-SALES PREDICTION USING ARIMA MODEL

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Shubham Tandon(181240)
Shaurya Pratap Singh(181492)

Under the supervision of

Mr.Deepak Gupta

to

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

TABLEOFCONTENTS

I
Candidate's Declaration

I hereby declare that the work presented in this report titled "Sales prediction using Arima model" in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out Under the supervision of Mr.Deepak Gupta(Asst.Professor Senior Grade)
The report's contents have not been submitted for the granting of any other degree or certificate.

(Student Signature)
Name:Shaurya Pratap
Roll no.:181492

(Student Signature)
Name:Shubham Tandon
Rollno.: 181240

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Supervisor Name: Mr.Deepak Gupta

(Asst.Professor Senior Grade)

Department name: CSE and IT department

Dated:6/03/2022

CERTIFICATE

This is to certify that the work which is being presented in the project report titled **"**Sales Prediction using ARIMA model" is in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Shaurya Pratap Singh and Shubham Tanodn during the period from January 2022 to May 2022 under the supervision of MR. Deepak Kumar Gupta , Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Shaurya Pratap Singh(181492)

Shubham Tandon(181240)

The above statement made is correct to the best of my knowledge.

III
ACKNOWLEDGEMENT

To begin, I would like to offer my heartfelt gratitude and appreciation to Almighty God for His wonderful grace, which has enabled us to successfully finish the project work.

I am really grateful and desire to express my heartfelt gratitude to my supervisor, Mr.Deepak Gupta(Asst.Professor Senior Grade) Department of CSE Jaypee University of Information Technology, Wakhnaghat. My supervisor's deep knowledge and genuine interest in the subject of Machine Learning are essential for carrying out this research. His unending patience, intellectual direction, consistent encouragement, persistent and vigorous supervision, constructive criticism, helpful counsel, reading many poor versions and correcting them at all stages, and reading and correcting them at all stages enabled us to accomplish this project.

I would like to convey my heartfelt thanks to Mr.Deepak Gupta, Department of CSE, for his generous assistance in completing my research. I would also like to express my gratitude to everyone who has directly or indirectly assisted me in making this project a success. In this unusual scenario, I would like to thank the many staff members, both teaching and non-teaching, who have created their convenient assistance and helped my project.

Finally, I must express my gratitude for my parents' unwavering support and patience.

Shubham Tandon
(181240)

Shaurya Pratap Singh
(181492)

ABSTRACT

The work provided in this paper a contribution to utilising time series technique to model & forecast sales for shampoo manufacturer. Our research shows how previous sales data may be used to estimate future sales, as well as how these forecasts influence the supply chain. Several autoregressive integrated moving averages were developed using the early sales data (ARIMA) You may forecast time series using the ARIMA model & the series previous values. We design an optimum ARIMA model from the ground up in this post, & then expand it to Seasonal ARIMA (SARIMA) & SARIMAX models. The chosen model was the ARIMA (1, 0, 1), which was confirmed using previous sales data under the same conditions. The acquired findings show that the model may be used to predict & anticipate future sales in this shampoo firm. An ARIMA model is type of statistical model i.e used to analyse & forecast data using time series. The acronym ARIMA refers to AutoRegressive Integrated Moving Average. It is generalisation of AutoRegressive Moving Average that incorporates concept of integration. Large-scale health initiatives are increasingly being evaluated using interrupted time series analysis. While segmented regression is frequent method, it is not always sufficient, especially when seasonality & autocorrelation are present. An alternate technique that potentially address these concerns is Autoregressive Integrated Moving Average (ARIMA) model.

A systematic approach of discovering, fitting, verifying, & employing integrated autoregressive, moving average (ARIMA) time series models is known as Box - Jenkins Analysis. The approach is suitable for medium to long-term time series (at least 50 observations). In this chapter, we'll go through the basics of Box-Jenkins technique, focusing on practical aspects rather than theory. The majority of what is discussed here is taken from George Box & Gwilym Jenkins' seminal work on time series analysis (1976). A time series is collection of values that are observed consecutively across time. X1 X2 Xt, where t is time period & X is value, can be used to represent series. The series is said to be deterministic if

Xs are precisely defined by a mathematical formula. The series is said to be statistical or stochastic if future values can only be represented by their probability distribution. A stationary stochastic process is type of stochastic process. If the probability distribution is same for all starting values of t, the statistical process is stationary. This means that for all t values, mean & variance are constant. Because values of the series are dependent on t, a series with a simple trend is not stationary. The mean, variance, & autocorrelation function completely define a stationary stochastic process. One of procedures in Box-Jenkins method for converting non-stationary series to stationary series is to use chevaliers.

CHAPTER-1

INTRODUCTION

1.1 About Sales Prediction Using Arima Model:

We use Arima Model to forecast sales in this project. Sales forecasting is essentially process of estimating future sales. With use of an algorithm & cutting-edge technology, businesses may accurately anticipate product sales in week, month, or year. The ARIMA model with help of machine learning was used to create this model. The abbreviation ARIMA (Auto Regressive Integrated Moving Average) refers to group of mathematical models that can be used to depict a phenomenon defined by time series. With confidence interval round forecasts, an ARIMA model can be used to predict future values of series characterising the phenomena. Let {} ∈ℕ * denote time series that describes a certain phenomenon & its mean values. We shall observe descriptive analysis of our data & subsequently the application of ARIMA MODEL in this project. So that we can get information we need about the product's present & prospective sales. We also utilise SARIMA model in this which combines seasonal differencing with an ARIMA model & is used for time series data modelling with periodic characteristics or we can say seasonal ARIMA used with time series with seasonality.

AutoRegressive Integrated Moving Average (ARIMA) is an acronym for AutoRegressive Integrated Moving Average. It's more complex version of AutoRegressive Moving Average, with addition of concept of interation.

This abbreviation is descriptive, capturing model's major features. They are, in brief:
Autoregression (AR). The dependent relationship between an observation & set of lagged observations is used in this model.
Integrated (I) To make the time series steady, differencing raw observations (e.g.

subtracting an observation from an observation from preceding time step) is used.

MA stands for Moving Average. The dependency between an observation & residual error from a moving average model applied to lagged observations is used in this model.

Each of these elements is explicitly described as parameter in model. ARIMA(p,d,q) is a standard notation in which parameters are replaced with integer values to immediately indicate ARIMA model being utilised.
The following are parameters of the ARIMA model:
p: The lag order, or the number of lag observations incorporated in the model.

d: The degree of differencing is number of times raw observations are differenced.

q: The order of moving average, also known as the size of the moving average window.

A linear regression model with appropriate number & kind of terms, & data processed using a degree of differencing to make it stationary, i.e, to remove trend & seasonal structures that negatively cheval regression model.
A parameter with a value of 0 indicates that that piece of model should not be used. This allows ARIMA model to mimic functionality of an ARMA model or even a basic AR, I, or MA model.
When you use an ARIMA model to analyse a time series, you're assuming that underlying process that generated data is itself an ARIMA process. This may seem self-evident, but it helps to justify need to test the model's assumptions in raw observations & residual errors of model forecasts.
Next, let’s take a look at how we can use ARIMA model in Python. We will start with loading a simple univariate time series.

.

1.2 Problem Statements:

The background of project is that there is problem with previous systems in that accuracy rate of these models is low, & number of algorithms used in majority of projects is low, with the highest number of algorithms being 3, which is why for this system, where user can provide input & detect following error in pattern. Organizations are moving toward more effective sales-driven supply chain in today's competitive manufacturing market, in order to adapt rapidly to altering sales. Customers have become more salesy & discerning, dictating to suppliers what things they want & when they need them delivered, transforming the market into a "pull" environment.

Forecasting sales is essential for inventory management. Inventory levels are determined by sales estimates. In reality, poor sales forecasting can result in significant expenses, demonstrating that process has not improved. As result, many systems invest heavily on inventory to minimise "stock outs." Another difficulty is that certain sales are intermittent, meaning that there are times when we have no sales & other times when we have consecutive sales. Traditional statistical sales forecasting approaches are challenged by intermittent sales. In general, there are a variety of ways to forecasting sales, including exponential smoothing. However, in order to use these methods, we require previous data. Because there is no information about the history at outset, we must make an educated guess based on previous circumstances or engineer experience. We have a lot of ambiguity in this case, but it will pass with time.

For most businesses, controlling sales is difficult due to difficulties of effectively projecting future consumer needs. Poor forecasting accuracy & sales volatility are developing important obstacles to supply chain flexibility, according to more than 74 percent of respondents in research survey. The most successful firms improve supply chain flexibility,agility, & responsiveness by enhancing forecasting accuracy across whole supply chain.

Forecasting must be linked to improvement goals, & historical performance must be used to avoid past errors & achieve a high level of efficiency.

Researchers have done a lot of work in forecasting arena & proposed a lot of methods, but two of most popular are time series approaches & artificial neural network (ANN) techniques. ANN models have had great success with forecasting sales.The se models are characterised by sales intervals with significant variety. When it comes to ability to capture nonlinearity in data sets, ANN technique is considered an alternative. ANN is used in various fields. For scenario of deterministic time-varying sales, Gaafar & Choueiki used neural network model to solve a lot-sizing problem as part of material requirements planning.

1.3 OBJECTIVE:

The main goal of this project is to forecast sales of shampoo firm using company's prior sales data. ARIMA model & Box-Jenkins time series are used to forecast.

1.4 METHODOLOGY:

The technique we employed in this project was Machine Learning (ML), which is study of computer systems that improve themselves over time. Artificial intelligence is associated with it. Machine learning algorithms create a model based on training data to make predictions or judgments without being specifically trained to do so. Python is an interpreted high-level general-purpose programming language that we employed here.

Time series properties

A time series is chronologically ordered sequence of data points at evenly spaced moments in time. Non-stationarity, autocorrelation, & seasonality are three characteristics of time series.

Non-stationarity

The time series must be stationary for ARIMA modelling to work. A stationary series has three characteristics: constant mean, constant variance, & constant covariance i.e independent of time interval between values. A stationary series (sometimes known as

"white noise process") is easier to analyse since it has fewer parameters to model. While it fluctuates, it always returns to consistent mean, making it easier to predict. The first is changing variance with time (heteroscedasticity), which may easily be addressed by using log trans formation; second is an increasing or declining trend, which can often be avoided by taking first difference (i.e. Yt Yt

1). A second differencing may be necessary to establish stationarity on rare occasions, although third order differencing & above is uncommon. To be precise, definition above applies to a weakly stationary series. If prob ablity distribution of a sequence of observations is unaltered by time changes, time series is deemed strictly stable. Strictly stationary series are uncommon, thus weak stationarity is generally assumed.

Autocorrelation

Time series observations are frequently associated with data from prior time points, & thus are not dispersed randomly. Autocorrelation or serial correlation is name given to this type of correlation. Time series with autocorrelation do not meet normal regression analysis assumptions, as previously stated. Because autocorrelated data are rarely steady, differencing data frequently enough to re-move autocorrelation is necessary before testing for autocorrelation. Stationarity & autocorrelation can be checked using autocorrelation functions (ACFs). An ACF depicts correlation between each observation & previous values at different lags, where a lag is number of time points between two observations The partial ACF (PACF), which is the correlation between an observation & historical values i.e not explained by correlations at lower order lags, is a companion to ACF. For example, after adjusting for correlation between Yt & Yt 3, Yt 2, & Yt 1, PACF value at lag 4 is the correlation between an observation (Yt) & prior observation at lag 4 (Yt 4). The autocorrelation in ACF plot should decrease quickly for stationary series; the ACF will decay slowly for a non-stationary series.

Seasonality

Seasonality refers to changes in a definite or recognised frequency that occur at regular intervals, such as time of year or week. Seasonality is widespread in health data time series, & it can be caused by natural factors like weather patterns or business/administrative processes like weekend or holiday effects. Antibiotic prescriptions & influenza hospitalizations, for example, are more common in winter [11, 12]. Furthermore, due to financial incentives to stockpile pharmaceuticals, medicine dispensings are highest near end of a calendar or financial year in several countries [13, 14]. Seasonality will vary depending on series' unit of time; for example, seasonality is uncommon in time series measured in years intervals. With seasonal monthly data, ACF plot will almost certainly show strong autocorrelation at lag 12. Seasonality is commonly handled in ARIMA models by taking seasonal difference. You take difference between each observation & previous value at lag 12 (Yt Yt 12) using monthly data. You'd use lag 4 for quarterly data. The first 12 observations are lost when calculating seasonal difference for monthly data since seasonal difference cannot be calculated for those observations. This is critical to remember: if your data is seasonal, you'll need additional time points in your series to adequately adjust for seasonal impacts.

