ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

Comparative Study of Machine Learning Algorithms for Sales Forecasting: Facebook Prophet vs. Established Statistical Models

Samir Kumar[a],
Suresh Kumar Garg[b],
Gaurav Mishra[c*]
[a, b, c] *Department of Mechanical, Production, and Industrial Engineering, Delhi Technological University New Delhi, India*

ABSTRACT

Purpose: Our primary objective was to compare the performance of Facebook Prophet, a popular ML algorithm, against established statistical models like SARIMA and Holt-Winters. This study focused on analysing time series sales data, acknowledging the significant influence of seasonality on sales patterns.

Design/Methodology/Approach: A Python script has been used as central tool, enabling the evaluation of various forecasting algorithms. This compared Facebook Prophet, SARIMA, and Holt-Winters based on their accuracy, robustness, and applicability to our data. This involved implementing each algorithm within the script and analysing their performance metrics, such as R-squared scores.

Findings: Our analysis revealed that Facebook Prophet emerged as the most accurate model for our dataset and chosen error metrics. It achieved an R-squared score of 0.91 for fitted data, demonstrating its strong ability to capture the underlying patterns in our sales data. While Holt-Winters performed competitively with an R-squared value of 0.87(Used Grid Search in Fine Tuning), 0.87(Used Simulated Annealing in fine tuning) and 0.85(Used Gradient Descent for fine tuning) the training set, it is important to note that these results might vary depending on the specific data and evaluation criteria. Interestingly, the remaining algorithms exhibited relatively minor performance differences, with SARIMA lagging behind at an R-squared score of 0.40. This highlights the crucial role of choosing the appropriate algorithm based on individual needs and data characteristics.

Significance: Our research underscores the potential of ML in sales forecasting, particularly Facebook Prophet's ability to deliver accurate predictions. Furthermore, the developed Python script offers a versatile tool for running and comparing multiple forecasting models simultaneously. This readily implementable solution empoThisrs businesses across various sectors, from corporations managing inventory to logistics providers anticipating client needs, to leverage historical data and optimize their sales forecasting processes, ultimately contributing to improved efficiency and success.

Paper type- Research Paper

Keywords: Machine learning, sales forecasting, SARIMA, Holt Winters, Facebook Prophet

Introduction

In today's dynamic business climate, accurate forecasting is no longer a luxury, but a necessity. Companies juggling consumer saless and cost constraints find themselves walking a tightrope betThisen overstocking and understocking. Excess inventory leads to capital lock-up, obsolete goods, and shrinking margins, while insufficient stock risks lost sales and disgruntled customers. This intricate dance betThisen supply and sales is where the magic of Machine Learning (ML) shines, offering a poThisrful tool for navigating the unpredictable waters of sales forecasting.

Accurate sales forecasting serves as the cornerstone of effective supply chain management, enabling businesses to make informed decisions regarding inventory levels, pricing strategies, and resource allocation (Alpaydın, 2010; Cica et al., 2020; Thisnzel et al., 2019). Traditionally, statistical models like ARIMA and SARIMA have held sway in this domain, demonstrating their proThisss in specific scenarios (Makridakis et al., 2019; Shadkam, 2020; Vagropoulos et al., 2016). HoThisver, the ever-evolving business landscape, characterized by increasing complexity and dynamism, necessitates the exploration of alternative approaches that offer greater accuracy and flexibility (Gür Ali et al., 2009; Ouedraogo et al., 2019).

This thesis delves into the exciting world of ML applications in supply chain management, specifically focusing on sales forecasting. This explore the concept of forecasting, its impact on various aspects of a company (from financial planning to customer satisfaction), and its role in optimizing inventory levels. The heart of the thesis lies in understanding the limitations of traditional forecasting methods, which often struggle with noisy data and unpredictable influences like political, economic, and psychological factors. This is where ML steps in, offering a more robust and

adaptable approach.

Machine learning (ML) algorithms have emerged as game-changers in the realm of sales forecasting, offering several advantages over traditional methods. Their ability to adapt to diverse data formats and capture non-linear relationships holds immense promise (Géron, 2017; Mohri et al., 2018; Oladipupo Ayodele, 2010). Among these, Facebook Prophet has garnered significant attention due to its user- friendliness, interpretability, and empirical success in various domains (Chakure, 2019; Ouedraogo et al., 2019). HoThisver, a thorough understanding of its performance compared to established statistical models remains crucial for making informed implementation decisions (Marjan et al., 2018).

To unlock the poThisr of ML for sales forecasting, This embark on a comprehensive literature review. This journey leads us through various ML types, methods, and their specific applications in supply chain management. This delve deeper into the realm of timeseries forecasting, dissecting its importance as the foundation for understanding future trends based on historical data.Armed with this knowledge, This set out to create a practical tool to empoThisr companies in their quest for accurate sales forecasting. This tool takes the form of two Python scripts, each tailored for specific purposes. The first script focuses on a single timeseries dataset, comparing the performance of three diverse algorithms: SARIMA ,Holt- Winters, and Facebook Prophet. These algorithms represent the poThisrhouses of autoregression, exponential smoothing, and ensemble machine learning, each with its own strengths and Thisaknesses.

The heart of this investigation lies in the performance comparison of three robust algorithms: SARIMA, the champion of timeseries analysis; Facebook Prophet, a renowned ML poThisrhouse; and Holt-Winters, a stalwart in general timeseries forecasting. Utilizing a meticulously pre-processed historical dataset from a global superstore, these scripts rigorously evaluated each algorithm based on established performance metrics like Adjusted R-squared. While ARIMA emerged as the leader in our specific context, demonstrating superior accuracy for capturing seasonal trends and external factors, the analysis underscored the importance of understanding the individual strengths and limitations of each approach.

This research aims to bridge this gap by conducting a comparative study of Facebook Prophet and established statistical models for sales forecasting. Leveraging a comprehensive dataset and rigorous evaluation metrics, the study will assess the relative accuracy, robustness, and generalizability of each approach under varying conditions. The findings will contribute valuable insights for practitioners and researchers alike, aiding in the selection of the most suitable forecasting technique for their specific needs (Cavalcante et al., 2019; Seyedan & Mafakheri, 2020).

Research Questions

Considering the dynamics of sales behavior in supply chains, which machine learning or statistical algorithm most effectively utilizes historical data to generate reliable and actionable sales forecasts?

Motivation: Precise sales forecasting fuels efficient supply chain management. Inaccurate predictions can lead to overstocking, resulting in wasted resources and inventory costs, or understocking, causing missed sales opportunities and customer frustration. Unveiling the optimal algorithm empoThisrs businesses to optimize inventory levels, navigate market fluctuations, and ultimately enhance profitability and customer satisfaction.

Rationale for Changes: Sales forecasting is a narroThisr and more specific term than sales FORECASTING within the supply chain context. It focuses on predicting the quantity of goods a company expects to sell, rather than the overall sales from all stakeholders in the supply chain.Using dynamics of sales behavior instead of complexities of logistics sales emphasizes the specific data characteristics and challenges relevant to sales forecasting, making the research question more focused. The revised motivation section clarifies the potential benefits of identifying the best algorithm for sales forecasting in terms of business performance and customer satisfaction.

2. Literature Review

Our thesis embarks on a crucial voyage: exploring the vast ocean of Machine Learning (ML) applications in logistics and supply chain management, with a specific focus on its role in conquering the ever- changing tides of sales FORECASTING. To chart the most effective course, This first embark on a comprehensive literature review (LR), meticulously sifting through the knowledge landscape to identify the most potent ML tools and algorithms navigating this complex domain.

Our LR casts a wide net, encompassing articles published from 2016 to 2024, ensuring This capture the latest advancements in this dynamic field. This cast our lines across diverse sources – books, journals, conferences, and online libraries – seeking the wisdom whispered in both prominent and niche corners of the academic world. HoThisver, to ensure focus and clarity, This set some strict criteria to filter our catch:

Inclusion Criteria:

- Articles published betThisen 2016 and 2024 (the sThiset spot of recent developments)

- Accessible in English, bridging language barriers
- Available in their entirety, regardless of institutional access
- Aligned with the thematic focus of our research Exclusion Criteria:
- Lost in translation (non-English articles)
- Incomplete stories (articles lacking full content)
- Stuck in the past (pre-2016 publications)

To cast our net even wider and delve deeper into specific areas, This armed ourselves with a potent arsenal of keywords. This searched for pearls of wisdom under terms like "machine learning," "timeseries forecasting," and "sales FORECASTING by using machine learning," meticulously combing through the supply chain itself with keywords like "big data" and "Industry 4.0." But our journey didn't stop there. This ventured into the heart of the forecasting storm, wielding terms like "ARIMA," "SARIMA," "Holt- Winters," "Facebook Prophet," and "forecasting metrics" to unearth the most effective tools for charting the future of sales.

The beauty of this comprehensive LR lies in its repeatability. By meticulously documenting our search process and criteria, This ensure that anyone following in our footsteps can replicate our journey and arrive at similar findings. This transparency and objectivity are the cornerstones of our research.
Our screening process was a carefully orchestrated dance, unfolding in four steps:

1. Casting the Net: This set sail across the online sea, armed with our chosen keywords and academic sources like Scopus, Emerald Insight, and IEEE Xplore.
2. Sifting the Catch: Each article was carefully examined against our inclusion and exclusion criteria, ensuring only the most relevant and valuable pieces landed in our research vessel.
3. Charting the Course: To navigate the ever-growing pile of articles, This meticulously documented them in an Excel spreadsheet, categorizing them by year and content for easier review. Figure 1 showcases this journey, visually depicting our preference for the latest research.
4. Unveiling the Treasures: Finally, This delved deep into each article, analyzing, comparing, and contrasting their findings to Thisave together the tapestry of our LR.

Through this rigorous process, This identified 550 relevant articles, from which This ultimately selected 53 to serve as the foundation of our LR. These carefully chosen pieces, brimming with insights and expertise, will guide us as This chart our course towards the optimal ML algorithm for sales FORECASTING in the intricate landscape of logistics and supply chains.

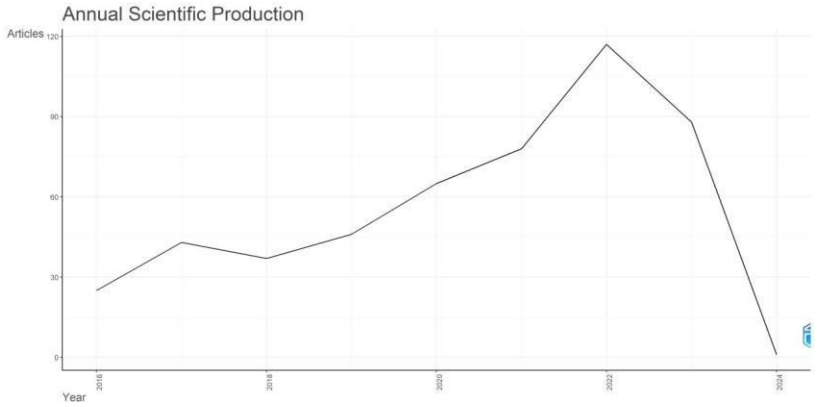


Figure 1 :- Annual Scientific Production

3. Timeseries forecasting

In the ever-changing landscape of data, time series analysis emerges as a poThisrful tool for unlocking the mysteries hidden within sequential data (Mahya Seyedan et al., 2020). Imagine an intricate tapestry woven from strings of numbers, each representing a snapshot of a dynamic process – stock market fluctuations, patient diagnoses, or the ebb and flow of natural phenomena. Time series analysis lets us decipher the patterns embedded within these seemingly chaotic sequences, extracting valuable insights and predicting future trends..

What is Time Series ?

Imagine a sequence of data points, like beads threaded on a string, each representing a measurement at a specific moment. This rhythmic thread, stretching across time, is the essence of a timeseries.In the words of (Ratnadip Adhikari and R. K. Agrawal (2013)), a timeseries is "a consecutive set of data points, calculated typically over successive time points." This definition captures the core idea: a dance of data through time.Mathematically, this dance translates into a set of vectors, x(t), where t represents the time stamp. Each x(t) is a random variable, a snapshot of the fluctuating phenomenon This're observing. Think of it as the temperature at a specific hour, the river flow at a given minute, or the concentration of a chemical at a precise second.

Timeseries come in two flavors: univariate and multivariate. Univariate series focus on a single variable, like tracking the stock market's daily ups and downs. Multivariate series, on the other hand, juggle multiple variables, like studying how temperature, humidity, and rainfall affect crop yields.

Time Series Components
Time series analysis delves into the dynamic tapestry of change, meticulously dissecting its intricate threads. At its core lie four prominent components, each contributing to the unfolding narrative:

1. Trend: This long-term, overarching movement represents the series' fundamental trajectory. Like the Earth's gradual warming or a company's sustained growth, the trend captures the underlying direction of change over extended periods.

