1, Dec. 2000, 342–351 vol.1. DOI: 10.1109/WSC.2000.899737.

[3] S. Punia, S. P. Singh, and J. K. Madaan, "A cross-temporal hierarchical framework and deep learning for supply chain forecasting," *Computers Industrial Engineering*, vol. 149, p. 106 796, 2020, ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106796. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835220305040.

[4] R.-Z. Xu and M.-K. He, "Application of deep learning neural network in online supply chain financial credit risk assessment," in *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, 2020, pp. 224–232. DOI: 10.1109/CIBDA50819.2020.00058.

[5] M. Abolghasemi, E. Beh, G. Tarr, and R. Gerlach, "Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion," *Computers Industrial Engineering*, vol. 142, p. 106 380, 2020, ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106380. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835220301145.

[6] Y. Feng and S. Wang, "A forecast for bicycle rental demand based on random forests and multiple linear regression," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, May 2017, pp. 101–105. DOI: 10.1109/ICIS.2017.7959977.

[7] C.-H. Wang and J.-Y. Chen, "Demand forecasting and financial estimation considering the interactive dynamics of semiconductor supply-chain companies," *Computers Industrial Engineering*, vol. 138, p. 106 104, 2019, ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2019.106104. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S036083521930573X.

[8] R. Carbonneau, K. Laframboise, and R. Vahidov, "Application of machine learning techniques for supply chain demand forecasting," *European Journal of Operational Research*, vol. 184, no. 3, pp. 1140–1154, 2008, ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2006.12.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221706012057.

[9] L. Abderrezak, M. Mourad, and D. Djalel, “Very short-term electricity demand forecasting using adaptive exponential smoothing methods,” in *2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Dec. 2014, pp. 553–557. DOI: 10.1109/STA.2014.7086716.

[10] H. Abbasimehr, M. Shabani, and M. Yousefi, “An optimized model using lstm network for demand forecasting,” *Computers Industrial Engineering*, vol. 143, p. 106 435, 2020, ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106435. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835220301698.

[11] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A comparison of arima and lstm in forecasting time series,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401. DOI: 10.1109/ICMLA.2018.00227.

[12] H. Nguyen, K. Tran, S. Thomassey, and M. Hamad, “Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management,” *International Journal of Information Management*, vol. 57, p. 102 282, 2021, ISSN: 0268-4012. DOI: https://doi.org/10.1016/j.ijinfomgt.2020.102282. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026840122031481X.

[13] J. Noh, H.-J. Park, J. Kim, and S.-J. Hwang, “Gated recurrent unit with genetic algorithm for product demand forecasting in supply chain management,” *Mathematics*, vol. 8, p. 565, Apr. 2020. DOI: 10.3390/math8040565.

[14] Y. Li, Y. Yang, K. Zhu, and J. Zhang, “Clothing sale forecasting by a composite gru–prophet model with an attention mechanism,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8335–8344, 2021. DOI: 10.1109/TII.2021.3057922.

[15] S. Piramuthu, “Machine learning for dynamic multi-product supply chain formation,” *Expert Systems with Applications*, vol. 29, no. 4, pp. 985–990, 2005, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2005.07.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417405001351.

[16] E. Hofmann and E. Rutschmann, “Big data analytics and demand forecasting in supply chains: A conceptual analysis,” *The International Journal of Logistics Management*, 2018.

[17] A. Kochak and S. Sharma, “Demand forecasting using neural network for supply chain management,” 2014.

[18] X. V. Pham, A. Maag, S. Senthilananthan, and M. Bhuiyan, “Predictive analysis of the supply chain management using machine learning approaches: Review and taxonomy,” in *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, 2020, pp. 1–9. DOI: 10.1109/CITISIA50690.2020.9371842.