**Project Title**

**TIME SERIES SALES FORECASTING**

**A data-driven approach to forecast upcoming sales using time-based analysis and predictive modeling**

**School of Computer Science Engineering and Technology**

|  |
| --- |
| **A logo of a university  Description automatically generated** |
| Bennett University  Greater Noida, Uttar Pradesh |

Submitted By Aanchal Mittal - E22CSEU0617

Utkarsh Tripathi - E22CSEU1501 Vishal Thakur - E22CSEU1500

Submitted to Mala Saraswat

1. **Introduction and Objective**

**Objective of the Project**

### The objective of this project is to develop an interpretable and stable time series forecasting model that can predict future sales from historical Amazon sales data which mainly includes clothing . The model will use daily sales patterns and seasonality to inform business decisions regarding inventory management, promotion planning, and revenue forecasting.

### ****Problem Statement****

### As dynamic is the current e-commerce situation, precise forecasting of sales has become the need of the hour not to end up with overstocking, understocking, and wasteful utilization of resources. The dynamic nature of demand from the consumers is hard to manage on the part of the businesses, the course of which may be guided by drivers like seasonality, promotional events, and distribution performance. This project seeks to tackle these issues by utilizing time series analysis on historical sales data. The forecasting models will dig into past information to reveal patterns and trends, ultimately offering reliable predictions for the next 30 days.

### ****Feasibility Justification****

### This project is feasible and can be done under the following conditions:

### The data set includes day-to-day sales for a continuous time period, thus allowing us to make time-based predictions.

### Our EDA (Exploratory Data Analysis) shows clear trends in the data, as well as clear seasonality. Thus, this allows the forecasting techniques to make use of ARIMA (Autoregressive Integrated Moving Average), SARIMA (Seasonal Autoregressive Integrated Moving-Average), or ETS (Exponential Smoothing State Space with Trend and Seasonality).

### The forecasting initiative has the backing of sound business rationale and real-world use for sales optimization and demand planning.

### We will employ measures such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to gauge model accuracy and enable comparisons.

### ****Data Collection and Preprocessing****

### Dataset Description

### Data used for this research is drawn from an Amazon Sales Report, which has transaction-level sales data. It includes several important variables such as:

### Date – The date on which all orders were effectively placed.

### Amount – Overall sales amount of the transaction.

### Qty – Quantity of items ordered.

### Category – The category of product (e.g., clothing items, electronics).

### Sales Channel – Whether the sale was online or offline.

### Courier Status – Indicates whether or not the order was delivered.

### Promotion Ids – Promotion or discount details applied.

### The data is for the period 03-31-22 to 06-29-22.

### In an effort to achieve our goal of predicting daily sales, information was gathered each day by adding the total amount for each day.

### Preprocessing Steps

### In order to prepare the data for prediction and analysis, the following pre-processing steps were executed:

### 1. Date Conversions and Indexing

### Date was also redefined as datetime and was used as the index to facilitate easier time-based operations. This was for simpler resampling and implementation of time series models.

### 2. Daily Aggregation

### The data, originally at transaction level, was resampled with.resample('D').sum() to get total daily sales. This provided a clean, unbroken time series (df\_daily) ready for forecasting.

### 3. Missing Value Removal

### Missing dates within the day-to-day series were forward-filled in order to maintain continuity.

### Missing or unknown values in categorical columns like promotion-ids and Courier Status were imputed with 'Unknown'.

### 4. Duplicate Elimination

### All duplicate entries were removed to preclude bias in training and testing.

### 5. Outlier Detection and Handling

### Outliers in the Amount column were treated using the Interquartile Range (IQR) method. Extreme values were capped to avoid skewing the model.

### 6. Engineering Features

### To improve the dataset and forecast, several columns were added, including Month, Weekday, and Is\_Weekend, which represent weekly and seasonal sales trends.

### Unit Price expressed as Amount / Qty.

### Is\_Promotion - A binary flag that indicates whether a promotion was applied

### Delivery Success - A binary variable based on courier status

### These allow the algorithms to detect additional trends that affect sales.

### 7. Stationarity Check

### The Augmented Dickey-Fuller (ADF) test was used to check for stationarity in the time series. This made it easier to determine if differencing was required for ARIMA models.

After all preprocessing steps, we obtained a clean and enriched daily time series dataset. It was now ready for Exploratory Data Analysis (EDA) and the application of forecasting models like ARIMA, SARIMA, and ETS. These steps ensured that the data was accurate, consistent, and suitable for reliable prediction.

### ****Time Series Modeling and Diagnostics****

### Model Selection and Fitting

### For this project, three time series forecasting models were considered and implemented:

### 1. Exponential Smoothing (ETS)

### ETS is a simple but effective method of short-term forecasting. It was selected as the baseline model so that the frequent patterns and seasonality in sales could be detected.

### Why chosen: Automatically detects level, trend, and seasonality in the series.

### Fitting: Fitted using Holt-Winters method with trend and additive seasonality.

### Output: Forecasted 30-day sales with a smooth seasonal pattern

### 2. ARIMA (AutoRegressive Integrated Moving Average)

### ARIMA was employed in fitting non-seasonal patterns in data.

### Why used: Most suitable for univariate time series with trend and autocorrelation, particularly when the data is reduced to stationarity through differencing.