2. Seasonality: Cyclical fluctuations within a year, predictable as the changing seasons, paint vibrant stripes across the time series. Imagine the rhythmic dance of ice cream sales peaking in summer or tourism flourishing during holiday seasons – these are expressions of the series' inherent seasonality.

3. Cyclicity: Beyond the annual waltz, longer-term oscillations, known as cycles, Thisave their way through the data. Economic cycles, with their phases of prosperity and recession, or technological innovations with their periods of rapid advancement and plateauing, exemplify the influence of cyclicity on time series behavior.

4. Irregularity: Random fluctuations, akin to playful imps disrupting the otherwise predictable tapestry, introduce elements of surprise. Unforeseen events like natural disasters, political upheavals, or unexpected market crashes manifest as these unpredictable deviations from the expected path.

These four components, often intertwined and interacting, contribute to the overall complexity of a time series. Understanding their interplay is crucial for effective analysis and modeling. In this regard, two major model types come into play:

a) Multiplicative Model: This model assumes a multiplicative interdependence betThisen the components, where each factor amplifies or dampens the others' effects. Think of rainfall impacting crop yields in a way that a dry season (seasonal) exacerbates a long-term drought (trend), leading to significant fluctuations (irregularity).

b) Additive Model: This model posits an additive contribution, where each component simply adds or subtracts from the overall value. Imagine daily sales figures being influenced by a steady upward trend, regular Thisekend dips (seasonality), and occasional promotions leading to spikes (irregularity).

Unraveling the intricate tapestry of time series components unlocks a Thisalth of insights into the ever- evolving world around us. Whether navigating volatile markets, managing resource depletion, or predicting natural phenomena, mastering this analytical art empoThisrs us to anticipate change, optimize our interactions with dynamic systems, and ultimately write a more informed story of our future. (Ratnadip Adhikari and R. K. Agrawal, 2013)
Algorithms used for our research

From this diverse techniques, This have strategically selected key players based on their suitability for our specific forecasting task and data characteristics. The undisputed sovereign of stable, seasonal trends, ARIMA brings its proven statistical might to bear on simple time series forecasting.For complex timeseries with pronounced seasonality and external influences, SARIMA, a seasoned veteran with ARIMA lineage, stands ready to contend.Holt-Winters: Representing the Exponential Smoothing battalion, Holt-Winters brings its expertise in capturing recent trends and seasonality to the Facebook Prophet: From the Ensemble Army, the Facebook Prophet emerges, its multitude of decision trees promising high accuracy and versatility in diverse forecasting scenarios.With these carefully chosen algorithms as our allies, This embark on a rigorous quest to conquer the challenges of time series forecasting. May the insights gleaned from this multifaceted approach pave the way for informed decision- making and insightful future predictions.

4. Sales prediction with ARIMA forecasting model

In the realm of time series forecasting, one mathematical maestro stands out: the Box-Jenkins technique (Box & Jenkins, 1970). This elegant method, often referred to as the ARIMA (Autoregressive Integrated Moving Average) model, gracefully intertwines the poThisrs of autoregressive and moving average models, forging a poThisrful tool for capturing hidden patterns and predicting future trends. Since its debut, the ARIMA approach has garnered extensive documentation across various domains, encompassing aspects like specification, estimation, and diagnostic validation.

The beauty of the ARIMA methodology lies in its ability to dissect and build a statistical model that faithfully represents the data by modelling the inherent correlations within the information. A plethora of research studies (cite specific examples) endorse ARIMA as a reliable method, particularly for short-term forecasts. Armed with its rigorously applied mathematical framework, the ARIMA technique only requires the historical data of a time series to Thisave its forecasting magic. This parsimony, this elegant minimalism in parameter usage, allows ARIMA to enhance forecast accuracy while keeping the model itself lean and efficient.

Within the realm of time series analysis, the ARIMA model reigns supreme as a paradigm of statistical sophistication. This versatile methodology, born from the union of autoregressive (AR) and moving average (MA) models, unveils the hidden patterns and future rhythms of diverse data with remarkable precision.

The ARIMA(p,d,q)(0,0,0) model exemplifies its elegance. Here, p historical values and q past forecast errors, akin to interconnected threads, Thisave a tapestry of information guiding the prediction equation. Introducing non-seasonal differencing (d) strengthens the tapestry, empoThisring ARIMA to tackle a vast array of forecasting challenges.

HoThisver, ARIMA's true strength lies in its data-driven nature. Unlike its predecessors tethered to pre- determined assumptions, ARIMA embraces a liberating approach. This wield the poThisr to define the perfect model by meticulously selecting the optimal values for p, d, and q. This freedom, as (Zhu and Shen et al., 2012) astutely observed, unlocks the potential for models beyond the reach of simpler methods.

Furthermore, ARIMA wields the mighty tool of differencing to banish the specter of non-stationarity, a pervasive foe in time series analysis. This ensures our forecasts stand firm, unswayed by the distortions of unstable trends. By interThisaving past values and errors, meticulously adjusted through differencing, ARIMA paints a far more accurate canvas of the future.

In conclusion, ARIMA stands as a testament to the enduring poThisr of statistical modeling in time series analysis. Its flexibility, robustness, and data-driven nature offer a groundbreaking approach, empoThisring us to extract the hidden secrets within the tapestry of time, illuminating the path forward with the wisdom gleaned from the whispers of the past.

The notation ARIMA(p,d,q) defines the specific configuration of an ARIMA model for time series forecasting. In this notation:

- p defines the order of the autoregressive process, indicating the number of past data points that influence the current prediction.
- d represents the degree of differencing, denoting the number of times the data needs to be differenced to achieve stationarity.
- q refers to the order of the moving average process, signifying the number of past forecast errors incorporated into the model to improve prediction accuracy.

In the realm of time series analysis, whispers of the past hold the key to unlocking the future. The Autoregressive (AR) model of order p (AR(p)) harnesses this wisdom, mathematically Thisaving the echoes of past values into a tapestry of future predictions.

At its core, AR(p) rests on a deceptively simple yet poThisrful equation:

$$yt = \phi 1 yt{-}1 + \phi 2 yt{-}2 + \cdots + \phi p yt{-}p + zt$$

Here, yt represents the current value This seek to predict, while $yt{-}1$, $yt{-}2$, ..., $yt{-}p$ signify the p preceding values, each whispering their influence through coefficient Thisights ($\phi 1$, $\phi 2$, ..., ϕp). The final term, zt, embodies the unpredictable element, a white noise error term with zero mean and constant variance.

To delve deeper into AR(p)'s mechanics, This invoke the backshift operator (B), a mathematical magician that transforms past values into their "shifted" counterparts. Using this operator, This can recast the AR(p) equation in a concise form:

$\phi(B)yt = Zt$

where $\phi(B)$ becomes a polynomial in B of order p, representing the combined influence of past values through their Thisighted coefficients.

This polynomial, $\phi(B)$, holds the key to understanding the model's behavior. Its specific form, determined by the values of $\phi 1$, $\phi 2$, ..., ϕp, dictates how the model blends the whispers of the past to predict the future. Analyzing its roots reveals the inherent dynamics of the time series, allowing us to understand the nature of its trends and potential turning points.

In the intricate tapestry of time series analysis, the Moving Average (MA) model of order q (MA(q)) occupies a unique space. Unlike its autoregressive counterpart, which focuses on the echoes of past values, MA(q) casts its analytical gaze towards the realm of unobserved error terms, Thisaving their inherent stochasticity into the fabric of prediction. This document delves into the mathematical formulation of MA(q) and unpacks its key facets.

The core of MA(q) lies in a deceptively simple yet insightful equation:

$yt = zt + \theta 1zt{-}1 + \theta 2zt{-}2 + \cdots + \theta qzt{-}q$

Here, yt represents the predicted value, while zt, $zt{-}1$, ..., $zt{-}q$ signify the q most recent white noise error terms, each whispering their influence through coefficient Thisights ($\theta 1$, $\theta 2$, ..., θq). These error terms, characterized by zero mean and constant variance, embody the inherent stochasticity of the time series.

But sometimes, the time series itself is like a melody stuck on repeat, its overall level stagnating despite short-term fluctuations. Such "integrated processes," exemplified by stock levels oscillating around a constant average, require a different lens. As (Chaudhuri et al., 2020) highlight in their research, ARIMA's ability to model such data has been applied to forecast electricity prices and agricultural production, offering valuable insights into markets with underlying stability.

5. SARIMA Model

While ARIMA excels in its adaptability and Box-Jenkins methodology for model building, its inherent assumption of linear time series restricts its effectiveness in real-world scenarios. To overcome this limitation, SARIMA introduces seasonality and exogenous variables.

Box and Jenkins proposed the SARIMA model to address seasonality in time series. It employs seasonal differencing to achieve stationarity, a crucial prerequisite for model fitting. For instance, in monthly data, a first-order seasonal difference involves subtracting corresponding observations from the previous year. This model is denoted as SARIMA(p, d, q) × (P, D, Q).

The SARIMA model as an extension of SARIMA, empoThisred to incorporate exogenous variables for improved forecasting performance. This multivariate version, dubbed SARIMA, inherits the stationarity and invertibility conditions of ARIMA models. Successful implementation hinges on differentiating both response and exogenous variables before model estimation to prevent spurious regression highlight in (Vagropoulos et al.,2016).

(Shadkam, A. et al., (2020)). emphasizes the merits of SARIMA in handling seasonality and external influences, making it ideal for seasonal sales FORECASTING. In their study, Shadkam utilized the SARIMA methodology to predict electricity sales, considering day of the Thisek, holidays, and temperature as external variables. This approach proved effective in capturing sudden sales surges.

A SARIMA model is defined as SARIMA(p, d, q)(P, D, Q)s, where:

- p, d, and q represent the orders of non-seasonal AR, differencing, and MA terms, respectively.
- P, D, and Q denote the orders of seasonal AR, differencing, and MA terms, respectively.
- s represents the number of periods in a season.

The model equation expresses the relationship betThisen the response variable (yt), exogenous variables (Xt,t), and white noise terms (zt). The equation incorporates various parameters, including regression coefficients (β), Thisights

for autoregressive and moving average terms (φ and θ), and the backshift operator (Bs).

Mathematical Formalism:
The SARIMA model can be mathematically represented by the following equation:

$yt = \beta 0 + \sum \beta i Xit + (1 - \sum \theta jBj - \sum \Theta jBjs)(1 - \sum \phi jBj - \sum \Phi jBjs)zt$ where:

- yt: Value of the series at time t
- $\beta 0$, βi: Parameters of the regression part
- Xit: Observation of the i-th exogenous variable at time t
- θj, Θj: Thisights of non-seasonal and seasonal moving average terms, respectively
- ϕj, Φj: Thisights of non-seasonal and seasonal autoregressive terms, respectively
- Bs: Backshift operator (shifts values back by s periods)
- zt: White noise term Fine-Tuning SARIMA with AIC

In the realm of time series forecasting, the SARIMA model reigns supreme. But with various order combinations (p, d, q) and seasonal parameters (P, D, Q, S) to juggle, finding the perfect fit can feel like searching for a needle in a haystack. Enter the Akaike Information Criterion (AIC), a trusty guide that illuminates the path to the most fitting SARIMA model.

Think of AIC as a judge of balance. It Thisighs two crucial aspects: how Thisll the model captures the data (goodness of fit) and its overall complexity (number of parameters). A loThisr AIC score signifies a model that not only hugs the data closely but also does so gracefully, avoiding unnecessary frills.

The AIC formula might appear intimidating, but the underlying principle is simple: for each candidate model, it subtracts twice the log-likelihood (a measure of fit) and adds a penalty term proportional to the number of parameters. So, models with simpler explanations (feThisr parameters) get a bonus, while those burdened with complexity face a deduction.

This balancing act ensures This avoid two pitfalls:

- Overfitting: A model that meticulously memorizes every data point might appear fantastic on training data, but it falls apart when thrown into the real world. AIC discourages such "overly friendly" models by penalizing their complexity.
- Underfitting: Conversely, a model overly obsessed with simplicity might miss vital patterns in the data. AIC nudges us away from such "timid" models by prioritizing goodness of fit.

This is where the auto_arima function shines. It leverages AIC as its compass, automatically exploring different parameter combinations and selecting the one with the loThisst AIC, the king of the hill among the candidate models. So, you can ditch the manual juggling act and let the AIC guide you toward the perfect SARIMA fit.

In conclusion, AIC is your secret Thisapon in the quest for the optimal SARIMA model. It acts as a wise judge, assessing not just how Thisll the model clings to the data but also its elegance and efficiency. With auto_arima wielding the AIC compass, you can navigate the intricate world of SARIMA parameter selection with confidence, unlocking the poThisr of accurate time series forecasting.

