

Sales Forecasting and Analysis for Retail Businesses

KAMARAJUGADDA SREE KALYANI¹, GURRAM MANASWI², KANKANALA KRISHNA BHAVITHA³, BATTALA KRISHNA BHARAT⁴

^{1,2,3,4}Department of Electronics and Communication Engineering Sreenidhi Institute of Science & Technology, Hyderabad, Telangana, India

Abstract- Sales forecasting and analysis are crucial in business decision-making because they allow organizations to predict future trends and streamline their strategies. The following article introduces a hybrid model which combines Facebook Prophet and XGBoost to enhance the accuracy of the sales prediction. Prophet, being a time-series forecasting tool, excels in detecting seasonality and trends, and XGBoost, a powerful machine learning algorithm, finds complex relationships and non-linear trends in sales information. The technique suggested leverages past sales transaction history to predict future revenues for the next month. The combined model houses the strengths of both worlds, with Prophet doing time-series decomposition, and XGBoost refining residual trends, ending up with more accurate forecast results. The performance is gauged with Mean Squared Error (MSE) and Mean Absolute Error (MAE) to establish its accuracy. Experimental outcomes show the hybrid method exceeds the performance of individual models when forecasting accuracy and hence is an appealing solution to retail companies aiming for data-based decision-making. The hybrid model can be made to include extrinsic variables such as promotions, holidays, and economic indicators, thus further enhancing prediction performance. The above changes make the approach sufficient for dynamic retail environments where demand patterns change over time. Moreover, investigating real-time forecasting possibilities and automation can give businesses real-time insights, allowing pre-emptive decision-making and strategic planning in a competitive marketplace.

Indexed Terms- Sales Forecasting, Hybrid Model, Prophet, Xgboost, Forecast Accuracy

I. INTRODUCTION

Forecasting and analysis of sales are of utmost significance to retail companies so that they can make appropriate decisions by knowing what happened yesterday and what would happen tomorrow. Companies can identify patterns in customer behavior, seasonality, and effects of variables such as promotion, price change, and general economic conditions using past sales information. Successful inventory control, price competitiveness, and promotional activity result in increased profitability by utilizing such analysis and Forecasting refers to the extrapolation of future sales using statistical and machine learning methods. Forecasting accurately assists businesses in managing the inventory effectively, reducing stockouts, overstocking, save resources, and maximize more coherent marketing campaigns. Using more sophisticated data analysis and AI capabilities, retailers can now exploit more sophisticated forecasting techniques utilizing multiple variables such as historical sales, economic factors, weather, and competitor activity.

Sales forecasting models are generally available in three types: statistical, machine learning, and mixed models. Statistical models such as ARIMA and exponential smoothing (such as Holt-Winters) perform well with structured, time-series data and are appreciated for being easy to understand and interpret. Linear regression will typically be used when external influences, as well as the like advertising or economic, trigger sales results.

Machine learning models provide a stronger and more stable solution, particularly with difficult and high-value data. XGBoost, for instance, can detect non-linear feature correlations, while Random Forests have high performance even with noisy or missing data. Deep learning algorithms like LSTMs

have a specific talent for dealing with sequential sales data, and CNNs can identify patterns and outliers in time series.

Hybrid models integrate the strengths of both machine learning and statistical methods to enhance predictive accuracy. One well-known example is employing Prophet to capture trends and seasonality and XGBoost to capture external influences. Likewise, stacking ARIMA with neural networks captures both linear and non-linear trends. Stacked models take it further by stacking models on top of each other, where the output of one model feeds into the next, increasing predictive capability.

In the end, the model selection hinges on data complexity, forecasting horizon, and enterprise-specific goals. Statistical models prove best in structured data, but machine learning and hybrid models deliver higher accuracy for multi-variable, unstructured data case.