Components of ARIMA models

ARIMA models feature single dependent variable (Yt), which is function of previous Y values as well as error term (t). ARIMA models may accommodate any continuous output (such as rates or means) as well as high counts that are not bounded by zero because they assume mistakes are normally distributed. While ARIMA cannot be utilised with small counts that follow a Poisson distribution, generalised linear models have been used to model serially correlated count data in recent years. We introduce essential components before moving on to comprehensive ARIMA models..

1. Autoregressive (AR) model: one or more lagged values of Yt predict Yt. This is expressed by equation below, in where c is a constant, is autocorrelation magnitude, p is number of lags, & t is error.

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \epsilon_t$$

2. Model of average (MA): One or more lagged values of error (t) are used to predict Yt. This is not same as moving average smoothing. q is number of lags, & is value of autocorrelation of mistakes in equation below.

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

3. Seasonal model: Lagged Yt readings at regular intervals s are used to predict Yt (the season). The autocorrelation value is & seasonality is in equation below (e.g. 52 for weekly, 12 for monthly, 4 for quarterly). Differentiation, as well as autoregressive &/or moving average components, are frequently required in seasonal models.

$$Y_t = c + \Phi Y_{t-s} + \epsilon_t$$

4. Differencing (Integration): To achieve meaningful predictions in an ARIMA model, time series being modelled must be stationary. Differentiating, or calculating difference between adjacent data, causes stationarity.

$$Y'_t = Y_t - Y_{t-1}$$

An ARIMA model is a hybrid of an AR, MA, & differencing models (Integration). If = 0, then time series is a white noise process given as Yt = c + t, where c is a constant. (p, d, q), where p, d, & q are positive integers, is basic notation for defining a non-seasonal ARIMA model:

p = the AR component of the model's order;

d = non-seasonal differencing degree; &

q = the MA portion of model's order.

A white noise (stationary) model ARIMA, for example (0, 0, 0). ARIMA(p, 0, 0) for an AR model, & ARIMA(q, 0, 0) for an MA model (0, 0, q) If seasonality exists, ARIMA model is written as (p, d, q)(P, D, Q)S. P & Q are AR & MA terms for the seasonal component, & D is the degree of seasonal differencing.

Using ARIMA to evaluate interventions

When used to evaluate interventions, goal of ITS analysis is to estimate influence of intervention's implementation on a specific outcome, or "intervention effect." While there are many different sorts of impacts that can be noticed, we'll focus on three of most common: step change, pulse, & ramp. These can be summarised as follows if we use T0 to denote intervention's start time:Step change (also called level shift): A sudden, sustained change where time series shifted either up or down by given value immediately following intervention. The step change variable takes value of 0 prior to start of the intervention, & 1 afterwards.

$$S_t = \begin{cases} 0, if\ t < T_0 \\ 1, if\ t \geq T_0 \end{cases}$$

- Pulse: A sudden, temporary change i.e observed for one or more time points immediately after intervention & then returns to baseline level. The pulse variable takes value of 1 on the date of the intervention, & 0 otherwise.

$$P_t = \begin{cases} 0, if\ t \neq T_0 \\ 1, if\ t = T_0 \end{cases}$$

- Ramp: A change in slope that occurs immediately after the intervention. The ramp variable takes the value of 0 prior to the start of the intervention & increases by 1 after the date of the intervention.

$$R_t = \begin{cases} 0, \ if \ \ t < T_0 \\ t - T_0 + 1, \ if \ \ t \geq T_0 \end{cases}$$

The shape of the intervention's impact should ideally be predicted ahead of time. The form is determined by a number of criteria, including the type of intervention, such as whether it is temporary or ongoing, & the specific outcome being measured. For example, in 2015 study, we looked at the infuence of unfavourable media surrounding statin medicine useand discovered that this transient event led in both a momentary rise in statin discontinuance (a "pulse") & a sustained decrease in statin dispensing (a "step changeLong-term effects are more likely with ongoing or permanent actions, such as tighter restrictions on medicine prescribing or the introduction of plain packaging on cigarette products, however these can be immediate or gradual (a "ramp"). For some interventions, combination of impact factors best represents the change; for example, it is usual to have both step change & change in slope (ramp). If there are several viable models, the Akaike information criterion (AIC) &/or the Bayesian information criterion (BIC) can be used to choose the best combination of effect variables It's also vital to evaluate whether changes will occur before the intervention is implemented; for example, when it was reported that prescribing of alprazo lam would be restricted in Australia, prescribing of this medicine began to decline in anticipation of this change. Finally, the influence may be suspected of being delayed by one or more time units in some circumstances. To avoid spurious associations, we recommend pre-determining fair period of time in which the influence should be noticed based on subject knowledge or past study. Within this range of alternatives, the most appropriate delay can be identified during the modelling stage ARIMA forecasts Yt in the absence of the intervention (the "counterfactual") in ITS analysis & assesses how the observed differs from this forecast. Unlike segmented regression, the ARIMA model does not require time or seasonal dummy variables since ARIMA can eliminate trends &

seasonality through differencing. The pre- & post-intervention trends cannot be estimated from the model if the trend is abolished via differencing. If estimation of the pre- &/or post-intervention slope is desired, time can be included as a covariate, & AR & MA terms can be used to account for autocorrelation (e.g. ARMA models).

Fitting an ARIMA model

The ARIMA model's parameters are determined in the next phase. The Box Jenkins technique, which involves model identification & selection, parameter estimation, & model checking, is a popular methodology. Automated algorithms in statistical packages (such as R) now make the procedure easier by identifying the best-fitting ARIMA model based on the information criteria (AIC, BIC). However, as shown in Fig. 1, we also detail the manual process.**Plot data to understand patterns**:

Before proceeding to model fitting, plot the time series to understand the patterns, specifically pre-existing trends, seasonal effects, & extreme or outlier values. If outliers are present, how to deal with will depend on their cause & influence on the model & the recommendations are the same for ARIMA as for other regression models. For instance, if the researchers are aware that these extreme values are due to external factors, such as other interventions or known misclassification, these should be explicitly modelled in the data.

- **Transform data to stabilise variance (if necessary).**

  If the variance is changing over time, log-transformation should be applied.

- **Model selection:** While automated algorithms in several statistical packages can identify candidate p & q parameters, they can sometimes be estimated based on the ACF/PACF plots.

    a. **Determine differencing order to induce stationarity**:

    A first order difference is required if there is trend, & d = 1. If there is seasonality, there must be a seasonal difference, & D = 1. The ACF plot or unit-root tests (such as the Dickey-Fuller test) can also be used to determine whether the time series is stationary & whether differencing is necessary. The d & D terms in the model may usually be prespecified in most automated techniques.

    b. **Plot the ACF/PACF of stationarity data to determine potential AR/MA orders:**

    Determine which AR (p/P) or MA (q/Q) orders are required to adjust for lingering autocorrelation after the time series has been transformed &/or differencing. AR terms are usually required if the stationary series has positive autocorrelation at lag 1. If the autocorrelation at lag 1 is negative, the model may require MA terms. Models usually just require AR or MA terms, & very rarely both. However, this is not always the case. Table 1 provides suggestions for choosing the best AR & MA phrases.

    Estimate model & use information criteria to find the best model:

    Estimate your model using the previously determined p, d, q, P, D, & Q variables, & apply information criteria (AIC, BIC) to assist you find the best model. If words were chosen using an automated method, it should be treated as a tool only, as it does not ensure well-fitting model.

- **Check if residuals of chosen model are white noise.**

This can be done by looking at residual plots & using the Ljung-Box test for white noise to formally test for the presence of autocorrelation. Choose other AR &/or MA orders if autocorrelation is still present in the residuals or your model is otherwise a poor fit. Non-normally distributed residuals may benefit from a transformation if the data have not been modified previously. Generally, identifying the AR & MA words is an iterative, trial-and-error process. There may not be a single "correct" model. The goal is to find the best cost-effective model (lowest p/P & q/Q) with a good fit & acceptable autocorrelation & sea sonality controls. The intervention impact can be calculated once the final ARIMA model has been chosen estimated.

- **Transfer functions**

Another advantage of ARIMA models is their ability to model more complicated affects via "transfer functions" beyond the basic intervention impact shapes. The relationship between the intervention & the outcome series Yt is described by transfer functions. They can add lagged effects & vary the relationship between the above inputs (step change, pulse, ramp) & the time series to mimic more complex relationships, such as progressive level shifts or a pulse that decays gradually over time.

The general form of a transfer function $\frac{\omega(B)}{\delta(B)}$, or:

$$Y_t = \mu + \frac{\omega_0 + \omega_1 B + \omega_2 B^2 + \cdots + \omega_h B^h}{1 - \delta_1 B - \delta_2 B^2 - \cdots - \delta_r B^r} X_t + \varepsilon_t$$

Bp Yt = Yt p, where B is the backshift operator. 0 denotes the initial value for the intervention's influence at the moment of intervention (T), is the decay rate, & Xt is the intervention variable in the transfer function (step change, pulse, or ramp). The researcher must specify the values of h & r; h indicates when the effect occurs, while r represents the decline pattern. Model fit statistics (such as AIC & BIC) can aid in determining the best shape for the transfer function as well as the event's

timing (i.e. if the impact was delayed & if so by how much). Table 2 describes the most common scenarios, using the intervention indicator variables described above, & where h = 0, & r = 0 or r = 1. The use of transfer functions is a complex topic, & several texts cover them in more detail

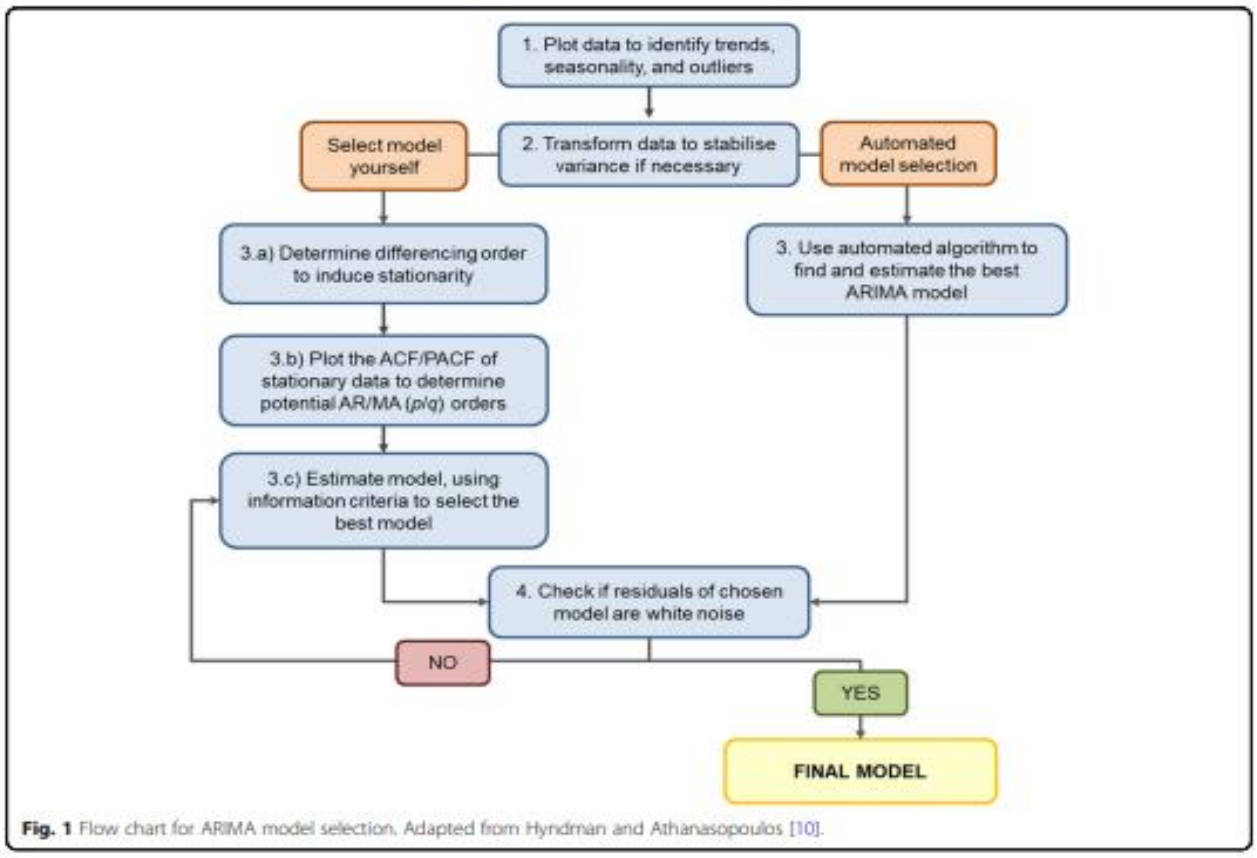
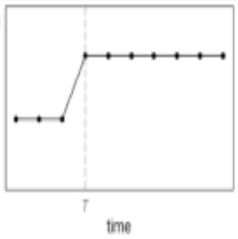
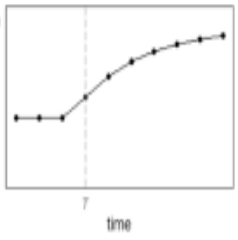
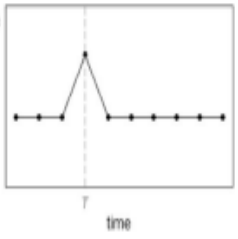
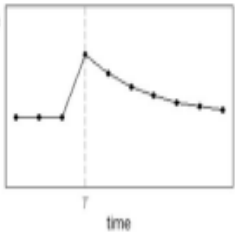
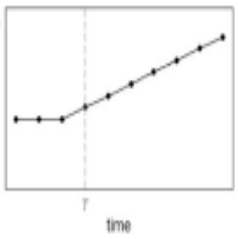