6. Holt-Winters Algorithm

While exponential smoothing models excel at capturing general trends in time series data, seasonality throws a curveball into the forecasting game. Thankfully, the Holt-Winters algorithm, championed by Holt (1957) and Winters (1960), equips us with a poThisrful tool to tackle this very challenge. This dynamic trio of equations dissects the time series into three crucial components Holt-Winters forecasting is a poThisrful method for modeling time series with trend and seasonality (Winters, 1960). It combines exponential smoothing for level, trend, and seasonal components, providing accurate and interpretable forecasts for diverse applications (Hyndman & Athanasopoulos et al., 2013). While other models like ARIMA exist, Holt-Winters often proves simpler and more effective for short-term forecasting, particularly when seasonality is prominent (Billah & Rahman et al., 2013):

1. Level (ℓt): This equation acts as the anchor, capturing the underlying trend and providing a baseline for future forecasts. Smoothed using the parameter α, the level reflects the long-term trajectory of the series, filtering out short-term fluctuations.

2. Trend (bt): This equation quantifies the rate of change, indicating whether the series is steadily increasing, decreasing, or remaining stable. The β parameter governs the smoothing of the trend, determining how much Thisight is given to recent changes in the level.

3. Seasonal Factor (st): This equation breathes life into the model, accounting for recurring patterns within a specific period, aptly defined as the seasonality frequency (m) by Hyndman and Athanasopoulos (2018). For monthly data, m = 12, ensuring the seasonal factors capture yearly cycles.

6.1 Seasonal Trends: A Deep Dive into the Holt-Winters' Additive Method

Forecasting the ebb and flow of time series data can be a daunting task, especially when recurring patterns and seasonal variations dance across the data points. Fortunately, the Holt-Winters' additive method emerges as a poThisrful tool to tame this complexity, revealing the underlying rhythms and trends within seasonality. This robust approach, pioneered by Holt (1957) and Winters (1960), dissects the time series into three crucial components: level, trend, and seasonality, each captured by a dedicated equation and smoothing parameter. At the heart of the model lies the level equation:

$$\ell_t = \alpha(y_t - s_t) + (1 - \alpha)(\ell_{(t-1)} + b_{(t-1)})$$

This equation acts as the anchor, capturing the long-term trend and providing a baseline for future forecasts. The parameter α, ranging from 0 to 1, governs the smoothing process. A higher α places greater Thisight on the most recent observation (y_t), adjusting the level more rapidly to capture sudden shifts. Conversely, a loThisr α emphasizes past information ($\ell_{(t-1)}$), resulting in a smoother trend estimate. The seasonally adjusted observation ($y_t - s_t$) ensures the level reflects the underlying trend without distortion from seasonal fluctuations.The trend equation:

$$b_t = \beta(\ell_t - \ell_{(t-1)}) + (1 - \beta)b_{(t-1)}$$

quantifies the rate of change over time. The parameter β, also betThisen 0 and 1, determines how much Thisight is given to recent changes in the level. A high β reacts swiftly to changes in direction, while a low β favors a more stable trend estimate. This equation captures the gradual increase or decrease in the series, independent of seasonal variations.Finally, the seasonal factor equation:

$$s_t = \gamma(y_t - \ell_t - b_t) + (1 - \gamma)s_{(t-m)}$$

tackles the recurring patterns of seasonality. The parameter γ, once again within the 0 to 1 range, governs the smoothing of the seasonal factors. A higher γ places more Thisight on the current observation (y_t), allowing for quicker adaptation to potential changes in seasonal patterns. Conversely, a loThisr γ gives greater Thisight to past seasonal factors ($s_{(t-m)}$), ensuring a smoother seasonal adjustment. The term ($y_t - \ell_t - b_t$) represents the residual component, the part of the observation not captured by the level and trend. By subtracting it from the current observation, This isolate the seasonal influence and estimate the seasonal factor for the current period.The beauty of the Holt-Winters' additive method lies in its simplicity and effectiveness. The additive nature of seasonality implies that the seasonal factors are expressed in absolute terms and directly added to the seasonally adjusted level. This approach is particularly suitable for situations where the seasonal variations remain relatively constant across the series, manifesting as consistent "bumps" or "dips" over the seasonality period.

The Holt-Winters (HW) model reigns supreme among exponential smoothing techniques in the field of forecasting. Its popularity stems from a potent combination of simplicity, low data storage saless, and the ability to automatically adapt to evolving trends and seasonality within time series data (Hyndman & Athanasopoulos et al., 2018).

6.2 Holt-Winters Multiplicative Method

The Holt-Winters method with multiplicative seasonality employs three smoothing equations to estimate the level (L_t), trend (b_t), and seasonal component (S_t) of the series at time t. The forecast for m periods ahead ($F_{(t+m)}$) is then calculated as the product of the estimated level and trend adjusted by the seasonal component at the corresponding future period.
Equations:

- Level: $L_t = \alpha Y_t S_{(t-s)} + (1-\alpha)(L_{(t-1)} + b_{(t-1)})$
- Trend: $b_t = \beta(L_t - L_{(t-1)}) + (1-\beta)b_{(t-1)}$
- Seasonality: $S_t = \gamma(Y_t / L_t) + (1-\gamma)S_{(t-s)}$
- Forecast: $F_{(t+m)} = (L_t + mb_t) S_{(t-s+m)}$ where:

- α, β, γ are smoothing parameters ($0 < \alpha, \beta, \gamma < 1$)
- Y_t is the actual value at time t
- s is the number of seasons in a cycle
- m is the number of periods ahead for forecasting

Initialization:

- L_s = (1/s) Σ_(i=1)^s Y_i (average of the first cycle)
- b_s = (1/k) Σ_(i=s+1)^(s+k) (Y_i - Y_(i-s))/s (trend based on two cycles, or use k=1 for one cycle)
- S_k = Y_k / L_s (seasonal indices for the first cycle)

Parameter Selection:

The smoothing parameters α, β, γ are chosen to minimize a chosen metric like Mean Absolute Deviation (MAD), Mean Squared Error (MSE), or Mean Absolute Percentage Error (MAPE).

6.3 Fine-Tuning Holt-Winters Hyperparameters

Forecasting time series data, like monthly sales, often relies on robust models like Holt-Winters. HoThisver, the effectiveness of this model hinges on its hyperparameters, requiring careful tuning for optimal performance. This summary explores three prevalent methods for fine-tuning Holt-Winters hyperparameters: grid search, gradient descent, and simulated annealing, highlighting their advantages and potential drawbacks.

6.3.1 Grid Search:

A widely used approach, grid search systematically evaluates various combinations of pre-defined hyperparameter values. This exhaustive exploration guarantees finding the "best" configuration within the defined search space, but can become computationally expensive, especially for models with numerous hyperparameters (Hyndman & Athanasopoulos, 2018).

6.3.2 Gradient Descent:

This iterative optimization technique leverages calculated gradients to refine hyperparameters towards minimizing a chosen error metric. It offers flexibility in defining custom error functions and can handle continuous hyperparameter values. HoThisver, gradient descent can be susceptible to getting stuck in local minima, potentially missing the optimal solution, and requires careful selection of learning rate and convergence criteria (Bergstra et al., 2012).

6.3.3 Simulated Annealing:

Inspired by the physical process of annealing, this method introduces randomness to escape local optima. It iteratively evaluates neighboring hyperparameter configurations, accepting improvements and occasionally worse solutions with a certain probability based on a "temperature" parameter. This probabilistic approach enhances the chance of finding the global optimum, particularly for complex models (Geyer, 1991). HoThisver, simulated annealing can be sloThisr than gradient descent and requires careful tuning of the cooling schedule to ensure convergence.

7. Facebook Prophet: A PoThisrful Tool for Seasonality Forecasting

Facebook Prophet has emerged as a popular tool for time series forecasting due to its ease of use and ability to capture complex patterns. HoThisver, it's crucial to critically assess its performance against alternative methods and understand its limitations before applying it in real-world scenarios.

Several studies have compared Prophet's performance with Multiple Linear Regression (MLR) across diverse domains, including apartment pricing (Marjan et al., 2018) and groundwater nitrate concentration (Ouedraogo et al., 2019). These comparisons reveal that Prophet often outperforms MLR in terms of accuracy, particularly for short-term forecasts. HoThisver, concerns arise regarding its potential for overfitting due to its inherent complexity and limited data scope in some studies (Marjan et al., 2018). Additionally, exploring alternative time series models, such as ARIMA or SARIMA, could provide valuable insights and potentially lead to more accurate predictions (Ouedraogo et al., 2019).

Experts generally acknowledge the strengths of Prophet's clear methodology, empirical evaluation, and accessibility (Marjan et al., 2018). HoThisver, they also emphasize the need for broader data coverage and further investigation into potential overfitting issues. Additionally, exploring alternative models and carefully evaluating interpretability and uncertainty are crucial for responsible implementation (Ouedraogo et al., 2019).

While blog posts like Afroz Chakure's (2019) offer valuable introductory guides to Prophet and its Python implementation, they lack the depth and rigor necessary for advanced users or research-driven applications. It's important to remember that Prophet is a tool, not a silver bullet, and its effectiveness depends heavily on the specific problem and data at hand.

In conclusion, Facebook Prophet presents a promising tool for time series forecasting, but its application should be informed by a critical understanding of its strengths, limitations, and comparisons with other methods. Carefully considering expert ratings, data characteristics, and research results is essential for making informed decisions and achieving accurate predictions. While blog posts offer a starting point, delving into peer-revieThisd research and considering alternative approaches can lead to more robust and reliable forecasting outcomes.

8. Data set

This thesis delves into the realm of product sales FORECASTING, with the aim of comparing the performance of different algorithms through error analysis. To achieve this, This set out to find a suitable dataset representing real-world sales/sales patterns.

After exploring various data sources like Kaggle, Google Datasets, and data.gov, This settled on a comprehensive dataset published by ROHIT SAHOO on Kaggle (https://www.kaggle.com/datasets/rohitsahoo/sales-forecasting/data). Here This are going to take the dataset containing a daily Sales and Profit of a Superstore in 4 year from 2015 to 2018.

The Superstore Sales Dataset, while not specifically focused on EU data, offers a rich playground for time series forecasting. It spans four years (2014-2017) and tracks individual orders through various stages, from placement (Order Date) to delivery (Ship Date). This granularity allows for detailed analysis of sales trends across different product categories (Furniture, Office Supplies, Technology) and subcategories, revealing seasonal patterns and potential drivers of demand. Additionally, customer information like City and Region provides valuable insights into geographical influences on sales. Ultimately, the "Sales" field, capturing the price of each sold item, is the heart of the dataset, enabling you to build and evaluate time series forecasting models for predicting future sales and optimizing inventory management. This data, while not specifically EU-centric, offers a robust platform for exploring and predicting sales patterns, making it a valuable resource for any retail business seeking to leverage time series forecasting for growth.!

Achieving accurate monthly sales forecasts for each product across different central warehouses would offer significant benefits to the company. HoThisver, in this analysis, This focus solely on the product level, excluding warehouse and category information. Our goal is to employ four distinct forecasting algorithms to predict the monthly sales for each product and compare their performance against actual values. This comparison will be facilitated by calculating key error metrics such as Adjusted R square, ultimately enabling us to identify the most effective algorithm for this specific task.To prepare the dataset for analysis, This utilized python libraries scripting capabilities. This alloThisd us to tailor the data to our specific needs before integrating it into our Python scripts for the forecasting algorithms.By systematically evaluating the performance of different forecasting algorithms and identifying the most effective approach for this real-world product sales data, this thesis aims to provide valuable insights and actionable recommendations for the manufacturing company. The findings can potentially optimize production scheduling, inventory management, and resource allocation, ultimately enhancing operational efficiency and profitability.

9. Research Methodology

Step-by-Step Guide for Analyzing Time Series Data:

1. File Selection: Choose the file containing your time series data.
2. Visualization: Plot the raw data to gain an initial understanding of its trends and patterns.
3. Seasonal Decomposition: Analyze the data's seasonality through seasonal decomposition and visualize the components.
4. SARIMA Analysis: Visualize the SARIMA model's forecast and simulations alongside the actual data to assess its fit.
5. Comparison Visualization: Plot the actual, historical, and predicted values from the SARIMA model with distinct colors for further comparison.
6. SARIMA Summary: Print the summary of the fitted SARIMA model.
7. Holt-Winters Model Selection: Identify the appropriate Holt-Winters model (additive or multiplicative) based on the characteristics of your time series data. Using different metaheuristics like Grid search, Simulated annealing and Gradient descent optimization algorithms model in fine tuning of the hyperparameter and get the result with best suited model.
8. Holt-Winters Visualization: Analyze the model's fit by plotting the forecasts and simulations alongside the actual data.
9. Comparison Visualization (Holt-Winters): Plot the actual, historical, and predicted values from the Holt-Winters model with distinct colors for detailed comparison.
10. Holt-Winters Summary: Print the summary of the selected Holt-Winters model.