II. LITERATURE SURVEY

Table I. RESEARCH TRENDS IN FORECASTING

Ref. No	Year of publication	Key Concepts Used
1	2018	Facebook Prophet, additive model, trend/seasonality/holiday effects, business forecasting, outlier handling
2	2024	XGBoost, TimesFM, retail sales forecasting, machine learning, MAE/RMSE reduction.
3	2024	XGBoost, Prophet, ARIMA, retail forecasting, data challenges, Rossmann dataset
4	2023	XGBoost linear regression, sales prediction, Favorita dataset, seasonality, Kaggle
5	2021	Hybrid model, LSTM + XGBoost, time series prediction, complex pattern handling

6	2019	Hybrid model, clustering, ARIMA, XGBoost, cross-border e-commerce, sales forecasting
---	------	--

Taylor, S. J., & Letham, B. (2018) [1] Taylor and Letham (2018) citation present Prophet, which is a highly effective open-source forecasting tool by Facebook to handle time series data effectively, chiefly for business usage. The writers present an additive model with trend, seasonality, and holiday effects to achieve strong forecasts even with missing values and outliers. Prophet is meant to be extremely flexible and user-friendly such that analysts with little experience in time series modeling can use it. Its scalability and stability have made it popular across all sectors for forecasting purposes such as sales, website traffic, and resource planning.

Prathana Dankorpho (2024) [2] The reference for Prathana Dankorpho (2024) is to her master's thesis research to enhance sales forecasting in retail using emerging machine learning methods. In her "Sales Forecasting for Retail Business using XGBoost Algorithm and TimesFM," which she submitted to the Department of Computer Engineering, Chulalongkorn University, Bangkok, Dankorpho explains the limitation of traditional statistical models to handle the dynamic situation of retail contexts. By using eXtreme Gradient Boosting (XGBoost) and TimesFM, she shows significant improvement in predictive power with an average reduction of 29.23% in Mean Absolute Error (MAE) and 34.54% in Root Mean Squared Error (RMSE) compared to conventional methods.

Malik, Saira, et al. (2024) [3] Malik, Saira, et al. (2024) refer to their research paper entitled "Sales Forecasting Using Machine Learning Algorithm in the Retail Sector" in the Journal of Computing & Biomedical Informatics. In their research, the authors discuss the intricacies of sales forecasting in retail, specifically data constraints, missing values, and outliers' problems. They describe employing machine learning algorithms—Extreme Gradient Boosting (XGBoost), Facebook Prophet (FB Prophet), and Autoregressive Integrated Moving Average (ARIMA)—to predict and model sales trends from the Rossmann sales data.

Li, Kunluo (2023) [4] The reference by Kunluo Li (2023) is to the research paper entitled "A Sales Prediction Method Based on XGBoost Algorithm Model," published in the BCP Business & Management journal. Li presents here an XGBoost linear regression model that is aiming to enhance the accuracy of large chain stores' sales forecasting. The model, working with the Ecuadorian Favorita dataset during a competition on Kaggle, is solving seasonality and data scarcity problems by constructing a number of training samples and testing sets in order to estimate performance.

Wei, He, and QingTao Zeng (2021) [5] In referencing the limitations of conventional time series predictive models, whose accuracy is poor in handling complex patterns as well as delay issues, authors suggest a mixed model based on the integration of LSTM neural networks with XGBoost to improve accuracy in predictability. The LSTM module possesses a very strong capacity to capture sequential data temporal relations, and the XGBoost module possesses a strong capacity to capture nonlinear interactions and high-efficiency capturing of feature interactions.

Ji, Shuang, et al. (2019) [6] Ji, Shuang, et al. (2019) present a novel and highly efficient sales forecasting approach for the scenario of cross-border e-commerce enterprises. Recognizing the particular challenges these businesses face—i.e., changing patterns of demand, global market subtleties, and seasonality variances—the authors designed a three-stage hybrid forecasting model merging clustering, the ARIMA model, and the XGBoost algorithm. The intention was to increase forecasting accuracy by taking advantage of the respective strengths of each model element.

III. METHODOLOGIES

A. Using Prophet Model

Meta's Prophet model is employed to predict future sales by tracking trends, seasonality, and special events (e.g., holidays). It works particularly well on time-series data with strong seasonal patterns and historical anomalies

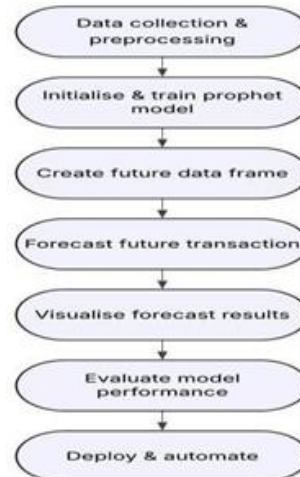


Fig. 1. Prophet model flow chart

Data Collection & Preprocessing:

This is the first phase whereby applicable data (i.e., sales time series data) is retrieved from sources like databases, spreadsheets, or APIs. Data is cleaned and preprocessed—this could include dealing with missing values, date column formatting, exclusion of unwanted records, and having the data organized with a time dimension (ds) and a target variable (y) that is Prophets-ready.