### Chosen order: (p=4, d=0, q=2) based on PACF/ACF plot and ADF test statistics.

### Fitting: The model was fit to the full set of daily sales (df\_daily) and used to predict the next 30 days.

### Key Output: AR and MA coefficients that indicated moderate lagged relationships in sales.

### 3. SARIMA (Seasonal ARIMA)

### SARIMA is a type of ARIMA with the inclusion of seasonal patterns, which were evident in the data (e.g., weekly spikes in sales).

### Why chosen: Weekly seasonality (7-day cycle) of sales data existed and was obviously modelable with SARIMA.

### Order selected: (p=2, d=0, q=4)(P=2, D=0, Q=4, s=7)

### Fitting: Learned on the entire time series and implemented to predict 30 days.

### Key Parameters: Achieved season and non-season coefficients, both identifying trend and repeating sales patterns.

**4. Prophet (by Facebook)**

Prophet is an efficient forecasting tool created by Facebook for time series data that have strong seasonal and trend patterns, and holiday effects.

* **Why chosen:** Simple to use, resistant to missing data and outliers, and great for business use cases like sales forecasting.
* **Fitting:** Prophet was fitted to the cleaned up df\_daily and made forecast for the next 30 days.
* **Key Function**: The Prophet automatically breaks down time series as:
* Trend: Long-term rise or fall.
* Weekly Seasonality: This macro-economic factor capturing the weekly sales cycle.
* Yearly Seasonality: Would be able to model variations that occur over a year (not applicable in this case).
* **Output**: 30days forecast and their confidence intervals and decomposed trend + seasonal components.decomposed trend + seasonal components.

**5. LSTM (Long Short-Term Memory)**

LSTM is one of the deep models that are particularly well-designed to handle time-based data by learning from past sequences of observations.

* **Why chosen:** LSTM is particularly helpful to identify long-term and non-linear temporal trends in time series, possibly not identified by typical models.
* **Fitting:** The sales data was normalized using MinMaxScaler.

A sliding window of 30 days of past sales was used to predict the next day's sales.

The model structure contained: One LSTM layer with 64 units, One Dense output layer, Trained on 50 epochs with Adam optimizer

* **Key Input Shape:** (samples, 30 timesteps, 1 feature)
* **Output:** The model estimated future 30 days of sales by learning trends from previous sequences. The model was compared using common metrics like MAE, RMSE, and MAE.

**Model Diagnostics**

### Diagnostic testing was carried out after model fitting and to test for model assumptions.

### 1. Residual Analysis

### Residuals were calculated as a difference between actual and predicted values. Residual plots were created for each of the three models to check randomness. Residuals should ideally look like white noise — no trends, no patterns.

### 2. Autocorrelation of Residuals

### The plot was checked for residual Autocorrelation Function (ACF). An adequate model would not leave significant lags in the residual ACF. In our special case, SARIMA residuals were characterized by a low autocorrelation, which demonstrates a good fitting.

### 3. Normality of Residuals

### Histogram and KDE plot of residuals was done to test for normality of errors. The residuals from the SARIMA model were approximately bell-shaped, indicating approximate normality.

### ****Forecasting and Evaluation****

### Forecasting

### After fitting the models (ETS, ARIMA, SARIMA, Prophet, and LSTM), a sales forecast of 30 days was also made from all the models. Features were generated using historical data on daily sales. Models were fitted using the entire historical daily sales time series and tested on the last 30 days of the actual sales (test\_data).

### Each model produced future values based on different strengths:

### The ETS captured level, trend, and seasonality.

### Non-seasonal trends have been modelled using autoregression in ARIMA.

### SARIMA integrated trend and the weekly seasonality.

### Prophet managed trends, weekly cycles, and all their uncertainties.

### LSTM obtained patterns of the previous sales sequences through deep learning.

### Evaluation Metrics

### To assess forecast accuracy, the following metrics were calculated for each model:

### MAE (Mean Absolute Error): Average magnitude of errors

### MSE (Mean Squared Error): Penalizes larger errors more than MAE.

### RMSE (Root Mean Squared Error): Square root of MSE; easier to interpret in the same unit.

### MAPE (Mean Absolute Percentage Error): Percentage-based accuracy, useful for business stakeholders.

### Comparison Table

|  |  |  |  |
| --- | --- | --- | --- |
| MODEL | MAE | MSE | RMSE |
| ETS | 224727.32 | 58688980187.13 | 242258.09 |
| ARIMA | 168320.47 | 43043767458.70 | 207469.92 |
| SARIMA | 256484.01 | 92175289514.32 | 303603.84 |
| PROPHET | 48038.00 | 3759777383.13 | 61317.02 |
| LSTM | 216.2565 | 77778.3516 | 278.8877 |

### After evaluating all five models using MAE, MSE, RMSE, and MAPE, we observe the following:

### LSTM achieved the lowest MAE (216.26) and lowest MSE (77,778.35) among all models, along with a very low RMSE (278.88), making it the most accurate model in terms of minimizing absolute and squared prediction errors.