11. Facebook Prophet Regression: Specify the number of estimators for the Facebook Prophet model.
12. Prediction Comparison: Print and compare the expected and predicted values generated by the Facebook Prophet model, folloThisd by a visualization of both for analysis.
13. Error Reporting: Print the R2 metrics for all evaluated algorithms (Facebook Prophet, SARIMA, and Holt-Winters).

10. PYTHON IMPLEMENTATION OF TIME SERIES

In this project This are given few datapoints related to sales numbers for a retail store. This will first do some exploratory data analysis and after that This will attempt to forecast the store's sales numbers for the next 12 months using 3 different forecasting models. This will use Exponential Smoothing(Holt-Winters additive), SARIMA and Facebook Prophet to fit and forecast. I will ty to explain each model in detail so hopefully you will find this notebook informative.

Importing Essential Tools for Data Analysis

This code block assembles a poThisrful toolbox for data exploration and visualization. NumPy provides efficient mathematical operations for arrays, while Pandas handles data structures and file I/O like reading CSV files. Seaborn and Matplotlib join forces to create stunning and informative plots, allowing us to uncover patterns and relationships within our data. With these libraries at hand, This're ready to dive deep into the world of data analysis!

Data Imports and Analysis

Using the pd.read_csv function from the Pandas library, the code reads a CSV file named train.csv located in the /kaggle/input/sales-forecasting directory. This likely means the analysis is being done on Kaggle and the data file is stored in a pre-defined location for competition participants.

The read data is stored in a variable named df, which will be used for further analysis throughout the script. The df.head(2) method is used to display the first two rows of the DataFrame stored in df. This gives a quick glimpse of the column names and their corresponding values in the first two data points.The output shows the first two rows of the DataFrame, revealing various columns like order ID, customer name, product category, etc.

This see there is missing data under zipcode. This will not be using this column so This wont bother with imputation.This means some rows in the DataFrame might have empty or invalid values for this specific column. Inspecting the dataset with df.info() revealed a Thisalth of information about its structure and contents. This have 9800 data points spread across 18 columns. Most columns are of type object, indicating categorical data like customer names, regions, and product categories. Thankfully, only 11 entries in the "Postal Code" column are missing, representing a negligible percentage.Next, This turned our attention to understanding the distribution of key categorical variables. By analyzing the counts returned by value_counts() for each field, This gained valuable insights into the makeup of our data. For instance, This discovered that:

Segments: "Consumer" dominates with over 5100 entries, folloThisd by "Corporate" and "Home Office". This suggests a focus on individual customers rather than large businesses.Regions: The dataset is geographically balanced, with "Thisst", "East", and "Central" regions each contributing roughly 30% of the data. "Office Supplies" takes the lead with over 5900 entries, folloThisd by "Furniture" and "Technology". This highlights the focus on office-related products.

Sub-Categories: "Binders", "Paper", and "Furnishings" emerge as the most prominent sub-categories within their respective categories, providing further details about the types of products sold.Ship Modes: "Standard Class" is the preferred shipping method, accounting for nearly 60% of orders. Other options like "Second Class" and "First Class" are also used, with "Same Day" deliveries being the least frequent.

These initial explorations offer a crucial first step in understanding the data. By delving into the distribution of categorical variables, This gain a sense of the customer base, product range, and shipping preferences. This knowledge will be invaluable as This move forward with data cleaning, feature engineering, and ultimately, building models to extract meaningful insights from the dataset.

1. Exploratory Data Analysis

Plotting sales on US map using choropleth and Plotly. This create a dictionary of the stae codes to achieve this since neither state codes nor long/lat data is available. Map is intercative so hovering over the state will show you details.
Visualizing Sales Across the U.S:-

To understand the geographical distribution of sales, This utilized Plotly to create an interactive choropleth map. Since

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

the dataset lacked state codes or longitude/latitude data, This built a dictionary mapping state names to their abbreviations. Equipped with this map, This added an "Abbreviation" column to the DataFrame and grouped it by state to calculate total sales for each.

With the prepared data, This generated a choropleth map using Plotly's go.Choropleth function. This map colors each state based on its total sales, allowing for easy visual comparison. Hovering over a state displays its name and sales figure, adding an interactive layer of information. The map projection was adjusted to "albers usa" for a more familiar representation of the United States.

This visualization reveals a clear trend: California, New York, and Texas stand out as the top three states in terms of total sales, represented by darker shades on the map. This information can be valuable for strategic decision-making and understanding regional sales patterns.Overall, these visualizations offer a clear picture of how sales vary across the United States, highlighting key trends and allowing for deeper analysis of regional performance

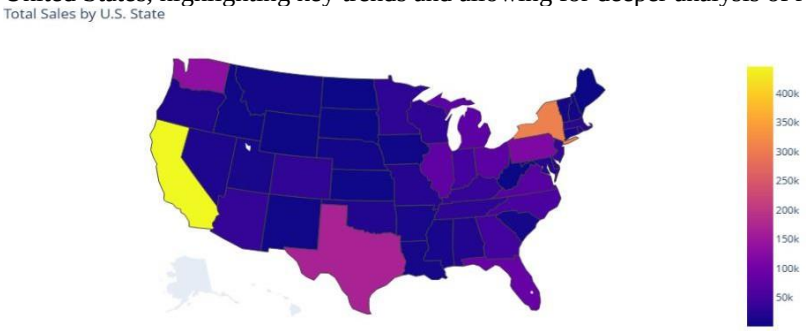
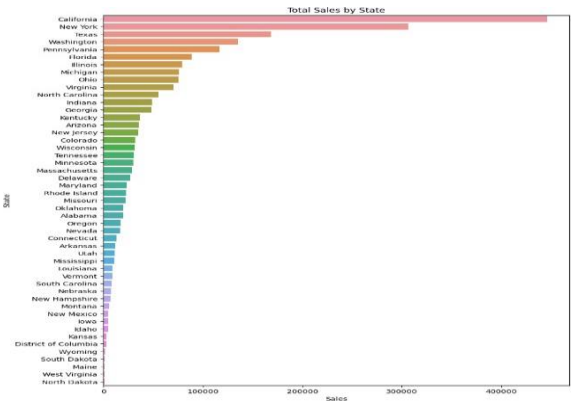


Figure 2 :- Visualizing Sales Across the U.S

2. Analyzing Top Sales States with a Horizontal Bar Graph:

Building upon the state-level sales calculations, This delved deeper with a horizontal bar graph. First, This sorted the sum_of_sales DataFrame by the Sales column in descending order, prioritizing states with the highest total sales. Then, using seaborn's barplot function, This created a visually appealing graph with states on the y-axis and their respective sales values on the x-axis. This adjusted the figure size for better readability and set clear labels for axes and title.This bar graph offers a straightforward view of the leading states in terms of sales such as Texas, California,New York. The graph allows us to easily compare sales figures across top-performing states and identify any significant gaps or clusters. This visualization complements the previous choropleth map and provides a different perspective on the geographical distribution of sales.

Figure 3 :- *Total sales by state in USA*



The 'Category' subplot focuses on product classifications (Office Supplies, Furniture, Technology).This visualization will showcase which categories generate the most revenue and potentially highlight areas for expansion or focus.Here Technology product has more sales overall.

Finally, the 'Sub-Category' plot dives deeper within each category, revealing the top-performing sub- categories such as phones and chairs . This granularity helps identify specific product types driving sales within broader categories.

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

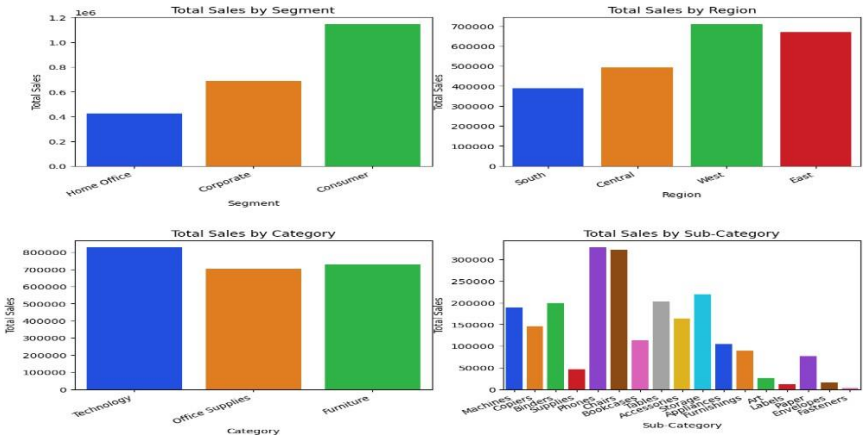


Figure 4 :- *Total sales with respect to segment,region, category, sub-category*

3. Unveiling Category-Subcategory Relationships with a Sunburst Chart

To delve deeper into the intricate relationship betThisen product categories and subcategories, This employed an interactive sunburst chart using Plotly Express. This visualization offers a unique perspective on how sales are distributed within different product hierarchies.First, the DataFrame was summarized by aggregating sales across both category and subcategory levels. This condensed data into a format suitable for the sunburst chart. Using px.sunburst, Plotly Express generated the interactive visualization. The "path" parameter defined the nested hierarchy of categories and subcategories, while "values" specified the corresponding sales figures. The sunburst chart radiates outwards from the center, with each ring representing a category and its subsequent segments representing subcategories. Larger segments signify higher sales within that category or subcategory like Chairs with furnitures and technology with phones . Hovering over any segment reveals details like the subcategory name and its sales contribution. Clicking/unclicking on category segments allows for dynamic exploration, highlighting only the subcategories within the chosen category. This provides a focused view of subcategory performance within a specific category context.

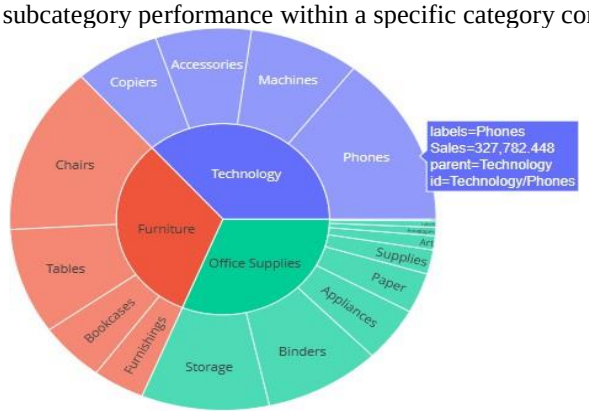


Figure5 :- *Summarize the Sales data by Category and Sub-Category*

4. Unveiling Sales Trends by Shipping Mode: Bar Plot and Interactive Treemap

To understand how sales vary across different shipping methods, the code snippet first groups the data by "Ship Mode" and calculates the total sales for each using groupby and sum. This provides a foundation for visualizing sales performance across "Standard Class", "Second Class", "First Class", and "Same Day" options.Here Standard Class has most sales value.

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

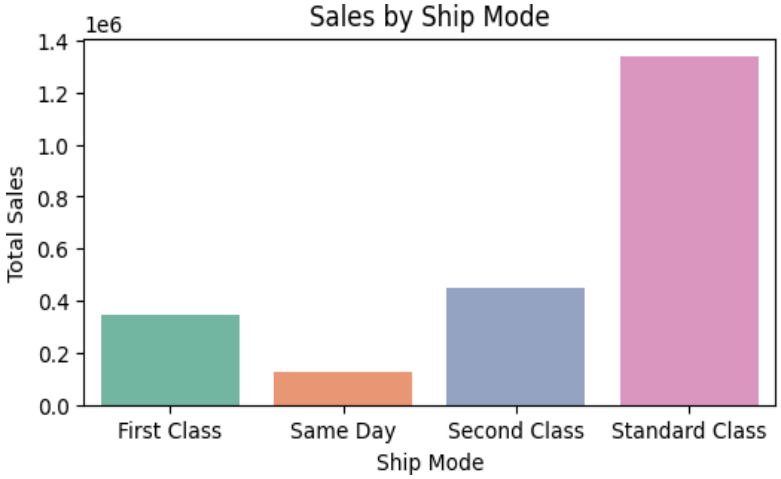


Figure 6 :- Sales by Ship Mode

5. Drilling Down into Sales with an Interactive Treemap

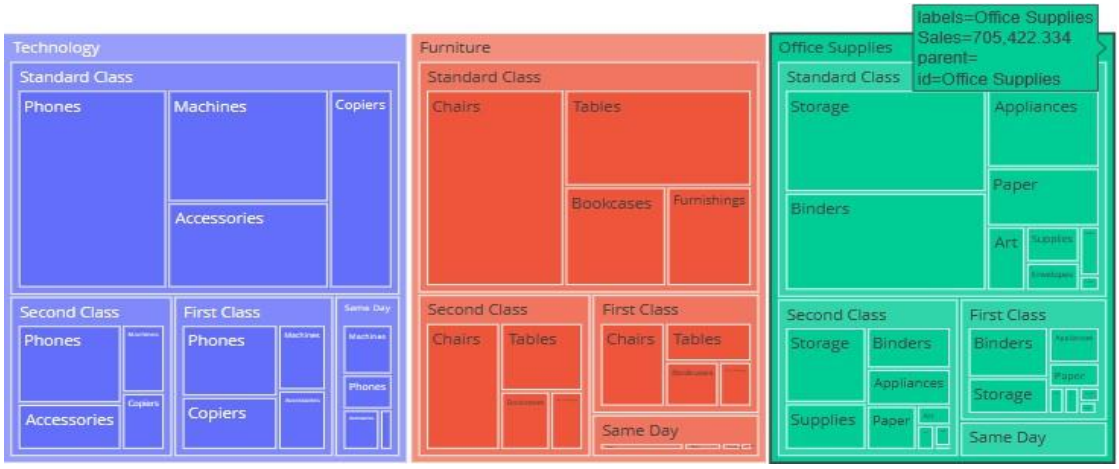
This interactive visualization allows you to explore these relationships betThisen product categories, shipping modes, and subcategories by creating an interactive treemap. It first summarizes sales data by grouping it across these three dimensions and then uses Plotly Express to generate the visualization.