Initialize & Train Prophet Model:

In this step, a Prophet model is created by initializing it with the Prophet () function of the Facebook Prophet library. The filtered time series data is then passed to the model through the .fit() function. Prophet automatically does seasonality, trends, and holidays, so this step makes foundations right for forecasting.

Create Future Data Frame:

After training the model, the second thing is to specify how many periods beforehand the forecasts are required.

This is done using the .make_future_dataframe() function, which provides a DataFrame extended from the original dataset to an imaginary number of future periods (i.e., days, weeks, or months).

Forecast Future Transactions:

The future DataFrame is passed to the trained model to make predictions. This is done by the use of the .predict() function, which not only returns the

forecasted values ($\hat{y}$) but also uncertainty intervals ($\hat{y}_{lower}$, $\hat{y}_{upper}$).

Visualize Forecast Results:

Graphic representation of the projected outcome is vital in interpretation and making choices. Prophet offers simple plotting functionality that displays the forecasted trend through time, the original data points, and the bands of uncertainty. Moreover, component plots decompose the forecast into trend, weekly and yearly seasonality, and holidays (if included), providing better understanding of what contributes to the predictions.

Test Model Performance:

Following forecasting, the outcomes are then plotted using Prophet's intrinsic plotting functions (`plot()`, `plot_components()`). The plots enable easier interpretation of trends, seasonality, and alignment of the forecast with past observations, hence allowing stakeholders to easily understand the outcomes. To ensure that the model has been rendered accurate, the performance of the model is checked by testing it against previously held-out data or through cross-validation. Mean Absolute Error (MAE), Mean Squared Error (MSE), are some of the performance measures used to determine to what extent accurately the model can predict using new data.

Deploy & Automate:

In the final step, the model is deployed into production—this can either be done by deploying it to a web application, dashboard, or pipeline with scheduled scripts, Streamlit, or Flask. Automation allows for the forecasting to be automatically deployed to new data without any direct human intervention and updating the forecasts in real-time.

B. Using XGBoost Model

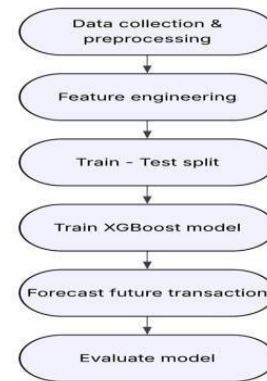


Fig. 2. XGBoost model flow chart

Data Collection & Preprocessing:

The process starts with gathering time series historical data—typically containing a date and value column (e.g., `ds` for dates and `y` for values like sales or transactions). Preprocessing involves data cleanup: deleting duplicates, dealing with missing values, fixing outliers, and organizing the data into a compatible structure with Prophet.

The information should be in chronological sequence and consistent so that the model can identify beneficial patterns.

Set up & Train Prophet Model:

After the dataset is ready, Prophet model is developed using Facebook's open-source library called Prophet. The model is tuned by fitting it to historical data. In this step, Prophet identifies and models changes in trends, seasonality (weekly, yearly, etc.), and holidays if they are specified. The model learns the patterns underlying the data that will subsequently be utilized to make predictions for the future.

Create Future Data Frame:

In order to understand and forecast ahead in the future, a new Data Frame is built that provides output beyond the boundaries of the original data. This is achieved by Prophet's `make_future_dataframe()` method, which declares the number of future periods to be forecasted (such as the next 30 days or 12 months). The new DataFrame is a blueprint for

creating future values from what the model absorbed when it was trained.

Predict Future Transactions:

From the future Data Frame, the model makes predictions with the `.predict()` function.

The return is the values predicted (`yhat`), and lower and upper confidence intervals (`yhat_lower`, `yhat_upper`). The predictions are the predicted future transactions or other statistics from past patterns and seasonal behaviors observed by the model.

Visualize Forecast Results:

Visualization of the predicted outcome is important for interpretation and decision-making. Prophet includes simple plotting functions that display the predicted trend over time, as well as the original data points and uncertainty bands. Also, component plots decompose the forecast into trend, weekly and yearly seasonality, and holidays (if included), providing a better idea of what drives the predictions.