Fig. 1 Flow chart for ARIMA model selection. Adapted from Hyndman and Athanasopoulos [10].

Table 1 Tips for selecting most appropriate autoregressive (p) and moving average (q) terms from autocorrelation and partial autocorrelation

Model type	Characteristics of ACF and PACF	
	ACF	**PACF**
ARIMA(p,d,0)	Tails off or is sinusoidal	Cuts off lag *p*
ARIMA(0,d,q)	Cuts off lag *q*	Tails off or is sinusoidal
ARMA(p,d,q)	Tails off or is sinusoidal	Tails off or is sinusoidal

Table 2 Description of transfer functions for interrupted time series analysis in ARIMA

Function	Values for h and r	Transfer function	Response i at times 0 through k post-intervention	Form of response	Interpretation
Step function $S_t = \begin{cases} 0, if\ t < T \\ 1, if\ t \geq T \end{cases}$	$h=0$, $r=0$	ω_0	$i_0 = \omega_0$ $i_1 = \omega_0$ $i_2 = \omega_0$ … $i_k = \omega_0$		The time series increases by ω_0 immediately following the intervention, and remains at this new level for the duration of the study period.
	$h=0$, $r=1$	$\frac{\omega_0}{(1-\delta_1 B)}$ $(\lvert\delta_1\rvert < 1)$	$i_0 = \omega_0$ $i_1 = \omega_0(1+\delta_1)$ $i_2 = \omega_0(1+\delta_1+\delta_1^2)$ … $i_k = \omega_0(1+\delta_1+\delta_1^2+\ldots+\delta_1^k)$		The time series increases by ω_0 immediately following the intervention, and increases by $\omega_0\delta_1^k$ each subsequent time point until it reaches a new level, calculated by $\frac{\omega_0}{(1-\delta_1)}$.
Pulse function $P_t = \begin{cases} 0, if\ t \neq T \\ 1, if\ t = T \end{cases}$	$h=0$, $r=0$	ω_0	$i_0 = \omega_0$ $i_1 = 0$ $i_2 = 0$ … $i_k = 0$		The time series increases by ω_0 immediately following the intervention and returns to baseline immediately afterwards.
	$h=0$, $r=1$	$\frac{\omega_0}{(1-\delta_1 B)}$ $(\lvert\delta_1\rvert < 1)$	$i_0 = \omega_0$ $i_1 = \omega_0\delta_1$ $i_2 = \omega_0\delta_1^2$ … $i_k = \omega_0\delta_1^k$		The time series increases by ω_0 the time of the intervention, and decays by $(1-\delta_1)$ each subsequent time point.
Ramp function $R_t = \begin{cases} 0, if\ t < T \\ t-T+1, if\ t \geq T \end{cases}$	$h=0$, $r=0$	ω_0	$i_0 = \omega_0$ $i_1 = 2\omega_0$ $i_2 = 3\omega_0$ … $i_k = (k+1)\omega_0$		The time series increases by ω_0 at each time point.

Incorporation of a control series

Because ITS cannot rule out the potential that any observed change was caused by the intervention of interest or another co-intervention or event, using a control series in the analysis improves causal inference. A control series is one i.e unaffected by the intervention; the process of selecting an appropriate control is discussed elsewhere [3]. Including a control series, as ITS in segmented regression, requires running an ARIMA model for the series of interest & separately for the control series [17]. If there is a change in the intervention series but not the control series, this indicates that the impact was unique to the intervention.

Sample size requirements

There is no precise guideline on the number of time points needed to apply ARIMA modelling. The commonly reported estimate of a minimum of 50 time points is based on a statement by Box & Jenkins [23], although it lacks empirical support & has not been rigorously evaluated. A one-size-fits-all strategy is, in reality, simple. More observations will be required to identify the underlying patterns from the noise as the data becomes more varied & noisy. ARIMA can handle short time series satisfactorily in simple circumstances, as long as there are enough time points to estimate all parameters [26]. There should be enough time points in the presence of seasonality to identify seasonal impacts & account for seasonal differences.

CHAPTER-2

LITERATURE SURVEY

Forecasting sales are becoming increasingly important in today's organisations, which are subject to abrupt & enormous changes that affect even the most established of 2 International Journal of Engineering Business Management structures, & where all requirements of the business sector require accurate & practical readings into the future. A forecast is a science that involves predicting the level of particular variables in the future. The variable in question is usually sales, although it might also be supply or pricing. Forecasting is the process of producing predictions about the future values of the variables being researched.

Forecasting sales is one of the most important challenges in inventory management in manufacturing, & it can be employed in a variety of operational planning tasks during the production process, including capacity planning & used-product acquisition management.

Sales forecasts are considered the foundation of supply chain planning for both forms of "push/pull" supply chain activities. The supply chain's pull activities are implemented in response to customer sales, whereas all push processes are implemented in anticipation of client purchases.

Such variables must be considered before a corporation chooses a proper forcasting methodology, as choosing a methodology is not as straightforward as it appears. There are four different sorts of forecasting methods: qualitative, time series, causal, & simulation.

A time series is a collection of observations organised in chronological order.

To forecast sales, time series forecasting models use mathematical methodologies based on previous data.

It is based on the premise that the future is an extension of the past; thus, historical data may be used to forecast future sales.

Many studies about sales forcasting by time series analysis have been done in several domains. They encircle sales forcasting for shampoo product sales,tourism,maintenance repair parts,electricity, automobile,and some other products & services. By time series analysis, the forcasting accuracies depend on the characteristics of time series of sales. If the transition curves show stability & periodicity, we will reach high forcasting accuracies, whereas we can’t expect high accuracies if the curves contain highly irregular patterns.

Yule & Wold's contributions to developing a viable technique to performing ARIMA models were the foundation for Box & Jenkins. Model identification, parameter estimates, & diagnostic checking are all iterative processes in the Box–Jenkins principle. If a time series is created from an ARIMA process, it should have some theoretical autocorrelation qualities, according to the principle rule for identifying the model. We can find one or more plausible models for a given time series by matching the theoretical & empirical autocorrelation characteristics. Box & Jenkins recommended using the sample data's autocorrelation function (ACF) & partial autocorrelation function (PACF) as basic tools for determining the ARIMA model's order In terms of the identification step, we need to create a stationary time series, which is a need for finding the ARIMA model, hence data modification is required. A stationary time series' statistical features, such as the mean & autocorrelation structure, remain

unchanged over time. Before fitting an ARIMA model, we normally need to eliminate the trend & stabilise the variance using differencing & power transformation. After that, calculating the model parameters & specifying the model becomes simple. These parameters are estimated in order to minimise the overall inaccuracy. Finally, we perform diagnostic model adequacy checks.

In this final stage, we confirm that our theory regarding the errors is correct. Diagnostic statistics & residual plots can be used to evaluate the suitability of future values for our data. If the model is inadequate, we must perform further parameter estimations before validating the model. Diagnostic data can assist us in developing new models. The Box–Jenkins model is a method that should be followed & repeated until the model is highly satisfied & errors are minimised. As a result, we can use this model to forecast our variable with ease. Researchers agree that parameter estimation necessitates a large number of observations. As a result, the ARIMA model has several limitations. Nevertheless, once we apply ARIMA model, we reach a high quality in the opposite of the time series models.

2.1 BOX-JENKINS MODEL

The Box-Jenkins Model is a mathematical model that uses inputs from a time series to forecast data ranges. The Box-Jenkins Model can be used to forecast numerous distinct forms of time series data.

Its approach of determining outcomes is based on disparities between data points. The methodology allows the model to recognise patterns & create forecasts utilising autoregresssion, moving & seasonal differencing.

A type of Box-Jenkins model is the autoregressive integrated moving average (ARIMA). ARIMA & Box-Jenkins are terms that are occasionally used interchangeably.

Box-Jenkins models are used to forecast a wide range of expected data points or data ranges, such as company data & future security prices.

George Box & Gwilym Jenkins, two mathematicians, developed the Box-Jenkins Model. In a 1970 publication titled "Time Series Analysis: Forecasting & Control," the two mathematicians outlined the concepts that make up this paradigm.

The Box-Jenkins Model's parameters can be extremely difficult to estimate. As with other time-series regression models, the best results will almost always be obtained by using programmable software. The Box-Jenkins Model is also best for forecasting for periods of 18 months or less.

When using programmed forcasting software, the Box-Jenkins Model may be one of several time series analysis models that a forcaster will encounter. The software will be built to apply the best suitable forcasting methodology depending on the time series data to be forecasted in many circumstances. For data sets that are mainly stable & have little volatility, Box-Jenkins is said to be a great pick.

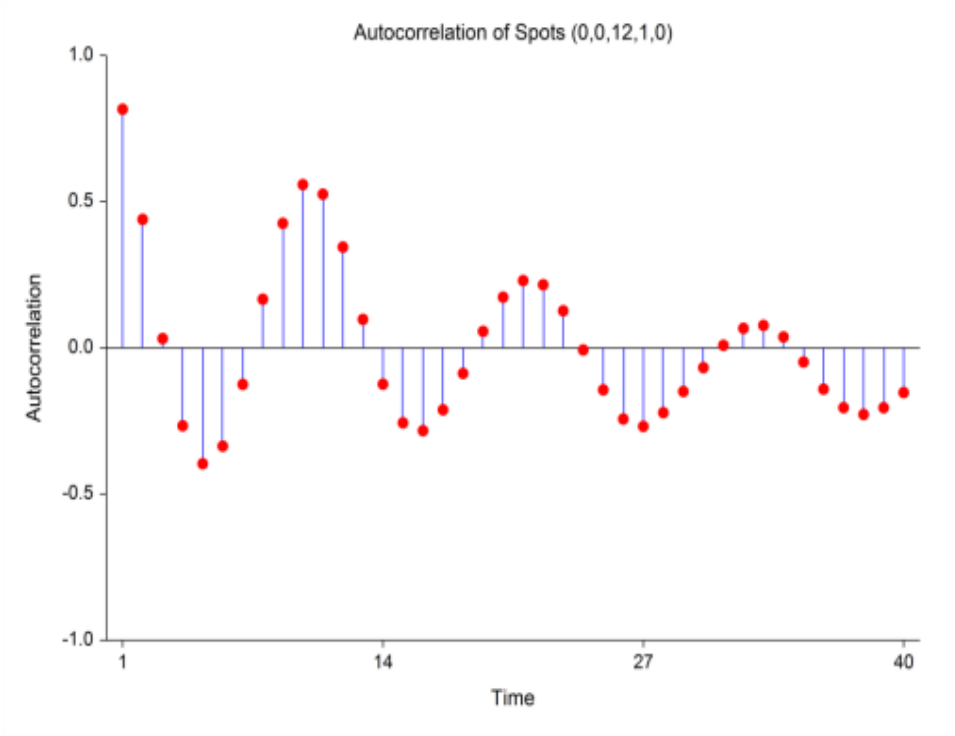
The Box-Jenkins Model uses three principles to forecast data: autoregression, differencing, & moving average. These three principles are referred to as p, d, & q. Each principle employed in the Box-Jenkins analysis is represented individually as ARIMA (p, d, q).