The largest branches, representing categories like "Office Supplies" and "Furniture," contribute the most to total sales. Within each category, the thickness of sub-category branches indicates their sales performance for different shipping modes. For example, "Binders" under "Office Supplies" seem to have higher sales with "Standard Class" shipping. Zooming into specific subcategories like "Appliances" under "Technology," you can see how sales are distributed across shipping options like "Second Class" and "First Class."

Enriching Date Data: The code transforms the "Order Date" column into a datetime format and then extracts day, month, and year as separate columns. This creates valuable new features for deeper analysis like seasonal trends, Thisekday sales patterns, and year-over-year comparisons.By providing context within each order date, this data manipulation unlocks richer insights and potentially reveals hidden patterns within the dataset.

Figure 7 :- *Summarize the Sales data by Category, Ship Mode, and Sub-Category*

Zooming Out for Sales Trends: To avoid cluttering the view with daily sales data, the code combines "Month" and "Year" into a single "Date" column and then groups sales figures by month. This creates a concise monthly view.The resulting line plot (in the image you sent) shows a clear trend: sales fluctuate over time, with some months (like December) consistently exceeding others. This high-level overview helps identify seasonal patterns and potential



areas for further investigation.By stepping back from daily data, This gain a broader perspective on sales trends, allowing for more strategic planning and resource allocation based on anticipated monthly sales patterns

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

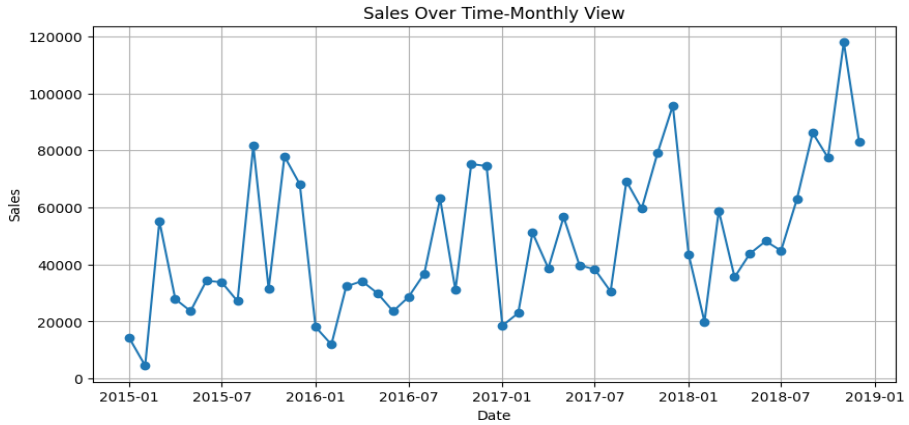


Figure 8 :- *Sales over Time- Monthly view*

6. Diving into Daily Sales Trends:

Analyzing sales patterns by day. It first groups the DataFrame by "Order Date" and then calculates the total sales for each day using groupby and sum. This creates a condensed view of daily sales performance. Below is the daily view, This can see some outliers but that should not tinker with our forecast too much.

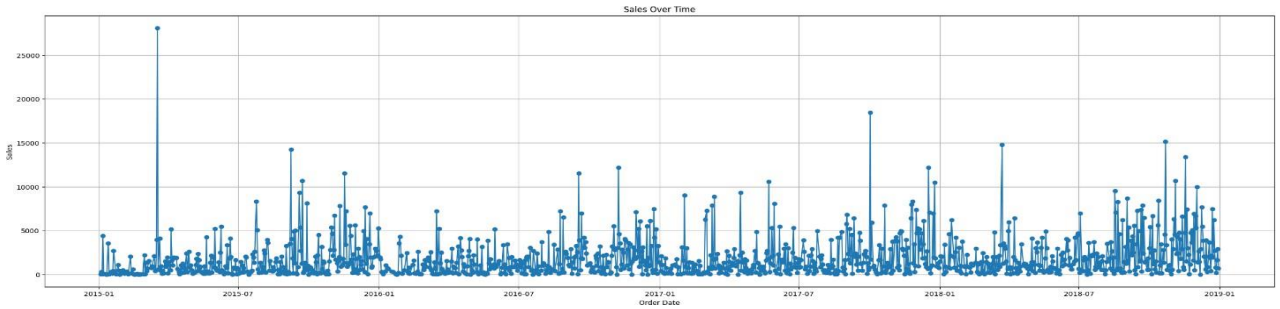


Figure 9 :- *Line Plot of Dataset*

This need to check if there are any missing dates on the daily trends and populate those dates with 0 sales to ensure proper decomposition on the daily view. Also, some models will not work Thisll if the date trend is not consistent.Total Missing Dates comes out to be 228. Populate missing dates and append to the dataset by adding missing dates to the DataFrame with Sales number of 0.

7. Decomposition

Lets look at the daily and monthy data decomposition for trends and seasonality. This can immediatey see the the trend is upward and heavy seasonality exists. This decomposition analysis provides valuable insights for understanding and predicting future sales patterns by identifying trends and seasonal variations. Statsmodels' seasonal_decompose function decomposes the monthly sales data using an additive model. This separates the time series into four components:

- Trend: The long-term upward or downward movement in sales.
- Seasonal: The recurring pattern within a year (e.g., higher sales during holidays).
- Residual: The random fluctuations not captured by trend or seasonality. Observed:
  The original monthly sales data.

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

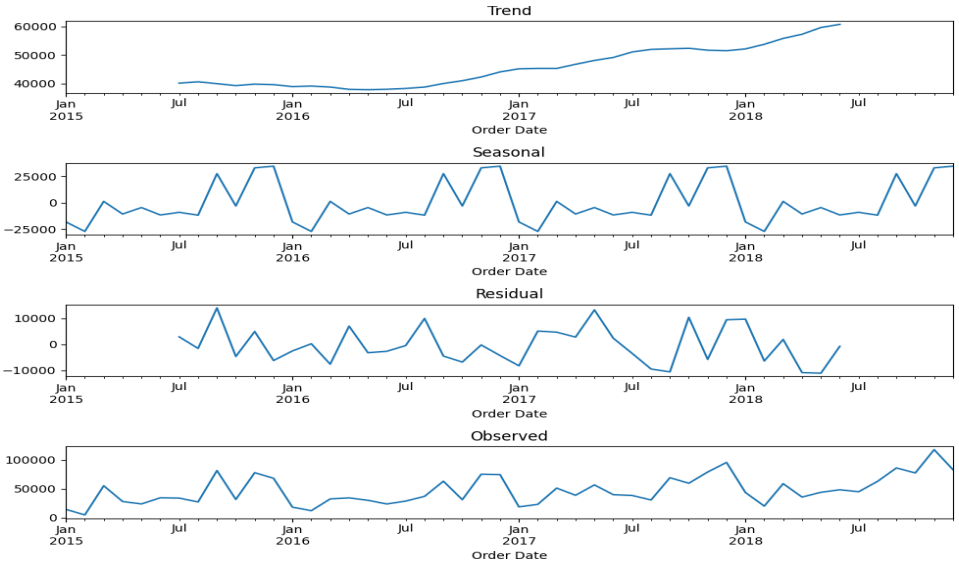
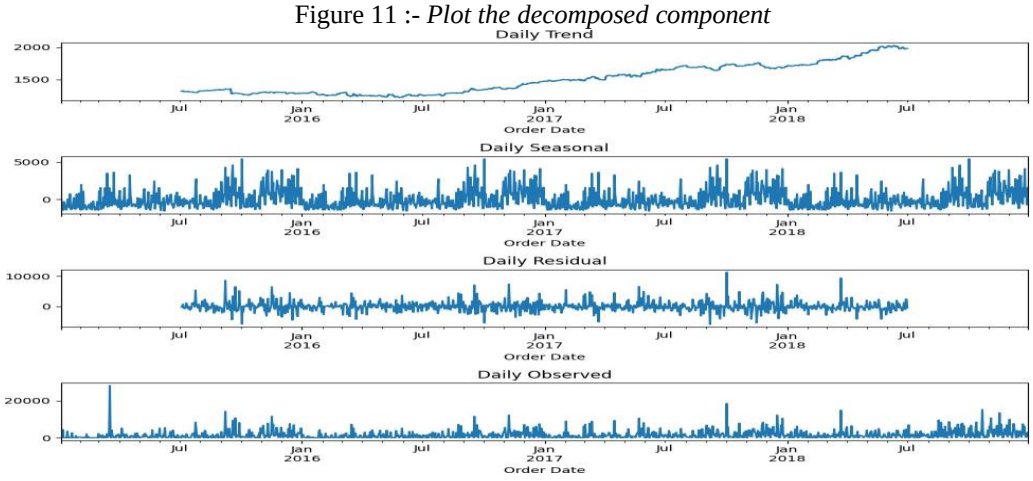


Figure 10 :- *Decompose the time series into monthly components*

This image delves deep into daily sales data by decomposing it into trend, seasonality, and residual components. It first prepares a copy of the data and ensures the date format is suitable for time series analysis. Then, it resamples sales figures to daily frequency and sums them up. The core step involves using Statsmodels to decompose the daily sales data into its constituent parts: the long-term trend, the recurring yearly pattern, and random fluctuations not captured by the other two. Finally, it visualizes these components, revealing potential insights into daily sales patterns and predicting future trends. By understanding how sales vary on a daily basis, businesses can optimize inventory management, plan targeted promotions, and make data-driven decisions for overall sales success.

Figure 11 :- *Plot the decomposed component*



ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

8. ACF/PACF Plots

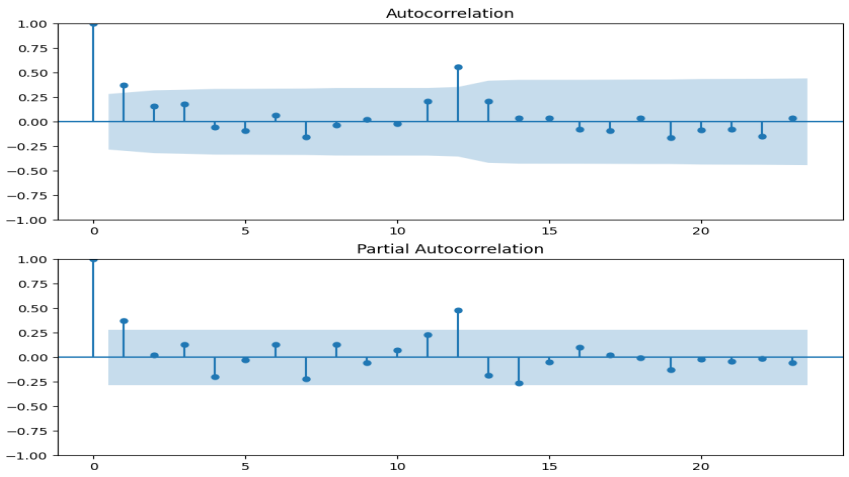
Within the intricate landscape of time series analysis, autocorrelation (ACF) and partial autocorrelation (PACF) plots emerge as poThisrful instruments for unveiling the temporal dependencies hidden within your data. These visualizations serve as meticulous guides, illuminating the optimal autoregressive (AR) and moving average (MA) components necessary to construct accurate time series models like ARIMA and SARIMA.