### Prophet did well too, having RMSE (61,317.02) considerably lower, and a fair MAPE (6.59%) therefore acceptable attractive accuracy of interpretable seasonal-patterns.

### The ARIMA and SARIMA models performed worse errorwise, likely because they were less capable of deal with non regular or more complex patterns in sales.

### ETS yielded the largest RMSE (242,258.09) and was the worst performing in this data set.

### Here are some graphs:

### Actual vs Forecasted Sales

### 

### Figure 1: Actual vs Forecasted Sales by Different Models

### This plot compares the actual sales with the 30-day forecast from all models (ETS, ARIMA, SARIMA, Prophet, LSTM). It helps visually evaluate which model closely tracks the real sales trend.

### Prophet Forecast Plot

### 

### Figure 2: Prophet Model – 30-Day Sales Forecast with Confidence Intervals

### This graph shows the Prophet sales prediction along with uncertainty bounds. The shaded region gives the 95% confidence interval, providing an idea of predictability.

### Prophet Components Plot

### 

### 

### Figure 3: Prophet Forecast Components – Trend and Weekly Seasonality

### This Prophet decomposition graph divides the forecast into weekly seasonality and long-term trend. We can see how patterns that repeat affect daily sales.

### LSTM vs Actual Sales Plot

### 

### Figure 4: LSTM Predictions Compared to Actual Sales

### This plot indicates the disparity between the prediction by the LSTM model and actual sales over the past 30 days. The closer the lines are to each other, the better learning from past sequences.

### Model Error Comparison Bar Chart

### 

### Figure 5: Comparison of RMSE across Forecasting Models

### This bar graph gives a summary of the RMSE of all the models and shows that LSTM recorded the lowest error, justifying it as the best-performing model in this project.

### ****Discussion and Conclusion****

### This project applied and compared five time series forecasting models — ETS, ARIMA, SARIMA, Prophet, and LSTM — to predict daily Amazon sales for a 30-day future horizon.

### In comparison with all models, LSTM resulted in the lowest error values (MAE = 216.26, RMSE = 278.88) and showed that deep learning performed best in replicating past patterns of sales.

### Prophet also did a good job, with a MAPE of 6.59%, and is an excellent model in terms of interpretability and trend-seasonality decomposition.

### Traditional statistical models like SARIMA and ARIMA worked moderately well but had larger error values since they provided linear assumptions and were unable to handle totally non-linear trends.

### Overall, the comparison of the numbers indicated that the most accurate model was LSTM followed by Prophet with more business-focused interpretation in terms of trend and seasonality factors.

### Implications of Forecasting

### The 30 day forecast to the right shows insights to take action on, for example, with inventory, promotions, and logistics decisions.

### Weekly seasonality detected by Prophet and SARIMA can inform seasonal selling or promotions based on the most popular days.

### Given its low forecast error, LSTM is the best method for short-run planning, especially when one can find stable historical patterns.

### Limitations

### Data-Only Driven: These models are based solely on historical sales, do NOT include any other external variables (economic factor, holiday, etc.) Holiday, promotion, stock out, etc. are NOT explicitly provided but can be essential to be able to generalize well.

### LSTM Requires More Data: LSTM needs more dataLSTM worked best in our case but it usually needs a large amount of data to generalize well and is less interpretable than statistical models.

### Assumes Pattern Continuity: Assumes Pattern Continuity: All models assume the future patterns and seasonality would be like the past, which is not always the case in the unusual market times.

### Conclusion

### This study presented application of traditional (classical) and modern time series models in short-term daily e-commerce sales forecasting.

### The LSTM model was the most data explanatory, and the Prophet model is data descriptive. Applied appropriately, these predictive solutions can greatly assist data-informed decision-making in retail as well as business operations.

**References**

1. Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed.). OTexts.  
   <https://otexts.com/fpp2/>  
   *Used for understanding ETS (Exponential Smoothing) and ARIMA modeling techniques.*
2. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.  
   *Theoretical background for ARIMA and SARIMA models.*
3. Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45.  
   <https://doi.org/10.1080/00031305.2017.1380080>  
   *Primary reference for the Prophet forecasting model by Facebook.*
4. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.  
   <https://doi.org/10.1162/neco.1997.9.8.1735>  
   *Original paper on the LSTM (Long Short-Term Memory) model used for deep learning forecasting.*
5. Kaggle. (n.d.). *Amazon Sales Dataset*. Retrieved from [https://www.kaggle.com](https://www.kaggle.com/)  
   *The source of the transaction-level sales dataset used in this project.*
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.  
   *For preprocessing, scaling, and model evaluation metrics like MAE, MSE, RMSE.*
7. Chollet, F., & others. (2015). *Keras*. GitHub repository. <https://github.com/keras-team/keras>  
   *Used for implementing the LSTM deep learning model.*
8. McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51–56.  
   *Used for data handling with Pandas.*