The autoregression (p) process determines whether the data is stationary. It can make the forecasting process easier if the data is static. If the data isn't stationary, it'll have to be differentiated (d). The data was also examined for its ability to fit a moving average (which is

done in part q of the analysis process). Overall, the data is prepared for forecasting by establishing the parameters (p, d, & q), which are then used to create a forecast.

Autocorrelation Function

We can make simple claims regarding the correlation between two subsequent values, Xt & Xt+k, using the stationary assumption. The autocorrelation of lag k of the series is the name given to this correlation. The autocorrelation function plots the autocorrelation of successive values of k on the horizontal axis on the vertical axis. The autocorrelation function of sunspot data is depicted in the diagram below.



Since a stationary series is completely specified by its mean, variance, and autocorrelation function, one of the major (and most subjective) tasks in Box-Jenkins analysis is to identify an appropriate model from the sample autocorrelation function. Although the sample autocorrelations contains random fluctuations, for moderate sample sizes they are fairly accurate in signaling the order of the ARIMA model.

The ARMA Model

The ARMA (autoregressive, moving average) model defined as follows:

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

The a's are a series of unknown random errors (or residuals) that are expected to follow the normal probability distribution. Box- Jenkins makes writing these models easy by using the backshift operator. The backshift operator, B, changes the time period t to the time period t-1. As a result, BXt = Xt1 & B Xt Xt 2 = 2. The aforementioned model could be written in this backshift notation. rewritten as:

$$\left(1-\phi_1 B-\cdots-\phi_p B^p\right)X_t=\left(1-\theta_1 B-\cdots-\theta_q B^q\right)a_t$$

This may be abreviated even further by writing:

$$\phi_p(B)X_t=\theta_q(B)a_t$$

where

$$\phi_p(B)=\left(1-\phi_1 B-\cdots-\phi_p B^p\right)$$

and

$$\theta_q(B)=\left(1-\theta_1 B-\cdots-\theta_q B^q\right)$$

These formulas show that the operators φ p (B) & θ q (B) are polynomials in B of orders p & q respectively. One of the benefits of writing models in this fashion that we can see why several models may be equivalent. For example, consider the model

$$X_t=0.8X_{t-1}-0.15X_{t-2}+a_t-0.3a_{t-1}$$

$$\left(1-0.8B+0.15B^2\right)X_t=\left(1-0.3B\right)a_t$$

Notice that the polynomial on the left may be factored, so that we can rewrite the model as

$$(1-0.5B)(1-0.3B)X_t = (1-0.3B)a_t$$

Finally, canceling the (1 - 0.3B) from both sides leaves the simpler, but equivalent, model

$$(1-0.5B)X_t = a_t$$

$$X_t = 0.5X_{t-1} + a_t$$

Please keep in mind that this is a much simpler model! Experienced Box-Jenkins forcasters employ this type of model rearrangement to get the simplest models feasible. The roots of the two polynomials, p (B) & q (B), are displayed in the Theoretical ARIMA programme, allowing you to see various model simplifications.

Nonstationary Models

Nonstationary behaviour is seen in many time series in practise. Nonstationarity is usually caused by a trend, a shift in the local mean, or seasonal variation. We must make certain adjustments before we can model these nonstationary series because the Box-Jenkins approach is only for stationary models. To convert a nonstationary series with trend to a stationary series (without trend), we utilise one of two methods:

1. Use Wt = Xt Xt1 as the first difference in the series. It's worth noting that Wt = (B)Xt 1 can be rewritten. This equation in a more general form:

$$\phi_p(B)(1-B)^d X_t = \theta_q(B)a_t$$

where d is the differencing order The ARIMA(p,d,q) model is used to describe this. 2. Fit the residuals to a least squares trend & the Box-Jenkins model. First differences will result in a stationary model if the model has an infrequent change of mean.

Seasonal Time Series

To deal with series containing seasonal fluctuations, Box-Jenkins recommend the following general model:

$$\phi_p(B)\Phi_P(B)(1-B)^d\left(1-B^s\right)^D X_t = \theta_q(B)\,\Theta_Q\left(B^s\right)a_t$$

where d the order of differencing, s the number of seasons per year, & D the order of seasonal differencing. The operator polynomials are

$$\phi_p(B) = \left(1 - \phi_1 B - \cdots - \phi_p B^p\right)$$

$$\theta_q(B) = \left(1 - \theta_1 B - \cdots - \theta_q B^q\right)$$

$$\Phi_P\left(B^s\right) = \left(1 - \Phi_1 B^s - \cdots - \Phi_p B^{sp}\right)$$

$$\Theta_Q\left(B^s\right) = \left(1 - \Theta_1 B^s - \cdots - \Theta_Q B^{sQ}\right)$$

Note that $(1- B) X = X - X - s t t t s$. Box-Jenkins explain that the maximum value of d, D, p, q, P, & Q is two. Hence, these operator polynomials are usually simple expressions.

Partial Autocorrelation Function

The autocorrelation function, which reveals the correlations between different lags of a series, was previously addressed. A second function, the Partial Autocorrelation Function, expresses information useful in establishing the order of an ARIMA model. This function was created by calculating the partial correlation between Xt & Xt1, Xt & Xt2, & so on, while statistically compensating for intermediate lags. For example, after statistically removing the influence of Xt1, Xt2, & Xt3 from both Xt & Xt4, the partial autocorrelation of lag four is the partial correlation between Xt & Xt4. The lag of the latest large partial autocorrelation is used to calculate the autoregressive order, p.

Consider the case when the partial autocorrelations were

Lag	Partial Autocorrelation
1	0.55
2	0.21
3	0.11
4	0.72
5	0.06
6	0.09
7	0.13

We would conclude that a reasonable value for p is four, since the partial autocorrelations are relatively small after the fourth lag

The Box-Jenkins method refers to the iterative application of the following three steps:

1.Identification. A class of basic ARIMA models was chosen based on data plots, autocorrelations, partial autocorrelations, & other information. This entails estimating acceptable p, d, & q values.

2.Estimation. Maximum likelihood approaches, backcasting, & other methods are used to estimate the phis & thetas of the chosen model, as discussed in Box-Jenkins (1976).

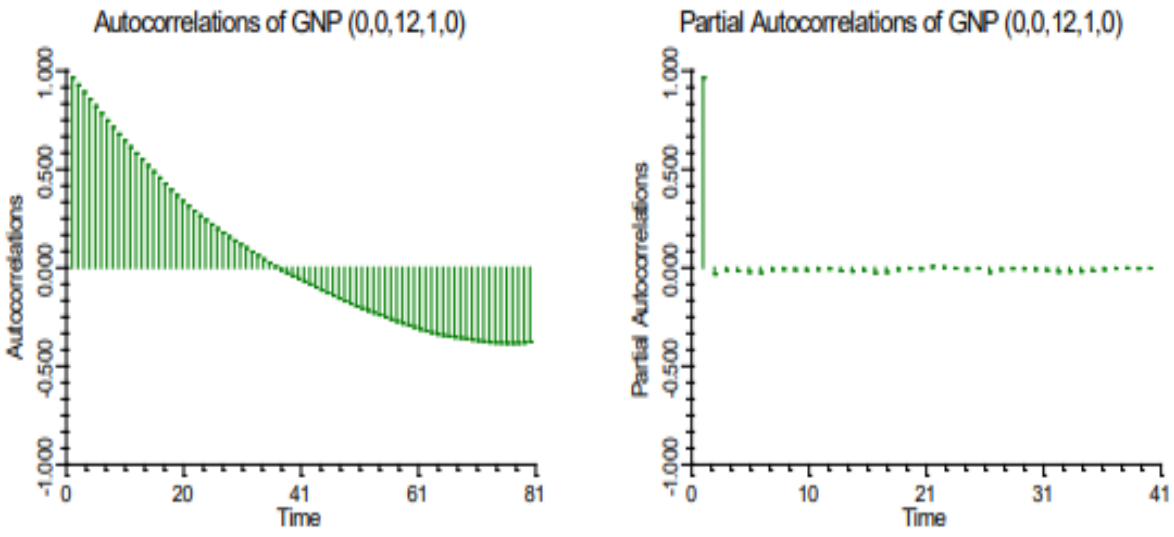
3.Confirming the diagnosis The residual series' autocorrelations were used to check for flaws in the fitted model (the series of residual, or error, values).

These steps are repeated repeatedly until the model does not improve after step three. We'll go through each step in detail now.

Model Identification

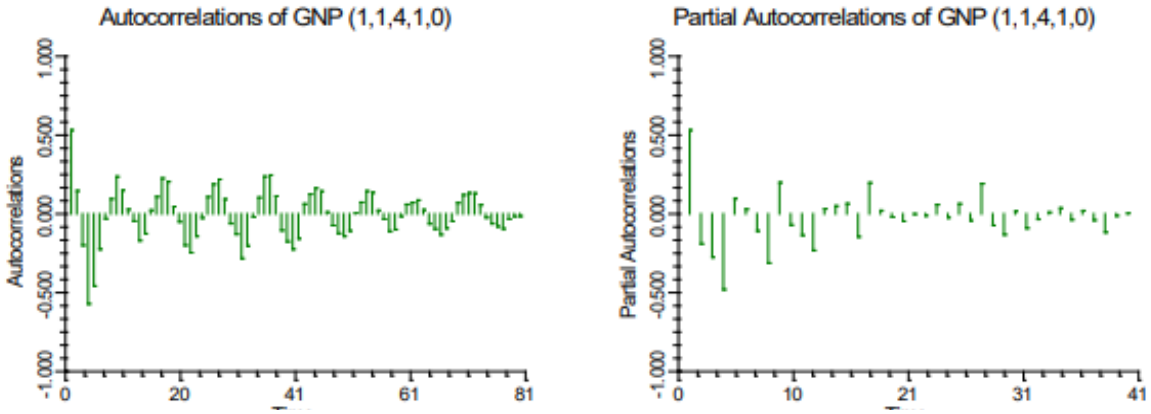
Assuming that there is no seasonal variation, the model identification step's goal is to choose values for d, then p, & finally q in the ARIMA(p,d,q) model. We can either fit & eliminate a deterministic trend or difference the series when it shows a trend. Differentiating appears to be preferred by Box-Jenkins, although deterministic trend elimination is preferred by numerous other writers. In any scenario, the initial step is to examine the plots of autocorrelations & partial autocorrelations. A trending series will have autocorrelation patterns that look like this.

.



We notice that the large autocorrelations persist even after several lags. This indicates that either a trend should be removed or that the series should be differenced. The next step would be to difference the series.

When the series is differenced, the autocorrelation plots might appear as follows:



Differencing generally significantly lowers the amount of big autocorrelations. If the differenced series does not appear to be stationary, we will have to difference it once more. The size of a big autocorrelation & partial autocorrelation coefficient is frequently beneficial. To be statistically significant, an autocorrelation must be at least 2 / N in absolute value. The table below shows several frequent significant autocorrelation values for various sample sizes. Even if an autocorrelation is statistically significant, it might not be substantial enough to cause concern.

N	Large Autocorrelation
25	0.40
50	0.28
75	0.23
100	0.23
200	0.14
500	0.09
1000	0.06

By considering the patterns of the autocorrelations & the partial autocorrelations, we can guess a reasonable model for the data. The following chart shows the autocorrelation patterns that are produced by various types of ARMA models.