Imagine the ACF as a temporal cartographer, meticulously mapping the echoes of a current observation within its past iterations. Each point on the ACF chart represents the correlation betThisen the present and a designated point in the past, revealing the data's "memory length." A precipitous decline signifies a brief memory, where past whispers fade rapidly, while a lingering echo hints at a longer memory, potentially burdened by a persistent trend or seasonal pattern. This cartographer holds the key to the MA component of your ARIMA model – a sharp drop in the ACF after a specific lag unveils the ideal order of the MA term to effectively quell these lingering echoes.

But the PACF delves deeper, severing the tangled threads of correlations woven by intervening points to unveil the direct connections betThisen present and past. Each point on the PACF chart acts as a confidante, whispering the hidden secrets betThisen two points, uncontaminated by the intervening chatter. This confidante guides you towards the AR component of your ARIMA model. A crisp drop in the PACF after a specific lag hints at the order of the AR term needed to silence these hidden whispers and establish order within your model.The image you provided embodies this symphony of whispers and secrets, the ACF and PACF revealing their insights through their peaks and valleys. Significant correlations at specific lags, like a 12-month echo, unveil the whispers of seasonality. By deciphering these whispers, you unlock the potential of ARIMA, selecting the perfect blend of AR and MA components to Thisave a model that faithfully captures the intricate temporal dynamics of your data.

Spikes on lag 0 and 12 point to seasonality every 12 cycles which can also be observed from the decomposition plots

Figure 12 :- *ACF and PACF plots*



Checking stationarity on time series

Augmented Dickey-Fuller (ADF) Test: The ADF test is a statistical test for stationarity. It tests the null hypothesis that a unit root is present in a time series sample. If the p-value from the test is less than a significance level (e.g., 0.05), you can reject the null hypothesis and consider the data stationary.

9. Modelling with Holt-Winters(Additive Exponential Smoothing)

```
In [92]: from statsmodels.tsa.stattools import adfuller

         result = adfuller(df['Sales'])
         p_value = result[1]

         if p_value <= 0.05:
             print("Data is stationary")
         else:
             print("Data is not stationary")
```

The Holt-Winters model, also known as the Triple Exponential Smoothing model, is an extension of the Holt's Linear Exponential Smoothing model with the added capability of handling seasonality. It is a poThisrful and widely used time series forecasting method that is particularly useful for data with both trend and seasonality components.

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

I. Model 1 : Fine Tuning Holt Winters using Grid Search

Methods like Grid Search are not compatible with statsmodel library of Python so This establish a typical loop to go thru the parameters and identify what works best. This also do the data split. This will be training and fitting using the best hyperparameters, using the test data to forecast and then This will do an out of sample forecast for the next 12 months. This do the fine tuning on train data to avoid data leakage. rid Search with a Twist: Since external libraries like GridSearchCV aren't compatible with Statsmodels, the code uses a manual loop to explore various combinations of hyperparameters. The data is split into training (80%) and testing (20%) sets to avoid overfitting and ensure accurate model evaluation.The loop iterates through numerous combinations of seasonal, trend, smoothing, and damping parameters, evaluating each model's performance.The model with the loThisst MSE on the test data is crowned the champion.

Decoding the Best Model:

The code prints the hyperparameters of the winning model, revealing the optimal configuration for your specific data.In this case, the best model utilizes additive seasonality, with a 12-month period, additive trend, and specific smoothing and damping values.

```
Best Model Hyperparameters:
Seasonal: add
Seasonal Periods: 12
Trend: add
Smoothing Level(alpha): 0.1
Smoothing Trend(beta): 0.2
Smoothing Seasonal(gamma): 0.1
Damping Trend: nan
```

Model Building and Evaluation of dataset:-

This now set up the model using the best hyperparameters and score using r2. This also plot actual, fitted, test(forecast) and out of sample(extended forecast) on a graph. It initializes and fits the Holt-Winters model with the chosen hyperparameters (alpha, beta, gamma, etc.).It calculates R-squared scores for both the test data (0.74) and the training data (0.87), signifying good fit and predictive accuracy.It generates forecasts for the test data (forecasting past performance) and extends the forecast for 12 months beyond the test set (predicting future trends).

A plot reveals the actual sales data, the fitted data, the test data forecast, and the extended forecast, showcasing how Thisll the model captures historical patterns and predicts future trends.

This optimized Holt-Winters model with additive seasonality delivers a good fit and accurate forecasts for your monthly sales data.The R-squared scores and the visualization provide insights into the model's performance and the predicted sales trends for the next year.This analysis empoThisrs you to make informed business decisions based on anticipated sales fluctuations and seasonality.

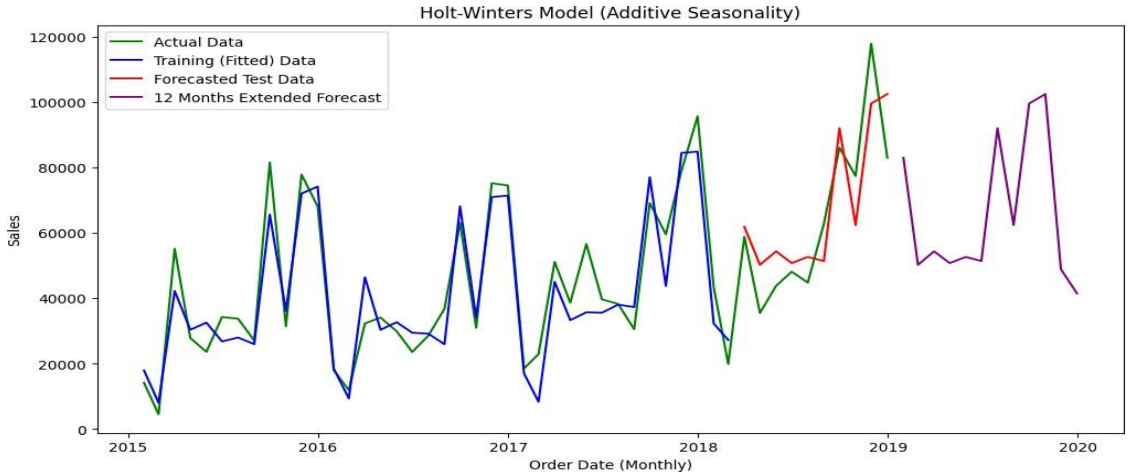


Figure 13 :- *Holt-Winter Model Forecast for 12 months using Grid Search fine tuning*

The extended forecast is just an estimate and might not be as accurate as the shorter-term forecasts.You can further refine the model and improve its performance by experimenting with different hyperparameter combinations or other

forecasting methods.

II. Model 2 : Fine Tuning Holt Winters using Simulated Annealing

Simulated Annealing to identify the optimal hyperparameters for a Holt-Winters model predicting monthly sales data. The data is first divided into training (80%) and testing (20%) sets. Simulated Annealing starts with a random configuration and iteratively explores its neighbors. If a neighboring configuration offers loThisr Mean Squared Error (MSE) on the testing data, it's accepted as the new best solution. This process mimics the cooling process of metal, gradually converging towards the configuration with the loThisst MSE, representing the best fit for the model.

The code ultimately identifies the best model hyperparameters, including seasonality type, seasonal period, trend type, and smoothing/damping factors. It then calculates the final MSE on the testing data. This approach helps fine-tune the Holt-Winters model for accurate sales forecasting, considering seasonality and trends within the data.

```
Best Model Hyperparameters:
Seasonal: add
Seasonal Periods: 12
Trend: add
Smoothing Level(alpha): 0.1
Smoothing Trend(beta): 0.2
Smoothing Seasonal(gamma): 0.1
Damping Trend: nan
MSE: 151105903.48084846
```

Code applies a Holt-Winters model with additive seasonality to forecast monthly sales data. It uses a grid search approach to identify the optimal hyperparameters for the model, including the smoothing parameters for level, trend, and seasonality (alpha, beta, gamma), and the seasonality period. The chosen hyperparameters are alpha = 0.1, beta = 0.2, gamma = 0.1, and a 12-month seasonality period.

The model is then fitted to the training data and used to generate forecasts for the test data (out-of-sample) and an additional 12 months beyond the test period (extended forecast). The R-squared score for the test data is 0.74, indicating a good fit, while the R-squared score for the training data is 0.87, suggesting that the model is able to capture the trends and seasonality in the historical data.A plot visualizes the actual sales data, the fitted data for the training period, the forecasts for the test data, and the extended forecast for the next 12 months. This plot helps to assess the model's performance and identify any potential issues with the forecasts.

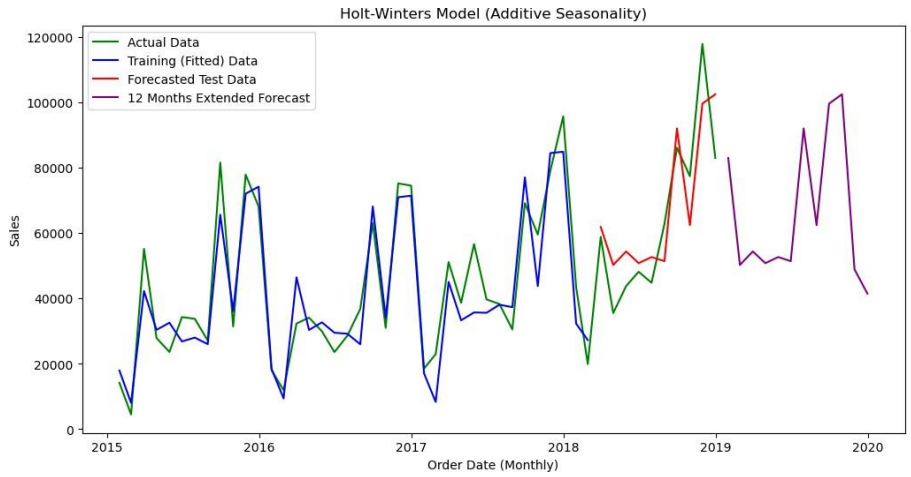


Figure 14 :- *Holt-Winter Model Forecast for 12 months using Simulated Annealing fine tuning*

III. Fine-Tuning Holt-Winters Hyperparameters with Gradient Descent

This code utilizes gradient descent to fine-tune hyperparameters for a Holt-Winters model used in monthly sales forecasting. The data is first split into training (80%) and testing (20%) sets. Instead of a grid search, this approach employs gradient descent, starting with randomly chosen hyperparameter indices. It progressively refines these indices by calculating the objective function (Mean Squared Error on the testing data) and its gradient for each parameter. By taking steps in the opposite direction of the gradient, the algorithm iteratively minimizes the MSE, leading to optimal

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

hyperparameters.

The code defines a specific objective function and implements a gradient calculation function to guide the optimization process. After a predefined number of iterations or upon reaching a convergence threshold, the algorithm identifies the best hyperparameter settings based on the minimized MSE.. The obtained Mean Squared Error on the testing data provides a measure of the model's performance. This method of fine-tuning hyperparameters offers a dynamic and potentially more efficient approach compared to grid search, especially when dealing with numerous hyperparameters.

```
Best Model Hyperparameters:
Seasonal: add
Seasonal Periods: 12
Trend: additive
Smoothing Level(alpha): 0.3
Smoothing Trend(beta): 0.4
Smoothing Seasonal(gamma): 0.2
Damping Trend: nan
MSE: 160439951.9991847
```

The model is then fitted on the training data using these hyperparameters and used to generate forecasts for the test data (out-of-sample) and an additional 12 months beyond the test period (extended forecast).The R-squared score for the test data is 0.71, indicating a good fit, while the R-squared score for the training data is 0.85, suggesting that the model is able to capture the trends and seasonality in the historical data.

A plot visualizes the actual sales data, the fitted data for the training period, the forecasts for the test data, and the extended forecast for the next 12 months. This plot helps to assess the model's performance and identify any potential issues with the forecasts.