Test Model Performance:

So that the model is as perfect as possible, its forecast is validated through methods such as cross-validation or prediction against the past actual values. To verify the extent to which accuracy the model performs, metrics such as MAE, MSE are employed. Fine-tuning of model—quite likely rescaling seasonality parameters or adding holidays in trying to improve it is reached through this stage.

Deploy & Automate:

The last step is deploying the forecasting solution in a production environment. This could mean establishing a dashboard (using Streamlit, Flask, or Dash) through which users can input data and view refreshed forecasts. Periodic retraining of the model and creation of new forecasts could be automated by setting up scheduled scripts or using cloud services to do so without any human intervention

Hybrid Forecasting Using Prophet+ XGBoost

The hybrid forecasting method combines Prophet's time-series modeling strength with XGBoost's predictive ability to improve the accuracy of sales forecasts. The method takes advantage of Prophet's ability to capture trend and seasonality, while

employing XGBoost to model residual errors influenced by external variables.

Final Forecast Equation:

$$\text{Final Forecast} = \text{Prophet Prediction} + \text{XGBoost} \quad (1)$$

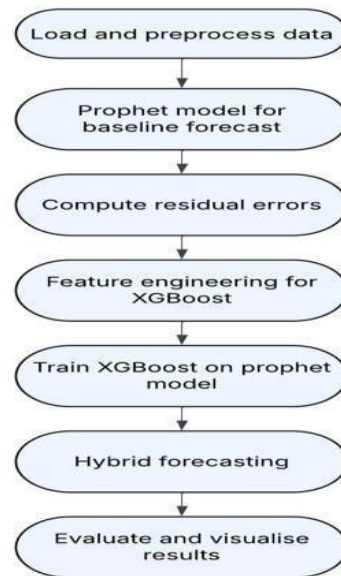


Fig. 3. Hybrid Forecasting flow chart

Load and Preprocess Data:

The workflow begins with loading historical time series data, commonly timestamps and target variable (e.g., transactions or sales). Preprocessing the data includes data cleaning—deletion of duplicate or missing records, standardizing the format, and possibly normalizing values. This step gets the data ready to be modeled well by Prophet and eventually consumed for advanced feature engineering for XGBoost.

Prophet Model for Baseline Forecast:

Next, the cleaned data is used to fit a Prophet model, which serves as the baseline forecasting approach. Prophet picks up major features like trend, seasonality, and holidays. Prophet gives a first estimate that aligns with the interpretation of the underlying patterns made by the model. Prophet, however, may not pick up complex nonlinear relations, and that is where XGBoost takes over to tweak the results.

Compute Residual Errors:

After Prophet generates the forecast, the residual errors (actual values - Prophet-predicted values) are calculated. The residuals point to where the Prophet model does less or does not pick up on subtleties in the data. These residual pieces of the data constitute the target variable for the XGBoost model.

Feature Engineering for XGBoost:

Other features are developed here to augment the XGBoost model. These could be time-based features (e.g., day of the week, month, holiday indicators), lag features, rolling averages, or external data such as weather or economic conditions. The notion is to give XGBoost precise inputs that are able to capture the residuals that are residualized by Prophet.

Train XGBoost on Prophet Model:

It is on the objective of engineered features and residuals that training of the XGBoost model is performed. XGBoost is a robust gradient boosting algorithm that has the capability to model complex, nonlinear relationships. It learns to forecast the residual errors of the Prophet model, effectively addressing its shortcomings and improving overall forecast accuracy.

Hybrid Forecasting:

The Prophet and XGBoost forecasts are combined in the hybrid forecasting stage. Precisely, the final forecast is calculated as the sum of the Prophet predictions and the residuals predicted by XGBoost. This ensemble method utilizes the explainability of Prophet and the predictive capability of XGBoost to produce a stronger and more accurate forecast.