Model	Autocorrelations	Partial Autocorrelations
ARIMA(p,d,0)	Infinite. Tails off.	Finite. Cuts off after *p* lags.
ARIMA(0,d,q)	Finite. Cuts off after *q* lags.	Infinite. Tails off.
ARIMA(p,d,q)	Infinite. Tails off.	Infinite. Tails off.

The identification phase determines the values of d (differencing), p (autoregressive order), & q (moving average order). By studying the two autocorrelation plots, you estimate these values.

Differencing

The autocorrelation plots are used to measure the amount of differencing. The suitable value of d has been established when the autocorrelations die off soon.

p's value

The partial autocorrelations of the correctly differenced series were used to calculate the value of p. The projected value of p would be the final lag with a big value if the partial autocorrelations broke off after a few lags. You have a moving average model (p=0) or an ARIMA model with positive p & q if the partial autocorrelations do not cut off.

q's value

The autocorrelations of the correctly differenced series yielded the value of q.

If the autocorrelations cut off after a few lags, the last lag with a large value would be the estimated value of q. If the autocorrelations do not cut off, you either have an autoregressive model (q=0) or an ARIMA model with a positive p & q.

Mixed Model

A mixed model was suggested when neither the autocorrelations nor the partial autocorrelations cut off. After the first q-p delays, the autocorrelation function in an ARIMA(p,d,q) model will be a mixture of exponential decay & damped sine waves. After p-q delays, the partial autocorrelation function has the same pattern. You might be able to figure out p & q by looking at the first few correlations in each graphic.

Directly finding the values of p & q in mixed models has proven problematic in our experience. Instead, we employ a trial-and-error method in which we fit more complicated models until the residuals reveal no more structure (large autocorrelations). Typically, we fit an ARIMA(1,d,0), an ARIMA(2,d,1), & an ARMA(1,d,1) (4,3).). We'd go with the simplest model that matched us quite well. (We normally start with the ARIMA(2,d,1) because it frequently works well.) A seasonal series is far more difficult to identify. Box-Jenkins describes model identification procedures, however to effectively identify the model order, the user must be extremely knowledgeable & experienced. We've discovered that trial & error is frequently required. Typically, you want to limit the number of parameters to a minimum, so the values you choose for p, P, q, Q, d, & D should be fewer than or equal to two. The identification stage, as you can see, is subjective.

One of the most common criticisms of the Box-Jenkins approach is that, although using the same software, two trained forcasters will come at different forcasting models. As we have demonstrated, models that appear to be extremely different on the surface are often surprisingly similar.

Model Estimation & Diagnostic Checking

Maximum Likelihood Estimation

You're ready to estimate the phis & thetas once you've calculated the values of p, d, & q. This software follows the Box-Jenkins maximum likelihood estimation technique (1976). Nonlinear

function maximization is used to solve the greatest likelihood problem. Estimates of the original residuals are obtained through backcasting. Because the estimating procedure is calculation-intensive & iterative, obtaining a solution might take a few seconds.

Diagnostic Checking

The diagnostic examination of the model comes after it has been fitted. The verification is done by looking at the residual autocorrelation plots to determine whether there is any additional structure (high correlation values). The model is deemed acceptable & forecasts are created if all autocorrelations & partial autocorrelations are modest. The values of p &/or q are modified & the model is re-estimated if some of the autocorrelations are high.

This procedure of reviewing the residuals & modifying the p & q values continues until the residuals have no more structure. The application may be used to create forecasts & related probability limitations after an appropriate model has been chosen.

CHAPTER -3

SYSTEM DEVELOPMENT

3.1 Data Formatting

Data from a shampoo manufacturer was gathered through a Kaggle sales forecasting competition. To examine how the data was formatted. Since the data comes from the shampoo firm, rather than just one store, there is sales data from numerous outlets throughout a three-year period. Holiday data was not included. The dataset contains almost 4,000 goods; however, the 10 most popular products were utilized to analyze the models for simplicity. For each of the most common goods, there were over 80 000 rows of data. All sales from each retailer were totalled up for each date for each unique product to form a time-series. Furthermore, only the sales parameter was used as a feature in the models. The data can be expressed as [s0, ..., st] where s the sales for a product & the subscript denotes which day, t the total amount of days logged. To be able to utilize the data for training, it had to be split up into time windows as follows:

$$
\begin{gathered}
\left[s_0, \ldots, s_d, s_{d+1}, \ldots, s_{d+p}\right] \\
\left[s_1, \ldots, s_{d+1}, s_{d+2}, \ldots, s_{d+p+1}\right] \\
\left[s_2, \ldots, s_{d+2}, s_{d+3}, \ldots, s_{d+p+2}\right] \\
\vdots \\
\left[s_{t-d-p}, \ldots, s_{t-p-1}, s_{t-p}, \ldots, s_t\right]
\end{gathered}
$$

Where d is the amount of prior time steps that the model considers while predicting p future time steps. Each row's first d columns are utilized for training, while the remaining columns are labeled for each matching row.

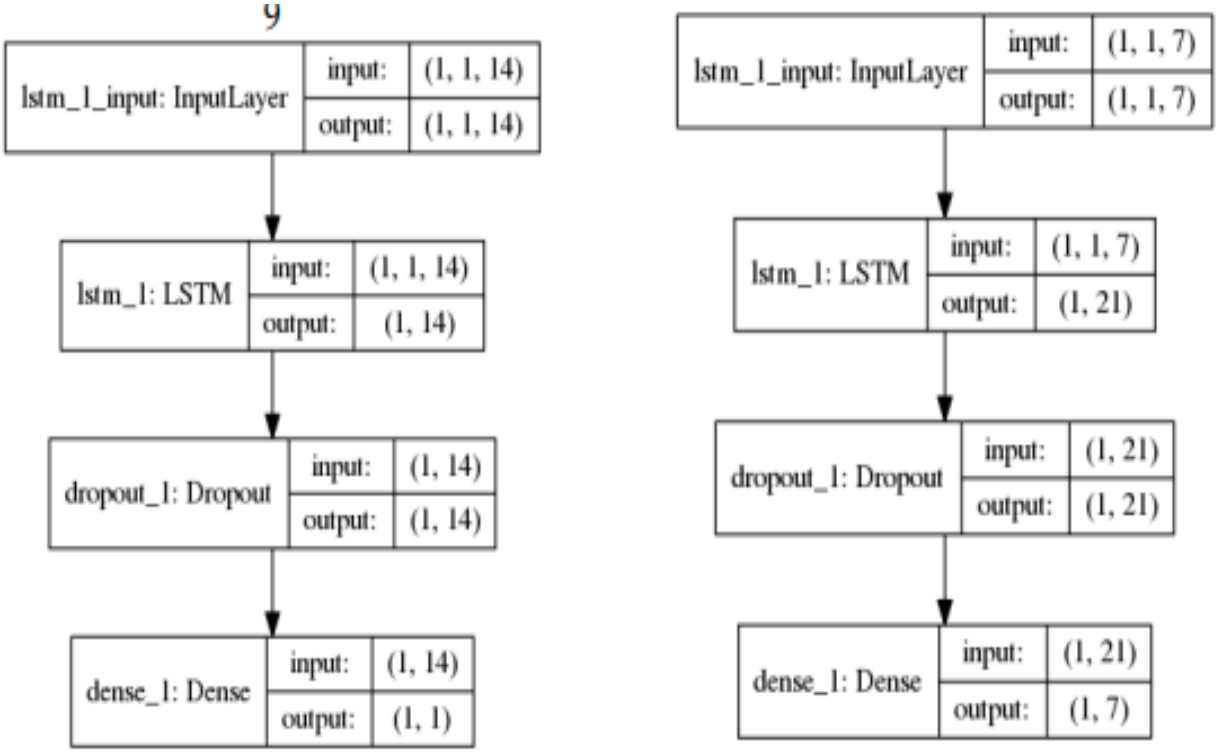
3.2 Model implementations

3.2.1 Baseline model

A naïve forcasting method was used to create one baseline model. The model basically examines each value in the time series & forecasts the same value for the next time step. The baseline model basically determines if the ARIMA & LSTM models can forecast more accurately than a naïve forcasting technique.

3.2.2 LSTM-implementations

Following the two cases previously described, two LSTM models were implemented. LSTM1 & LSTM7 are two models that anticipate one & seven days ahead, respectively. One projected only one day ahead, while the other anticipated the next seven days. Keras, a Python library, was used to implement both models. Both models used 80% of the data as training data, while the remaining 20% was used as the test set. The information was also standardized & divided..



Model summary (Figure 3): The layers & input/output-shapes in the network are depicted in the diagram. The greatest results were obtained when only one LSTMlayer was used. The left model depicts the structure of LSTM1, whereas the right model depicts LSTM7, as evidenced by the last output layer.

3.2.3ARIMA-implementation

The statsmodel library was used in Python for the ARIMA implementation. Similarly to the LSTM-model, 80% of the data was utilized for training & 20% for testing. The lag-order was set to 7 days, same as the previous days in the LSTM-model for each prediction, the degree of differencing was set to 1, & the moving average was set to 0. Both forecast scenarios employed the same model.

3.2.4 Hyperparameters

A grid search was used to determine which variables to utilize as hyperparameters - the values that personalize the models. Grid search is a well-known method for improving hyperparameters in machine learning models. [24] It's a type of exhaustive search in which a huge number of hyperparameters are examined & the ones that produce the best results are incorporated in the final

model. Because the MAE- & RMSE error measurements were both used in this test, the analysis had to take both into account when determining the best values. As a consequence, the best 10 RMSE & MAE results were picked & compared. The following values were chosen using the grid search:

LSTM1 parameter values The factors that affect the training include sequence length, which determines how long the LSTM technique should recall information, dropout, which prevents overfitting, & parameters that influence the training, as shown in table 3.1.

	chosen value	**grid search interval**
Sequence Length	14	[1, 7, ... , 28]
Dropout	0.2	[0.1, 0.2, ... , 1.0]
Epochs	15	[1, 5, ... , 50]
Neurons output layer	14	[1, 7, ... , 28]
Activation Function	tanh	[relu, tanh, sigmoid]
Optimization Function	adam	[adagrad, adam, RMSprop]

Table 3.1: Values used for the respective parameters in the $LSTM_1$ model as well as the grid search domain for each parameter is presented.

Parameters for $LSTM_7$

The values used for the $LSTM_7$ model can be seen in table 3.2.

	chosen value	grid search interval
Sequence Length	7	[1, 7, ... , 28]
Dropout	0.3	[0.1, 0.2, ... , 1.0]
Epochs	25	[1, 5, ... , 50]
Neurons output layer	21	[1, 7, ... , 28]
Activation Function	tanh	[relu, tanh, sigmoid]
Optimization Function	adam	[adagrad, adam, RMSprop]

Table 3.2: Values used for the respective parameters in the $LSTM_7$ model as well as the grid search domain for each parameter is presented.

Parameters for ARIMA

The values used for the ARIMA model can be seen in table 3.3.

	chosen value	grid search interval
P	7	[1, 7, ... , 28]
D	1	[1, 2, 3]
Q	0	[0, 1, 2]

Table 3.3: Values used for the respective parameters in the ARIMA model as well as the grid search domain for each parameter is presented.

3.3 EVALUATION MERSURES

Two separate assessment measures were utilized to compare the performance of LSTM & ARIMA: mean absolute error (MAE) & root mean square error (RMSE) (RMSE). Lower values for both metrics indicate more precision. RMSE & MAE may be defined as follows, with Ft as the forecast value (prediction), At as the actual value, & n as the number of time steps:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(A_t - F_t)^2}{n}}$$

$$\text{MAE} = \frac{\sum_{t=1}^{n}|A_t - F_t|}{n}$$

Previous research ([25], [26]) has utilized the assessment metrics to assess the prediction performance of several models, including LSTM & ARIMA [10], [20], suggesting that they are valid.

3.4 T-TEST

Four t-tests were used to confirm the findings. The t-test [23] was used to compare the means of two populations, in this case the LSTM & ARIMA models. For each prediction model, a t-test was run for both the 1-dayahead & 7-dayahead forecast scenarios. H0 was that the LSTM model has a lower prediction error than the ARIMA model, while Ha was that the LSTM model has a prediction error i.e equal to or larger than the ARIMA model. The same 10 goods were used to generate data for each t-test. Table 3.4 shows the null hypothesis for each of the four t-tests performed..