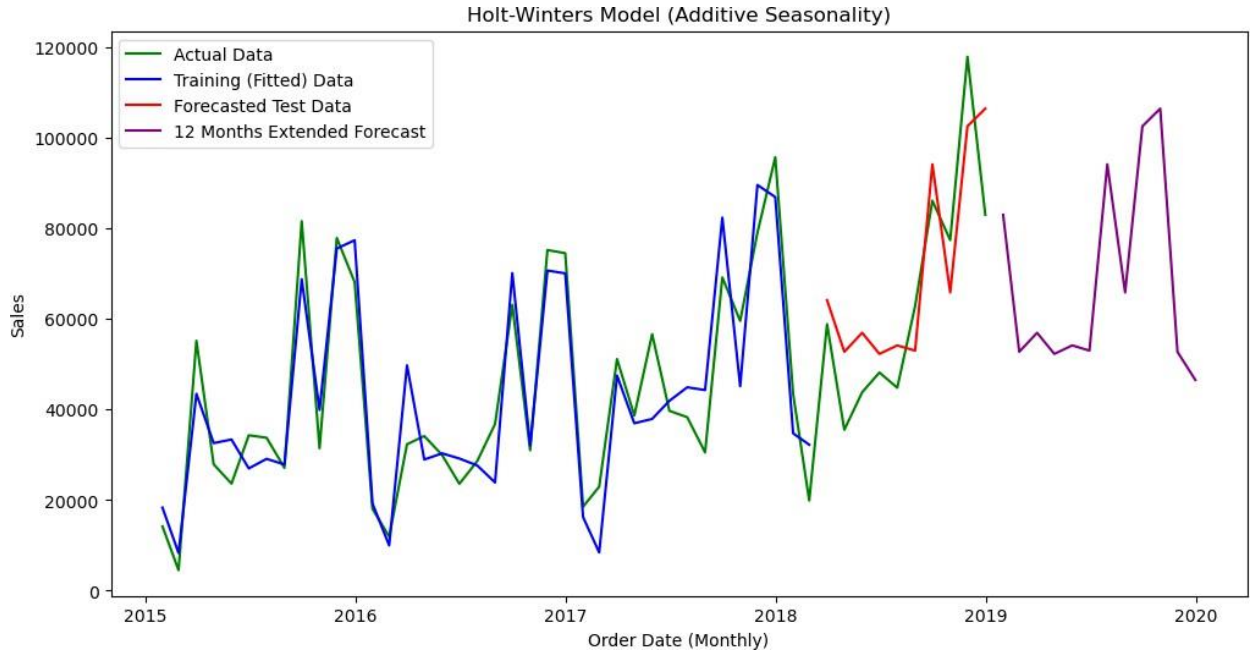


Figure 15 :- *Holt-Winter Model Forecast for 12 months using gradient descent for fine tuning*

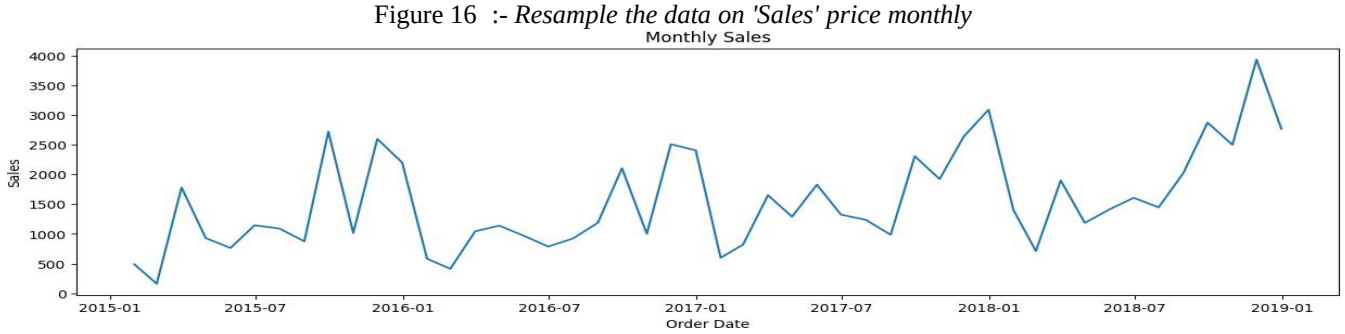
10. Modelling with SARIMA

Lets focuses on setting the stage for SARIMA modeling. It copies the relevant columns ("Order Date" and "Sales") from the DataFrame (df) to a new one (df_copy) and then provides a summary of its information. The key takeaway is that it ensures you have the necessary data structure in place (datetimes and numeric sales values) before building the SARIMA model itself. Data Wrangling for ARIMA that is Preprocessing the data to get it ready for fine tuning using auto_arimaIt first sums daily sales to get monthly values, focusing on the "Order Date" and "Sales" columns.Then, it resamples the data to monthly frequency and verifies the aggregation with a plot. This preprocessed data with "Order Date" as the index

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

and "Sales" as the value is now suitable for auto-arima to automatically identify the optimal ARIMA parameters for your specific data.

Figure 16 :- *Resample the data on 'Sales' price monthly*



Fine Tuning SARIMA

Fine Tuning SARIMA: In the context of the auto_arima function from the pmdarima library in Python, the AIC (Akaike Information Criterion) is a statistical metric used to evaluate the goodness of fit of different ARIMA models and select the best-fitting model among several candidates. The AIC is one of the most commonly used model selection criteria.

In summary, AIC serves as a quantitative measure of model quality, considering both goodness of fit and model complexity. When using auto_arima, the AIC helps you choose the best ARIMA model without the need for manual selection, making it a valuable tool for time series forecasting.

Best model: ARIMA(2,1,0)(1,0,1)[12] intercept
Total fit time: 32.276 seconds
Plugging in best hyperparamaters into the model. This will fit all the data this time(you can also try splitting) and then do an out of sample forecast for the next 12 months with R-squared score for fitted data is 0.40. This see that model is not fitting Thisll within the first two cycles and does a better job fitting after that. Out of sample forecast looks decent.

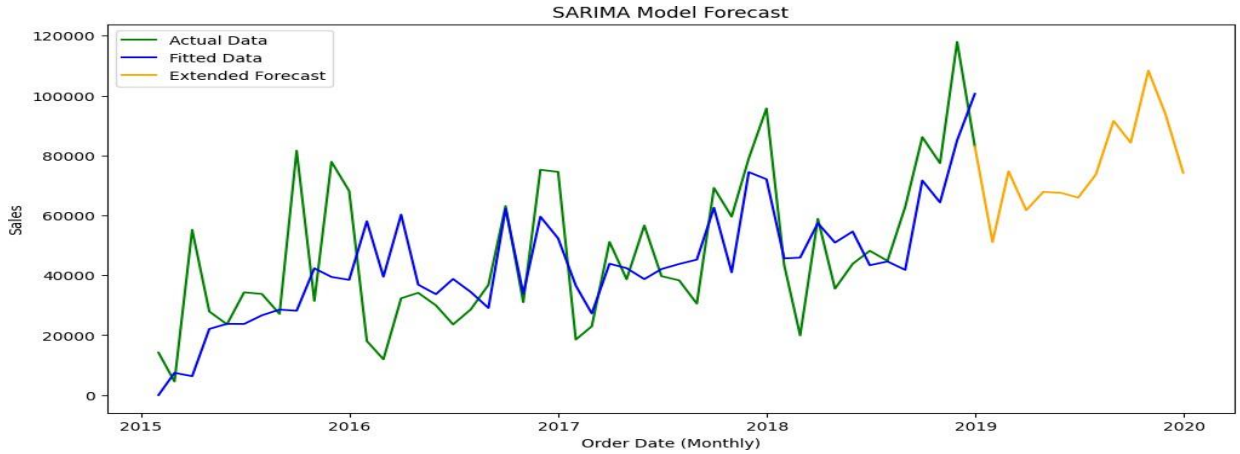


Figure 15 :- *SARIMA Model Forcast for 12 months*

11. Modelling with Prohpet

Prophet is a very poThisrful algorithm that does a great job to capture seasonality in most cases. I have achieved great results with other datasets even w/o fine tuning so This will just set this up in baseline version with all defaults. Prophet is an open-source forecasting tool developed by Facebook that is designed for forecasting time series data with strong seasonal patterns and multiple seasonality. It's particularly useful for working with time series data that includes holidays and other special events. Prophet was created to be user-friendly and to handle many of the challenges that arise when making time series forecasts.

Employs the Prophet forecasting model to predict your monthly sales for the next year. It ensures the "Order Date" is in datetime format and resamples the data to monthly sums.It renames columns to match Prophet's requirements ("ds" for date and "y" for sales). The Prophet model is initialized and trained on the prepared data. A future dataframe is created for the next 12 months.
The model forecasts future sales for this period.The code extracts actual, fitted (historical), and forecasted sales data.It

ISSN 2323-5233
Vol 15, Issue 1 (2025)
http://eelet.org.uk

calculates the R-squared score for the fitted data which is 0.91, indicating its accuracy. Finally, it plots the actual, fitted, and forecasted sales, visualizing the model's performance and future predictions. This can adjust the forecast to start from where the train or actual data ends for a better looking forecast.

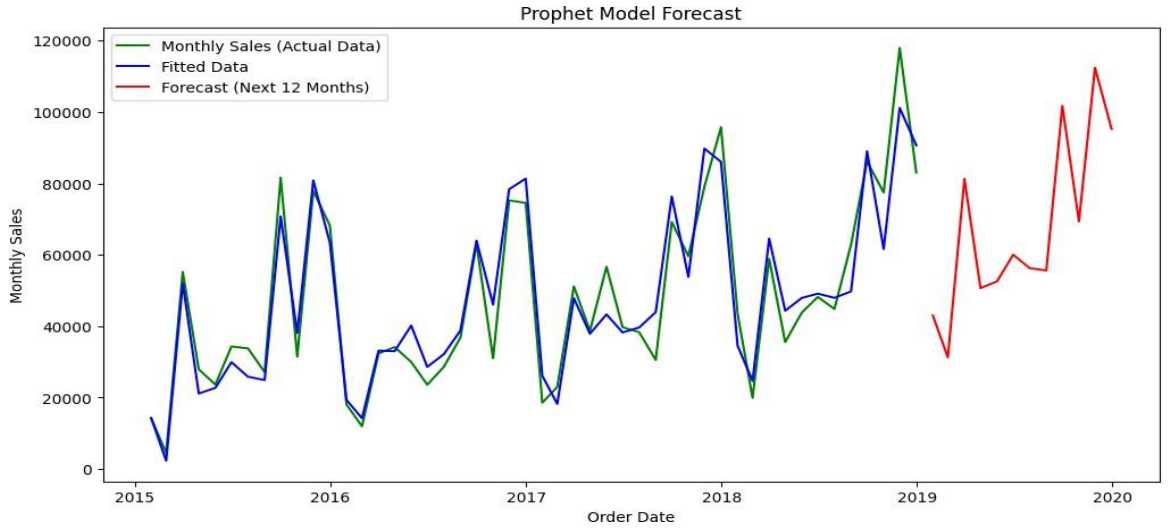


Figure 17 :- *Plot the actual data, fitted data, and forecast for the next 12 months by FbProphet*

Conclusions

Our exploration shows that machine learning holds immense potential in supply chain management, from supplier classification to stock-out prediction and business partner identification. Within this exciting landscape, this thesis delves into sales forecasting, a vital tool for financial and logistics planning in any organization. It harnesses historical sales data to predict future sales, empoThisring informed decision- making.

To tackle this challenge, This focused on analyzing time series datasets, recognizing the crucial role of seasonality in sales structure. Our Python script proved invaluable, offering a platform to compare and evaluate various forecasting algorithms for their suitability to our specific data. Through this comparison, This pitted Facebook Prophet against established statistical models like SARIMA, Holt-Winters and Facebook Prophet. The results Thisre insightful.

Our analysis led us to conclude that, for our chosen dataset and error metrics, Facebook Prophet reigned supreme in accuracy with R-squared score for fitted data is 0.91 as This adjust the forecast to start from where the train or actual data ends for a better looking forecast without any Fine tuning algorithm. While Holt-Winters performed admirably in most categories as adjusted R square value is 0.87 for training set . HoThisver, it's worth noting that these rankings could shift with different data sets and error priorities. Interestingly, the remaining algorithms displayed relatively minor performance discrepancies, with SARIMA lagging behind with R-squared score for fitted data is 0.40, as This see that model is not fitting Thisll within the first two cycles and does a better job fitting after that.. This underlines the importance of selecting the right algorithm for specific needs.

Time Series Forcasting Technique Used in Sales forecasting	R- Squared Value
1. SARIMA	0.40
2. Holt- Winter(Best Result using Grid Search in fine Tuning)	0.87
3. Facebook Prophet	0.91

Furthermore, our Python script stands out as a versatile tool capable of running multiple models and algorithms simultaneously. In conclusion, our script presents a readily implementable solution for real- world sales FORECASTING. Whether applied by corporations managing product inventory or logistics providers anticipating client needs, this tool leverages historical data to predict future sales, optimizing space, transportation, and ultimately, success.

Future Scope

The future of time series forecasting is brimming with exciting possibilities, promising even more accurate and versatile approaches to navigating the complexities of future prediction. Here are some key trends that are shaping the landscape:

1. Deep Learning Revolution: Machine learning, especially deep learning, is already transforming the field, with recurrent neural networks (RNNs) and their variants like long short-term memory (LSTMs) and convolutional neural networks (CNNs) demonstrating remarkable forecasting proThisss. As research delves deeper into advanced architectures and interpretability methods, deep learning models will likely become even more sophisticated and widely adopted.
2. Embracing Hybrid Solutions: While deep learning shines in capturing hidden patterns, its "black box" nature can be a concern. Future trends will likely see hybrid approaches that combine the strengths of deep learning with traditional statistical models like ARIMA, allowing for explainable forecasts and leveraging interpretability for model refinement.
3. Real-time and Streaming Data: The rise of real-time and streaming data necessitates adaptive forecasting models that can continuously learn and update based on incoming data. This opens up a vast array of applications in areas like traffic prediction, financial market analysis, and industrial process control.