$\mathbf{H_0}$	**Evaluation Measure**
$\text{LSTM}_1 < \text{ARIMA}_1$	RMSE
$\text{LSTM}_7 < \text{ARIMA}_7$	RMSE
$\text{LSTM}_1 < \text{ARIMA}_1$	MAE
$\text{LSTM}_7 < \text{ARIMA}_7$	MAE

Table 3.4: The null hypothesis for each t-test performed on the different scenarios and the evaluation measurement used in the particular t-test

CHAPTER 4

PERFORMANCE ANALYSIS

Over a three-year period, this dataset depicts the monthly number of shampoo sales.

There are 36 observations & the units represent a sales count. Makridakis, Wheelwright, & Hyndman are the authors of the original dataset (1998).

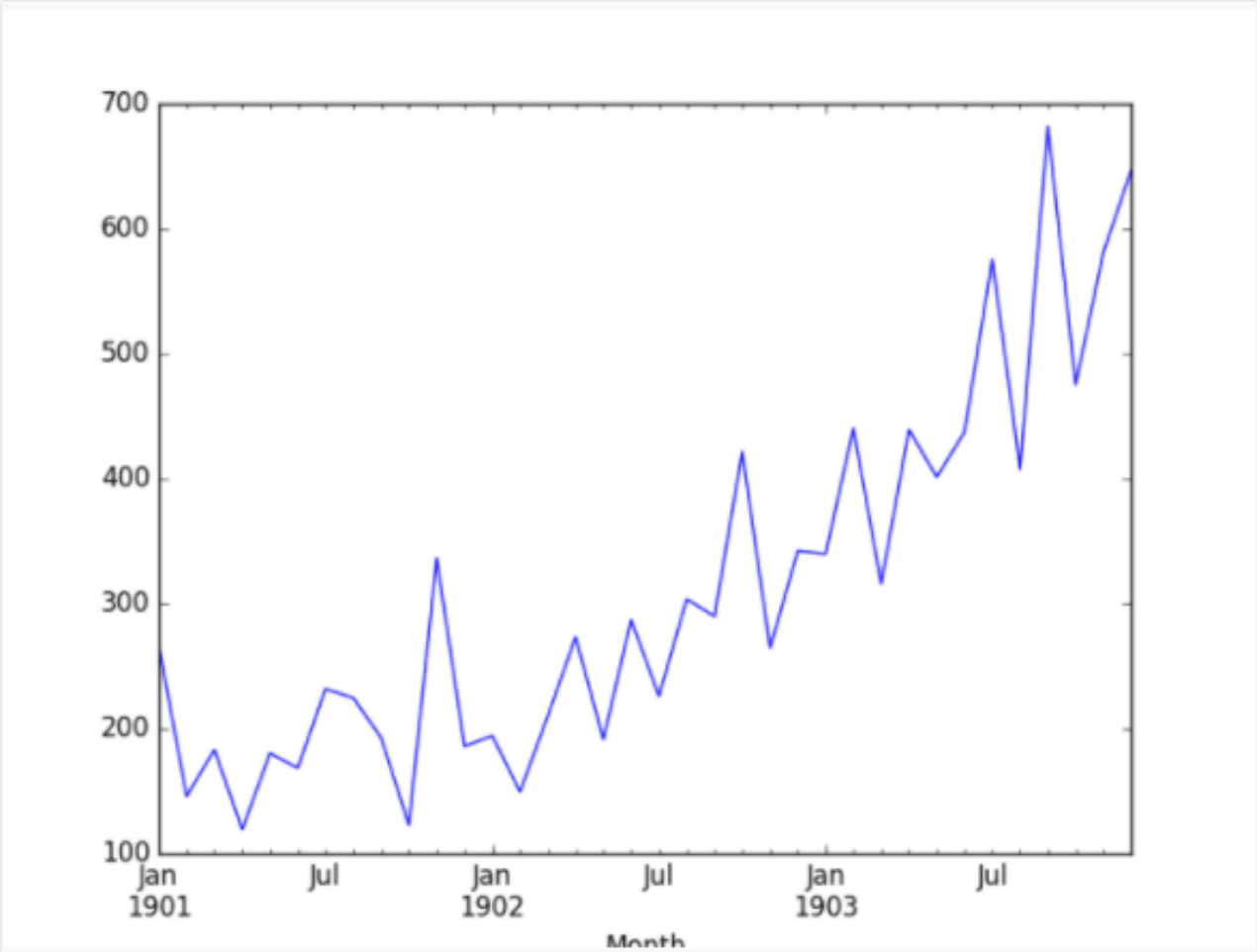
Using Pandas to import the Shampoo Sales information & a custom function to parse the date-time field. The dataset is fixed to a certain year, in this example 1900.

```
from pandas import read_csv
from pandas import datetime
from matplotlib import pyplot

def parser(x):
    return datetime.strptime('190'+x, '%Y-%m')

series = read_csv('shampoo-sales.csv', header=0, parse_dates=[0], index_col=0, squeeze=True, date_parser=par
print(series.head())
series.plot()
pyplot.show()
```

Running the example prints the first 5 rows of the dataset.

```
Month
1901-01-01 266.0
1901-02-01 145.9
1901-03-01 183.1
1901-04-01 119.3
1901-05-01 180.3
Name: Sales, dtype: float64
```

The data is also plotted as a time series with the month along the x-axis and sales figures on the y-axis.



Shampoo Sales Dataset Plot

We can see that the Shampoo Sales dataset has a clear trend.

This indicates that the time series is not stationary & will require differencing (at least a difference order of 1) to make it stationary.

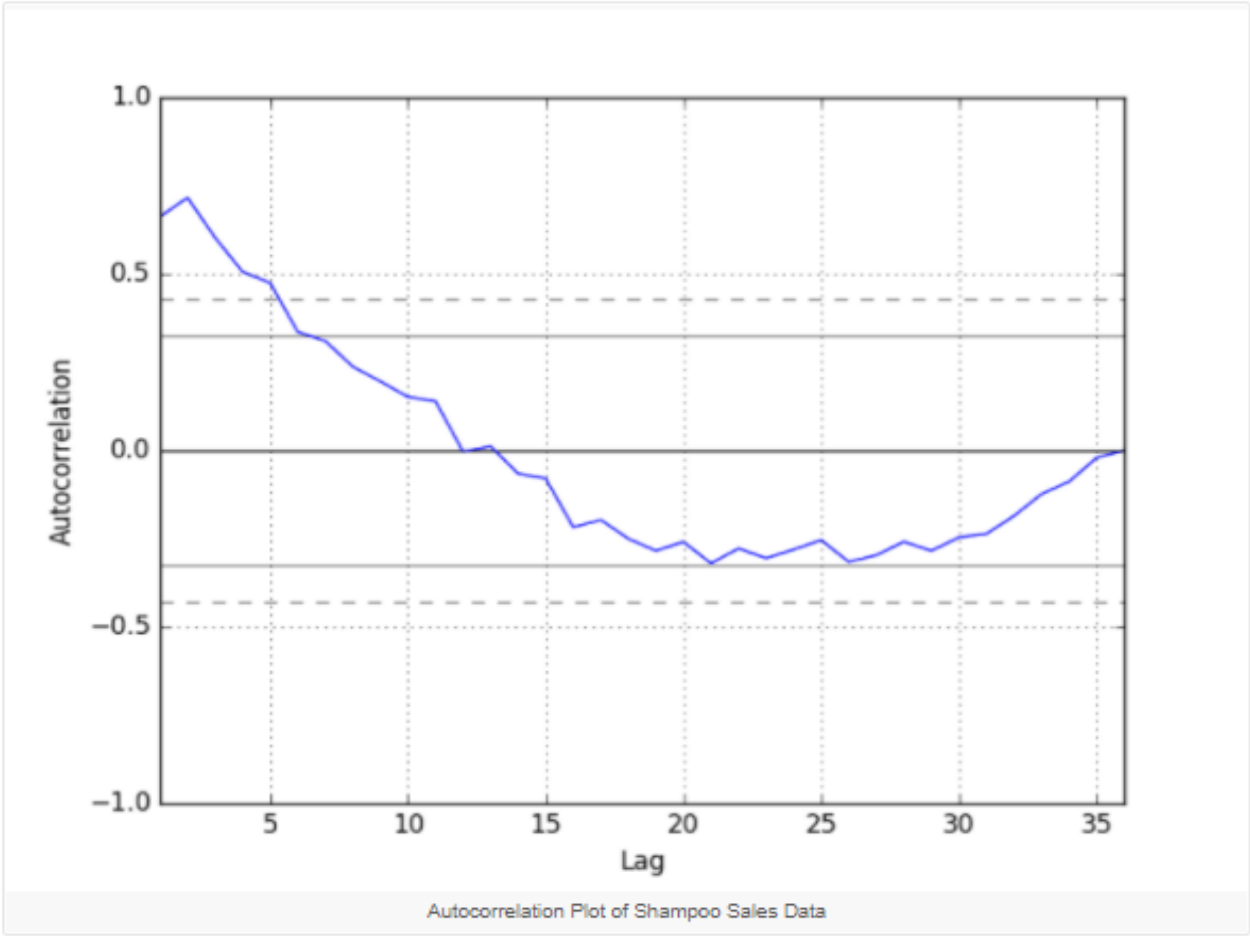
Let's take a brief glance at the time series' autocorrelation plot. Pandas have this as well. The autocorrelation for a high number of lags in the time series is plotted in the example below.

```
from pandas import read_csv
from pandas import datetime
from matplotlib import pyplot
from pandas.plotting import autocorrelation_plot

def parser(x):
    return datetime.strptime('190'+x, '%Y-%m')

series = read_csv('shampoo-sales.csv', header=0, parse_dates=[0], index_col=0, squeeze=True, date_parser=pars
autocorrelation_plot(series)
pyplot.show()
```

Running the example, we can see that there is a positive correlation with the first 10-to-12 lags that is perhaps significant for the first 5 lags.

A good starting point for the AR parameter of the model may be 5.

A good starting point for the AR parameter of the model may be 5.



Autocorrelation Plot of Shampoo Sales Data

Fitting an ARIMA model is possible with the statsmodels package.

The statsmodels package may be used to generate an ARIMA model as follows:

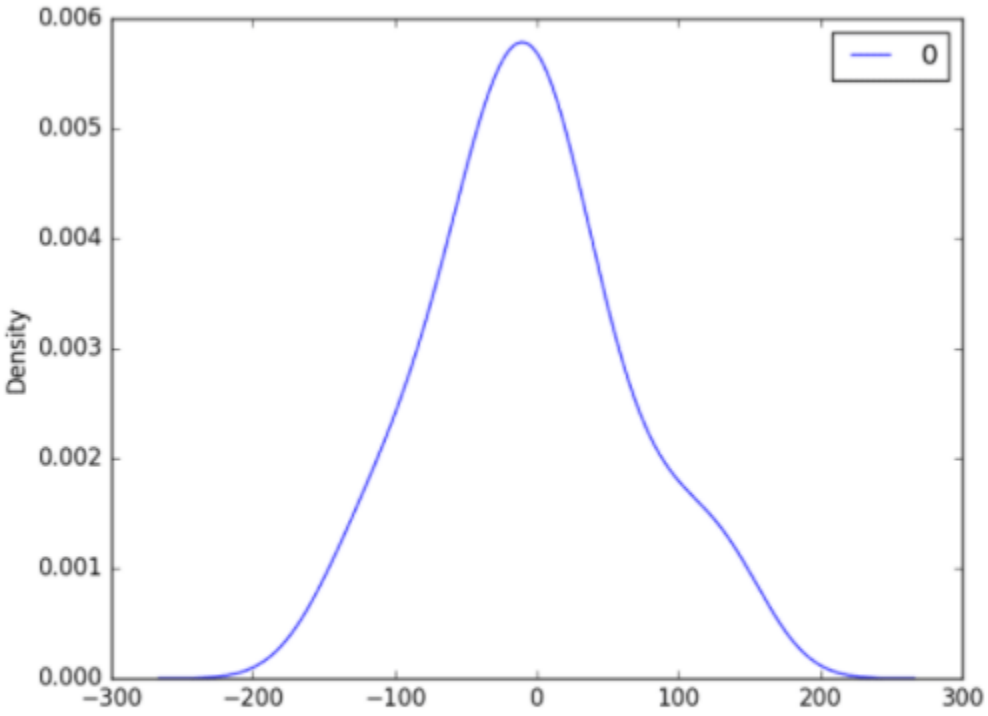
Call ARIMA() & pass in the p, d, & q parameters to define the model.

The fit() method was used to prepare the model on the training data.

Call the predict() method & give the index of the time or times to be forecasted to make predictions.

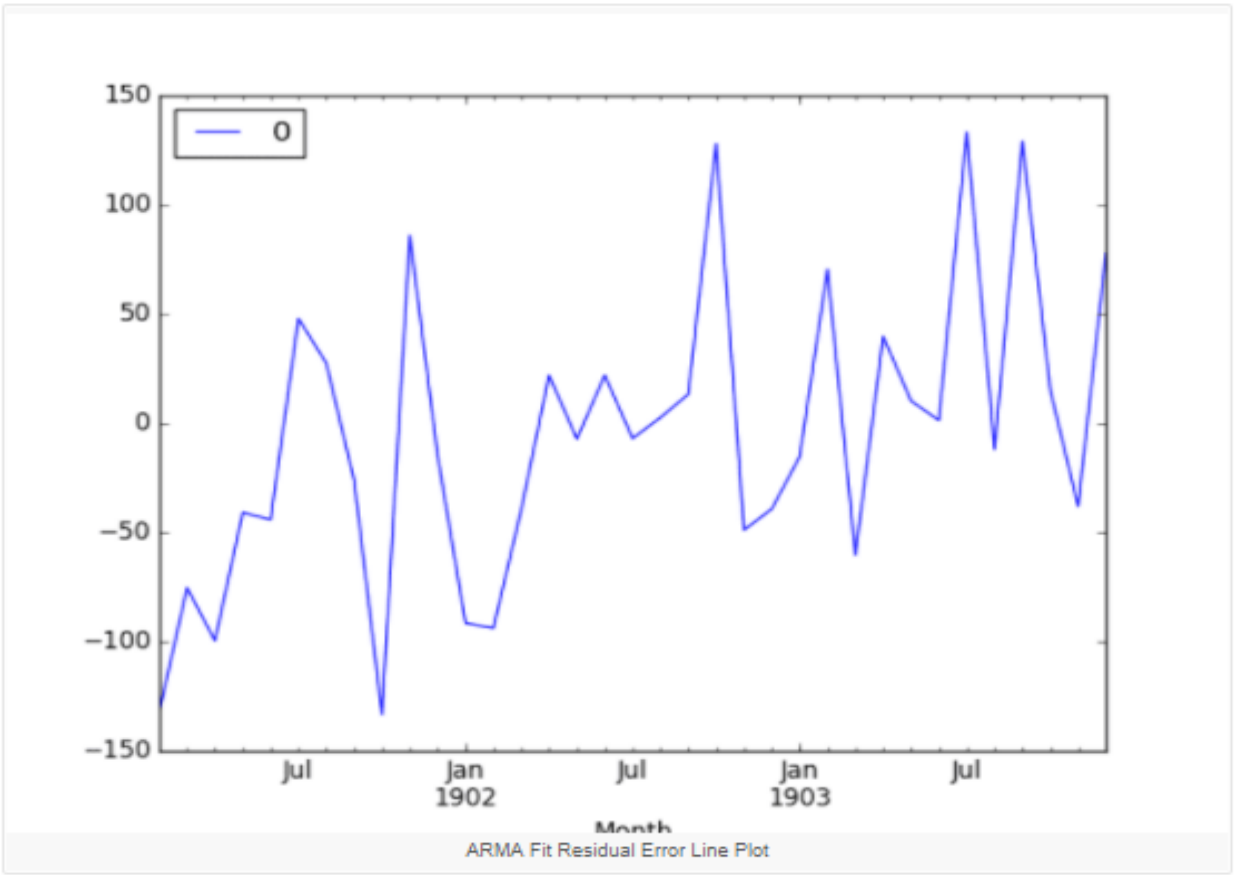
Let's start with something straightforward. We'll run the whole Shampoo Sales dataset through an ARIMA model & look at the residual errors.

We started by fitting an ARIMA(5,1,0) model. The lag value for autoregression is set to 5, the difference order is set to 1 & time stationary.



ARMA Fit Residual Error Density Plot

The distribution of the residual errors is displayed. The results show that indeed there is a bias in the prediction (a non-zero mean in the residuals).



ARMA Fit Residual Error Line Plot

Next, we get a density plot of the residual error values, suggesting the errors are Gaussian, but may not be centered on zero.

```
count   36.000000
mean    21.936144
std     80.774430
min   -122.292030
25%    -35.040859
50%     13.147219
75%     68.848286
max    266.000000
```

Note, that although above we used the entire dataset for time series analysis, ideally we would perform this analysis on just the training dataset when developing a predictive model.

Next, let's look at how we can use the ARIMA model to make forecasts.

Rolling Forcast ARIMA Model

Future time steps can be predicted using the ARIMA model.
To produce predictions, we may utilize the predict() function on the ARIMAResults object. It accepts the index of the time steps as arguments for making predictions. These indices refer to the beginning of the training dataset that was used to create predictions.
The index of the next time step for generating a prediction would be supplied to the prediction function as start=101, end=101 if we utilized 100 observations in the training dataset to fit the model. This would produce an array with the prediction as the first entry.

In the event that we conducted any differencing (d>0 when configuring the model), we'd also want the projected values to remain in the original scale.
Alternatively, we may use the forcast() method to generate a one-step forecast using the model, avoiding all of these parameters.
We can divide the training dataset into train & test sets, fit the model with the train set, then create predictions for each element on the test set using the train set.
Given the AR model's reliance on data from previous time steps, a rolling forecast is necessary. Re-creating the ARIMA model after each new observation received is a rudimentary technique to do this rolling forecast.

We manually maintain track of all observations in a history list, which is seeded with the training data & to which additional observations are added each cycle.

Putting this all together, below is an example of a rolling forecast with the ARIMA model in Python.

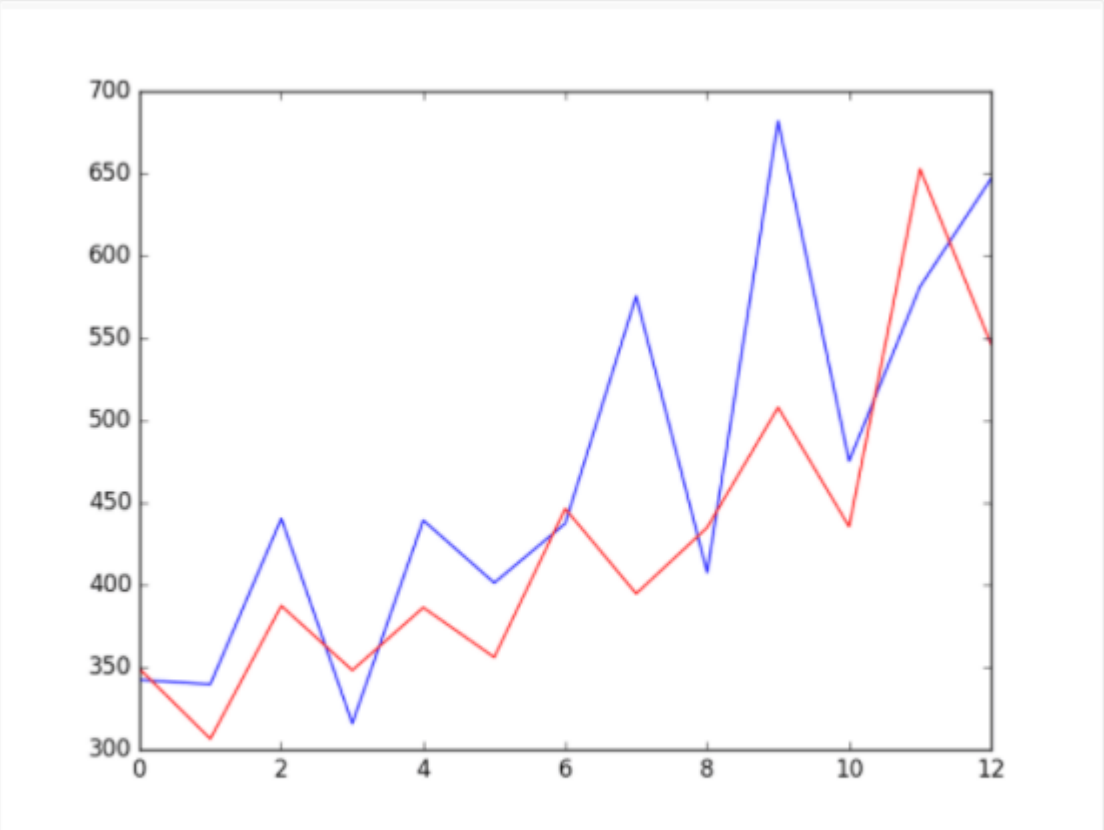
```
# evaluate an ARIMA model using a walk-forward validation
from pandas import read_csv
from pandas import datetime
from matplotlib import pyplot
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
from math import sqrt
# load dataset
def parser(x):
    return datetime.strptime('190'+x, '%Y-%m')
series = read_csv('shampoo-sales.csv', header=0, index_col=0, parse_dates=True, squeeze=True, dat
series.index = series.index.to_period('M')
# split into train and test sets
X = series.values
size = int(len(X) * 0.66)
train, test = X[0:size], X[size:len(X)]
history = [x for x in train]
predictions = list()
# walk-forward validation
for t in range(len(test)):
    model = ARIMA(history, order=(5,1,0))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
    print('predicted=%f, expected=%f' % (yhat, obs))
# evaluate forecasts
rmse = sqrt(mean_squared_error(test, predictions))
print('Test RMSE: %.3f' % rmse)
# plot forecasts against actual outcomes
pyplot.plot(test)
pyplot.plot(predictions, color='red')
pyplot.show()
```

Running the example prints the prediction and expected value each iteration.

We can also calculate a final root mean squared error score (RMSE) for the predictions, providing a point of comparison for other ARIMA configurations.

```
predicted=343.272180, expected=342.300000
predicted=293.329674, expected=339.700000
predicted=368.668956, expected=440.400000
predicted=335.044741, expected=315.900000
predicted=363.220221, expected=439.300000
predicted=357.645324, expected=401.300000
predicted=443.047835, expected=437.400000
predicted=378.365674, expected=575.500000
predicted=459.415021, expected=407.600000
predicted=526.890876, expected=682.000000
predicted=457.231275, expected=475.300000
predicted=672.914944, expected=581.300000
predicted=531.541449, expected=646.900000
Test RMSE: 89.021
```

A line plot is created showing the expected values (blue) compared to the rolling forecast predictions (red). We can see the values show some trend and are in the correct scale.



ARIMA Rolling Forecast Line Plot

The model could use further tuning of the p, d, and maybe even the q parameters.

Configuring an ARIMA Model

The Box-Jenkins Methodology is the traditional approach for fitting an ARIMA model.

This is a method for finding suitable ARIMA model parameters using time series analysis & diagnostics.

Model recognition. To estimate the amount of differencing & the magnitude of the lag that will be required, use plots & summary statistics to detect trends, seasonality, & autoregression features.

Estimation of parameters. To get the regression model's coefficients, use a fitting process.

Model verification. Determine the quantity & kind of temporal structure not represented by the model using graphs & statistical tests of residual errors.

The technique was repeated until the in-sample or out-of-sample observations had a satisfactory degree of fit (e.g. training or test datasets).
The procedure was outlined in George Box & Gwilym Jenkins' renowned 1970 textbook Time Series & Analyis: Forecasting & Control. If you're interested in learning more about this concept & technique, a new 5th edition is now available.
Grid searching parameters of the model can be a useful strategy since the model can be fit effectively on small time series datasets..