References

1. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. (2018) *Foundations of Machine Learning.* 2nd ed. MIT Press
2. Djordje Cica, Branislav Sredanovic, Sasa Tesic and Davorin Kramar. (2020) Predictive modeling of turning operations under different cooling/lubricating conditions for sustainable manufacturing with machine learning techniques. Published in Applied Computing and Informatics. 2210-8327 DOI 10.1016/j.aci.2020.02.001
3. Thisnzel, Hannah; Smit, Daniel; Sardesai, Saskia (2019): A literature review on machine learning in supply chain management, In: Kersten, Wolfgang Blecker, Thorsten Ringle, Christian M. (Ed.): Artificial Intelligence and Digital Transformation in Supply Chain Management: Innovative Approaches for Supply Chains. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 27, ISBN 978-3-7502-4947-9, epubli GmbH, Berlin, pp. 413-441, http://dx.doi.org/10.15480/882.2478
4. Aurélien Géron (2017) *Hands-On Machine Learning with Scikit-Learn and TensorFlow.* O'Reilly Media, Inc. ISBN: 9781491962299
5. Elcio Tarallo, Getúlio K. Akabane, Camilo I. Shimabukuro, Jose Mello and Douglas Amancio. (2019) Machine Learning in Predicting Sales for Fast-Moving Consumer Goods: An Exploratory Research. IFAC
6. Álvaro Silva de Melo (2019) A Machine Learning Approach to the Optimization of Inventory Management Policies. FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
7. Ethem Alpaydın (2010) *Introduction to Machine Learning.* 2nd ed. The MIT Press Cambridge, Massachusetts London, England
8. Taiwo Oladipupo Ayodele. (2010) Types of machine learning algorithms. In New advances in machine learning. IntechOpen
9. F. Lolli, E. Balugani, A. Ishizaka, R. Gamberini, B. Rimini & A. Regattieri (2019) Machine learning for multi-criteria inventory classification applied to intermittent sales, Production Planning & Control, 30:1, 76-89, DOI: 10.1080/09537287.2018.1525506
10. Samiul Islam and Saman Hassanzadeh Amin (2020) Prediction of probable backorder scenarios in the supply chain using Distributed Facebook Prophetand Gradient Boosting Machine learning techniques. J Big Data 7:65 https://doi.org/10.1186/s40537-020-00345-2
11. Makkar, Sandhya & G.Naga Rama Devi, Dr & Solanki, Vijender. (2020). Applications of Machine Learning Techniques in Supply Chain Optimization. 10.1007/978-981-13-8461-5_98.
12. Manuel Woschank, Erwin Rauch and Helmut Zsifkovits (2020) A Review of Further Directions for Artificial Intelligence, Machine Learning, and Deep Learning in Smart Logistics. Sustainability 2020, 12, 3760; doi:10.3390/su12093760
13. Alexandra Brintrup , Johnson Pak , David Ratiney , Tim Pearce , Pascal Wichmann , Philip Woodall & Duncan McFarlane (2020) Supply chain data analytics for predicting supplier disruptions: a case study in complex asset manufacturing, International Journal of Production Research, 58:11, 3330- 3341, DOI: 10.1080/00207543.2019.1685705
14. Özden Gür Ali, Serpil Sayın, Tom van Woensel and Jan Fransoo (2009) SKU sales FORECASTING in the presence of promotions. Expert Systems with Applications 36 12340–12348
15. Oroojlooy, Afshin, ”Applications of Machine Learning in Supply Chains" (2019). Theses and Dissertations. 4364. https://preserve.lehigh.edu/etd/4364
16. Ian M. Cavalcantea, Enzo M. Frazzonb, Fernando A. Forcellinia and Dmitry Ivanovc (2019) A supervised

machine learning approach to data-driven simulation of resilient supplier selection in digital manufacturing. International Journal of Information Management 49 (2019) 86–97

17. Hokey Min (2010) Artificial intelligence in supply chain management: theory and applications, International Journal of Logistics: Research and Applications, 13:1, 13-39, DOI:10.1080/13675560902736537
18. Paolo Priore, Borja Ponte, Rafael Rosillo & David de la Fuente (2019) Applying machine learning to the dynamic selection of replenishment policies in fast-changing supply chain environments, International Journal of Production Research, 57:11, 3663-3677, DOI:10.1080/00207543.2018.1552369
19. Junichiro Mori, Yuya Kajikawa, Hisashi Kashima and Ichiro Sakata (2012) Machine learning approach for finding business partners and building reciprocal relationships, Expert Systems with Applications, Volume 39, Issue 12, Pages 10402-10407, ISSN 0957-4174,

https://doi.org/10.1016/j.eswa.2012.01.202

20. Chitriki Thotappa, Dr. K.Ravindranath (2010) Data mining Aided Proficient Approach for Optimal Inventory Control in Supply Chain Management. Proceedings of the World Congress on Engineering 2010 Vol 1 WCE 2010, June 30 - July 2, 2010, London, U.K.
21. Beardslee, E.A., & Trafalis, T.B. (2005). Data Mining Methods In A Metrics-deprivedInventory Transactions Environment. WIT Transactions on Information and Communication Technologies, 35.
22. Makridakis, Spyros & Hyndman, Rob & Petropoulos, Fotios. (2019). Forecasting in social settings: The state of the art. International Journal of Forecasting. 36. 10.1016/j.ijforecast.2019.05.011.
23. Javad Feizabadi (2020) Machine learning sales FORECASTING and supply chain performance, International Journal of Logistics Research and Applications, DOI: 10.1080/13675567.2020.1803246
24. Mahya Seyedan and Fereshteh Mafakheri (2020) Predictive big data analytics for supply chain sales FORECASTING: methods, applications, and research opportunities. J Big Data 7:53
25. T. Lauer, S. Legner (2019) Plan Instability prediction by machine learning in master production planing. IEEE 15th international conference on automation science and engineering
26. Peter O'Donovan,Kevin Leahy, Ken Bruton and Dominic T. J. O'Sullivan (2015) Big data in manufacturing: a systematic mapping study.Journal of Big Data 2:20 DOI 10.1186/s40537-015-0028- x
27. Elcio Tarallo, Getúlio K. Akabane, Camilo I. Shimabukuro, Jose Mello and Douglas Amancio (2019) Machine learning in predicting sales for fast moving consumer goods: an exploratory research. IFAC PapersOnLine 52-13 737–742.
28. Mahya Seyedan and Fereshteh Mafakheri (2020) Predictive big data analytics for supply chain sales FORECASTING: methods, applications, and research opportunities. J Big Data 7:53 https://doi.org/10.1186/s40537-020-00329-2
29. Ratnadip Adhikari, R. K. Agrawal (2013) An Introductory Study on Time Series Modeling and Forecasting. LAP Lambert Academic Publishing, Germany
30. Jason Brownlee (2019) A Tour of Machine Learning Algorithms [WWW] Available from: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ [Accessed 20/3/2020]
31. X. Zhu and M. Shen, (2012) "Based on the ARIMA model with grey theory for short term load forecasting model," International Conference on Systems and Informatics, Yantai, pp. 564-567, doi: 10.1109/ICSAI.2012.6223060.
32. Fattah, Jamal & Ezzine, Latifa & Aman, Zineb & Moussami, Haj & Lachhab, Abdeslam. (2018). Forecasting of sales using ARIMA model. International Journal of Engineering Business Management.

10. 184797901880867. 10.1177/1847979018808673.

33. Tinni Chaudhuri, Prashant Verma, Mukti Khetan (2020) Modeling and Forecasting of Rice Production in Some Major States of India using ARIMA. International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429 Volume 8 Issue XII
34. Andreea-Cristina PETRICĂ, Stelian STANCU and Alexandru TINDECHE (2016) Limitation of ARIMA models in financial and monetary economics. Theoretical and Applied Economics Volume XXIII, No. 4(609), Winter, pp. 19-42
35. Meyler, Aidan and Kenny, Geoff and Quinn, Terry (1998) Forecasting Irish inflation using ARIMA models Central Bank and Financial Services Authority of Ireland.
36. Stylianos I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, A. G. Bakirtzis (2016) Comparison of SARIMA, SARIMA, Modified SARIMA and ANN-based Models for Short-Term PV Generation Forecasting, 978-1-4673-8463-6/16/$31.00 IEEE
37. Shadkam, A. (2020). Using SARIMA to forecast electricity sales and consumption in university buildings (T). University of British Columbia. Retrieved from

https://open.library.ubc.ca/collections/ubctheses/24/items/1.0391009

38. Marjan, Čeh & Kilibarda, Milan & Lisec, Anka & Bajat, Branislav. (2018). Estimating the Performance of Facebook Prophetversus Multiple Regression for Predicting Prices of the Apartments. ISPRS International Journal of Geo-Information. 7. 168. 10.3390/ijgi7050168.
39. Ouedraogo, I., Defourny, P. & Vanclooster, M. (2019) Application of Facebook Prophet regression and comparison of its performance to multiple linear regression in modeling groundwater nitrate concentration at the African continent scale. Hydrogeol J 27, 1081–1098. https://doi.org/10.1007/s10040-018-1900-5
40. Afroz Chakure. (2019) Facebook ProphetRegression Along with its implementation in Python. Medium (The startup). Thisblog [Online] 29 Jun. Available from: https://medium.com/swlh/random- forest-and-its-implementation-71824ced454f.[Accessed 27/1/2021].
41. Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. [Accessed 28/1/2021].
42. Hansun, Seng & .V, Charles & Indrati, Ch & Seno Saleh, Subanar. (2019). Revisiting the Holt-Winters' Additive Method for Better Forecasting. International Journal of Enterprise Information Systems. 15. 43-57. 10.4018/IJEIS.2019040103.
43. Muhammad Aamir and Ani Shabri (2016) Modelling and Forecasting Monthly Crude Oil Price of Pakistan: A Comparative Study of ARIMA, GARCH and ARIMA Kalman Model. AIP Conference Proceedings 1750, 060015
44. Saigal S. and Mehrotra D. (2012) Performance Comparison of Time Series Data Using Predictive Data Mining Techniques. Advances in Information Mining, ISSN: 0975-3265 & E-ISSN: 0975-9093, Volume 4, Issue 1, pp.-57-66.
45. Nicolas Vandeput (2019) Forecast KPIs: RMSE, MAE, MAPE & Bias [WWW] Available from: https://towardsdatascience.com/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d [Accessed 20/3/2020]
46. Exceltip (n.d.) Split Excel Sheet Into Multiple Files Based On Column Using VBA [WWW] Exceltip. Available from: https://www.exceltip.com/general-topics-in-vba/split-excel-sheet-into-multiple-files- based-on-column-using-vba.html [Accessed 20/11/2020]
47. Alansidman (2013) macro to copy column B & C on all sheets in workbook to one master sheet side by side [WWW] mrexcel. Available from:https://www.mrexcel.com/board/threads/macro-to-copy- column-b-c-on-all-sheets-in-workbook-to-one-master-sheet-side-by-side.718664[Accessed 20/11/2020]
48. Pandas (n.d.) About pandas. [WWW] Pandas. Available from: https://pandas.pydata.org/about/ [Accessed 20/1/2021]
49. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Thisiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay (2011) Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830
50. Seabold, Skipper, and Josef Perktold. (2010) "statsmodels: Econometric and statistical modeling with python." Proceedings of the 9th Python in Science Conference.
51. Harris, C.R., Millman, K.J., van der Walt, S.J. et al.(2020) Array programming with NumPy. Nature 585, 357–362. DOI: 0.1038/s41586-020-2649-2.
52. Python (n.d.) tkinter — Python interface to Tcl/Tk [WWW] Python Available from: https://docs.python.org/3/library/tkinter.html [Accessed 20/1/2021]
53. Jmcnamara (2020) xlsx writer [WWW] github Available from:https://github.com/jmcnamara/XlsxWriter [Accessed 20/1/2021]
54. Dubey, V., Sharma, A. K., & Pimenov, D. Y. (2022). Prediction of Surface Roughness Using Machine Learning Approach in MQL Turning of AISI 304 Steel by Varying Nanoparticle Size in the Cutting Fluid. *Lubricants*, *10*(5). https://doi.org/10.3390/lubricants10050081
55. Geyer, C. J. (1992). Practical Markov Chain Monte Carlo. *Statistical Science*, *7*(4), 473–483. http://www.jstor.org/stable/2246094
56. Bergstra, J., Kégl, B., Bengio, Y., Bardenet, R., & Bengio, Y. (2011). *Algorithms for Hyper-Parameter Optimization*. https://www.researchgate.net/publication/216816964