CHAPTER 5

CONCLUSION

We can plainly see that the model chosen can be used to model & forecast future sales in this shampoo manufacturing, but we must continually fill the historical data with fresh data in order to enhance the new model & forecasting. The projections provided through modeling helped this shampoo manufacturer make production decisions. In reality, the model allowed us to accurately estimate sales & make projections. Once we have a sales prediction, it will be much easier & clearer to design the appropriate production & therefore avoid large cost losses. This will assist us in making the best selections about raw material supply & daily production determination. Moreover, that will affect the whole production process eliminating then any kind of loss..

Forecasting sales is an important aspect of supply chain management. Because of its connection with other business operations, it is one of the most significant planning procedures a company may use in the future. Using the Box– Jenkins time series technique, we developed an ARIMA model to simulate the sales forecasting of the completed product in a shampoo manufacturing. Several models were developed using historical sales data, & the best one was chosen based on four performance criteria: SBC, AIC, standard error, & maximum likelihood. ARIMA is the model that we choose to minimize the four prior criteria (1, 0, 1).

The acquired findings demonstrate that this model may be utilized to predict & anticipate future sales in this shampoo production; these results will give managers with trustworthy decision-making recommendations. We will continue to create new models that combine qualitative & quantitative methodologies to make credible forecasts & improve forecast accuracy in the future. In order to confirm the ANN's strength in the shampoo industry, we will also test a neural network technique & compare it to ARIMA's findings.

In terms of both RMSE & MAE, the findings of total difference in error demonstrate that the LSTM-model appeared to have superior prediction accuracy than the ARIMA-model. The difference between the models in the one-day-ahead prediction scenario is not statistically significant, according to the t-test. The t-test only shows that the LSTM model is more accurate than the ARIMA model in the seven-days-ahead prediction scenario. Given that ARIMA is a commonly used state-of-the-art model, the LSTM-network displays promising results for sales prediction in the seven-days-ahead scenario, & is therefore considered to be a model that can compete with ARIMA.

5.1 Further Research

We can draw some inferences about things that need to be improved based on the conversation above. First & foremost, a more practical application of the concept appears appropriate in order to provide an underlying incentive for the real study. We feel that grocery shops do not order food on a daily basis, therefore developing a model that forecasts sales for two weeks/a month ahead should be a good place to start. As previously stated, this would also help to clarify if LSTM is preferable to ARIMA for more complicated models. Even if it is our own forecast, we cannot guarantee that the findings produced for our core models will be consistent when enlarged. Furthermore, hyperparameter fine tuning may be enhanced. Building the LSTM within Tensorflow, as well as implementing the ARIMA-model in a more precise manner, should be studied for greater control & fine-tuning.

5.2 Result Discussion

We predicted that the non-linear LSTM model would beat the linear ARIMA model in both instances due to the problem's complexity, as non-linear models are believed to provide superior accuracy for difficult issues. The one-day-ahead prediction scenario results were not statistically significant according to the t-test, despite the fact that the findings suggest that LSTM may be better. Future study might potentially rule out if LSTM is better at handling complexity, as recent research shows, by increasing prediction length & complexity. It didn't appear to matter how many prior values were examined & utilized as input for the LSTM..

We tested a sequence length of 7-28 days in the hopes that the LSTM would recognize the idea of weeks, but the prediction accuracy did not seem to improve. In our parameter grid search, we discovered that models with sequence lengths of 7, 14, & 21 were in the top 10. In retrospect, because we don't designate each weekday (added as a feature, for example), it was probably difficult for the model to grasp the idea of weeks, especially given the low datasets. Our findings are consistent with earlier studies comparing LSTM & ARIMA for time-series forecasting challenges, at least for the seven-days ahead prediction scenario, are also in agreement with previous research also comparing LSTM & ARIMA for time-series forecasting problems. In they compared ARIMA & LSTM on four datasets & their RMSE values favored LSTM in three out of the four datasets.

5.3 Limitations & Relevance

A sufficient amount of data is required for the network to be successfully trained & evaluated [5]. The results may have been influenced by the fact that the data was confined to only one supermarket chain over a four-year period. It would have been preferable to have access to sales data from numerous chains & locations over a longer period of time in order to create a more diversified training dataset & test dataset for the network. Furthermore, having additional data would have allowed for further experimentation, resulting in results that may have improved the study's credibility or highlighted methodological flaws. For the best prediction accuracy, each machine learning issue uses distinct hyper parameters.even if the underlying model the same. Given this, determining the appropriate settings may prove challenging. Another option is to draw inspiration from other models in comparable contexts, such as time-series forecasting in our instance. However, because we constructed a basic model, it was difficult to draw inspiration from the more complex models in the research articles we looked at. Instead, we used a manual grid search, which turned out to be wasteful in terms of time. There are libraries like Hyperas or Sklearn that can accomplish this considerably more quickly, but we learned this too late in our project. However, the grid search produced results, although they were fairly coarse & might have been fine-tuned if time had permitted.

Because our understanding of LSTM implementation was restricted, we chose a higher abstraction framework - Keras - to make the LSTM network construction easier. However, because Keras has a greater degree of abstraction, having complete control of the network becomes more difficult. Even while we gain from the absence of essential knowledge, the lack of control might lead to ambiguity in the results since it's unknown what sort of network is being constructed in the background. However, the models projected better than the baseline model, which is at least reassuring.

The LSTM-model has received the most attention, whereas the ARIMA-model has received less. Both in terms of understanding how it works & putting it into practice. In comparison to ARIMA, the LSTM model offers far more fine-tuning choices. At the very least, we can use the libraries we have. This might cast doubt on the ARIMA findings. Both models, however, are constructed using high-abstraction libraries & may certainly be much improved. Preprocessing the data differently might have possibly improved the ARIMA results. For example, aggregating revenues across weeks rather than days. It appears that a discussion of the model's real usefulness is also required.

Our ultimate objective was to create a model that may assist a retailer in placing more precise food orders, hence reducing food waste & empty shelves. However, this concept has no real-world applicability. Given delivery timeframes, the algorithm could never be utilized in a real business because it can only anticipate sales one to seven days ahead. However, with future development, this might be increased.

Furthermore, the models only cover one product & are not extended to other items, which might hamper usability because a more generalized prediction model may necessitate a more sophisticated system [8]. Finally, we'd like to develop a model that allows users to enter a product ID & receive sales forecasts for that product. The result that the LSTM-model predicts better than ARIMA-model in seven-day-ahead prediction scenario inspires additional study into field, despite model's lack of practical use.

REFRENCES

[1] A. Galatsidas, The guardian; sustainable development goals: Changing the world in 17 steps – interactive, 2015. (visited on 01/01/2018).

[2] ICA.se, Delat ansvar för matsvinnet, 2017. (visited on 01/03/2018).

[3] P.-F. Pai & C.-S. Lin, "A hybrid arima & support vector machines model in stock price forecasting", Omega, vol. 33, no. 6, pp. 497–505, 2005, ISSN: 0305-0483.

[4] J. W. Taylor, P. E. McSharry, R. Buizza, et al., "Wind power density forecasting using ensemble predictions & time series models", IEEE Transactions on Energy Conversion, vol. 24, no. 3, p. 775, 2009.

[5] Q. Yu, K. Wang, J. O. Strandhagen, & Y. Wang, "Application of long shortterm memory neural network to sales forecasting in retail—a case study", in Advanced Manufacturing & Automation VII, K. Wang, Y. Wang, J. O. Strandhagen, & T. Yu, Eds., Singapore: Springer Singapore, 2018, pp. 11–17, ISBN: 978-981-10-5768-7.

[6] P. Doganis, A. Alexandridis, P. Patrinos, & H. Sarimveis, "Time series sales forecasting for short shelf-life food products based on artificial neural networks & evolutionary computing", Journal of Food Engineering, vol. 75, no. 2, pp. 196–204, 2006, ISSN: 0260-8774
.

[7] I. Khandelwal, R. Adhikari, & G. Verma, "Time series forecasting using hybrid arima & ann models based on dwt decomposition", Procedia Computer Science, vol. 48, pp. 173–179, 2015, International Conference on Computer, Communication & Convergence (ICCC 2015), ISSN: 1877-0509.

[8] S. Krstanovic & H. Paulheim, "Ensembles of recurrent neural networks for robust time series forecasting", in Artificial Intelligence XXXIV, M. Bramer & M. Petridis, Eds., Cham: Springer International Publishing, 2017, pp. 34– 46, ISBN: 978-3-319-71078-5

[9] L. Wang, Y. Zeng, & T. Chen, "Back propagation neural network with adaptive differential evolution algorithm for time series forecasting", Expert Systems with Applications, vol. 42, no. 2, pp. 855–863, 2015, ISSN: 0957- 4174.

[10] X. Li, L. Peng, X. Yao, S. Cui, Y. Hu, C. You, & T. Chi, "Long shortterm memory neural network for air pollutant concentration predictions: Method development & evaluation", Environmental Pollution, vol. 231, pp. 997–1004, 2017, ISSN: 0269-7491.

[11] J. Kumar, R. Goomer, & A. K. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters", Procedia Computer Science, vol. 125, pp. 676–682, 2018, The 6th International Conference on Smart Computing & Communications, ISSN: 1877-0509.

[12] kaggle.com, Corporación favorita shampoo sales forecasting, 2017. (visited on 03/12/2018). [13] C. Chatfield, Time-series forecasting. CRC Press, 2000.

[14] G. E. P. Box, G. M. Jenkins, & G. C. Reinsel, “Introduction”, in Time Series Analysis. Wiley-Blackwell, 2013, ch. 1, pp. 7–18, ISBN: 9781118619193. DOI: 10.1002/9781118619193.ch1.

[15] C. Chatfield, The analysis of time series: an introduction. CRC press, 2016.

[16] G. Zhang, “Time series forecasting using a hybrid arima & neural network model”, Neurocomputing, vol. 50, pp. 159–175, 2003, ISSN: 0925-2312.

[17] I. A. Gheyas & L. S. Smith, “A novel neural network ensemble architecture for time series forecasting”, Neurocomputing, vol. 74, no. 18, pp. 3855– 3864, 2011, ISSN: 0925-2312.

[18] Z.-L. Sun, T.-M. Choi, K.-F. Au, & Y. Yu, “Sales forecasting using extreme learning machine with applications in fashion retailing”, Decision Support Systems, vol. 46, no. 1, pp. 411–419, 2008, ISSN: 0167-9236.

[19] R. Rojas, Neural networks: a systematic introduction. Springer Science & Business Media, 2013.

[20] B. Cortez, B. Carrera, Y.-J. Kim, & J.-Y. Jung, “An architecture for emergency event prediction using lstm recurrent neural networks”, Expert Systems with Applications, vol. 97, pp. 315–324, 2018, ISSN: 0957-4174.

[21] S. Ji, H. Yu, Y. Guo, & Z. Zhang, “Research on sales forecasting based on arima & bp neural network combined model”, in Proceedings of the 2016 International Conference on Intelligent Information Processing, ser. ICIIP ’16, Wuhan, China: ACM, 2016, 41:1–41:6, ISBN: 978-1-4503-4799-0.

[22] C. I. Permatasari, W. Sutopo, & M. Hisjam, “Sales forecasting newspaper with arima: A case study”, AIP Conference Proceedings, vol. 1931, no. 1, p. 030 017, 2018. DOI: 10.1063/1.5024076.

[23] V. ¸S. Ediger & S. Akar, “Arima forecasting of primary energy demand by fuel in turkey”, Energy Policy, vol. 35, no. 3, pp. 1701–1708, 2007, ISSN: 0301-4215.

[24] J. Bergstra & Y. Bengio, “Random search for hyper-parameter optimization”, Journal of Machine Learning Research, vol. 13, no. Feb, pp. 281–305, 2012.

[25] M. Khashei & M. Bijari, “An artificial neural network (p,d,q) model for timeseries forecasting”, Expert Systems with Applications, vol. 37, no. 1, pp. 479– 489, 2010, ISSN: 0957-4174.

[26] N. S. Arunraj & D. Ahrens, “A hybrid seasonal autoregressive integrated moving average & quantile regression for daily food sales forecasting”, International Journal of Production Economics, vol. 170, pp. 321–335, 2015, ISSN: 0925-5273.

[27] J. Larsson, Hållbara konsumtionsmönster: Analyser av maten, flyget och den totala konsumtionens klimatpåverkan idag och 2050. Naturvårdsverket, 2015.