Test and Visualize Results:

Once the hybrid model has been trained, its performance and results must be visualized and analyzed. The performance of the model is gauged by using such metrics as Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Mean Squared Error (MSE): Measures the average squared actual versus predicted differences. Smaller MSE, better model performance.

$$MSE = (1/n) * \sum_{i=1}^n [(Y_i - \hat{Y}_i)^2] \quad (2)$$

Mean Absolute Error (MAE): Average of absolute difference between actual and predicted values. More intuitive than MSE and linearly penalizes errors.

$$MAE = (1/n) * \sum_{i=1}^n [|Y_i - \hat{Y}_i|] \quad (3)$$

Where:

Y_i = Actual value

$\hat{Y}_i$ = Predicted value

n = Number of observations

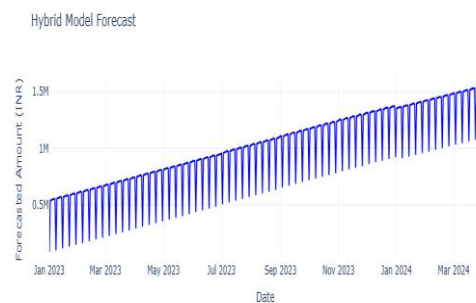
IV. RESULTS

Fig. 4. Forecasting Hybrid Model (Prophet + XGBoost)

Fig.4. The graph with the label "Hybrid Model Forecast" is forecasted monetary values in INR by time based on a hybrid model prediction, most likely based on a mix of Prophet and XGBoost. The x-axis is the timeframe from January 2023 until March 2024, the y-axis the amount forecast in INR but scaled down to approximately 1.6 million. The graph shows a consistent upward slope for the forecast values, which means that there is steady growth in the metric being tracked. Every point on the line is followed by vertical bars, which are the confidence intervals, representing the margin of error for each prediction. The uniform and slender nature of the intervals here shows that the model is very sure about its projections. Overall, this graph majorly shows that the hybrid model has indeed observed the trend and made accurate future prediction.

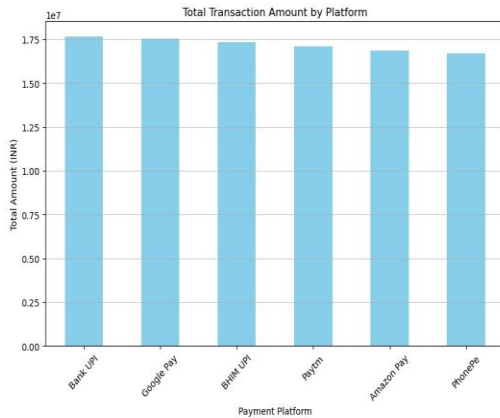


Fig. 5. Analysis Using Hybrid Model.

Fig.5. The bar chart "Total Transaction Amount by Platform" indicates the total transaction amount (in INR) through digital payment platforms. The x-axis represents the platforms—BHIM UPI, Google Pay, Paytm, Amazon Pay, Phone Pay, and Bank UPI—while the y-axis represents the total transaction amount in Rupees.

Of the platforms, BHIM UPI possesses the highest value of transactions, over 400,000 INR, reflecting its mass usage. Google Pay, Paytm, and Amazon Pay follow closely with comparable transaction amounts, all below 400,000 INR. Phone Pay also shows a high presence with a similar amount, while Bank UPI has the lowest transaction amount among the listed platforms.

This chart shows popularity and volume of transactions handled by each platform, showing BHIM UPI to be the most utilized service and Bank UPI in deficit in total transaction count.

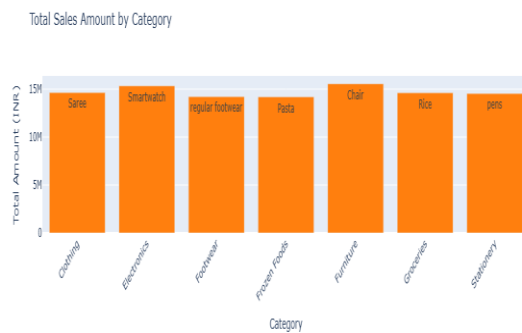


Fig. 6. Total Sales Amount By category

Fig.6. Bar chart "Total Sales Amount by Category" represents total sales in INR across various categories of products. The x-axis has products such as Saree, Smartwatch, pens, Chair, Rice, regular footwear, and Pasta, which are generic categories such as clothing, electronics, Stationery, furniture, grocery, footwear, and frozen foods. The y-axis represents the sales amount in crores (10^7 INR). Saree is topping from the chart when it comes to sales, followed by pens closely, which are signs of best performance in the garment, electronics, and stationery sectors. Other products such as Chair, Rice, Regular footwear, and Pasta also register top contribution with comparatively lower volumes of sales. In general, the graph exhibits a fairly equally distributed sale pattern across categories where garments and electronics are top sale categories.

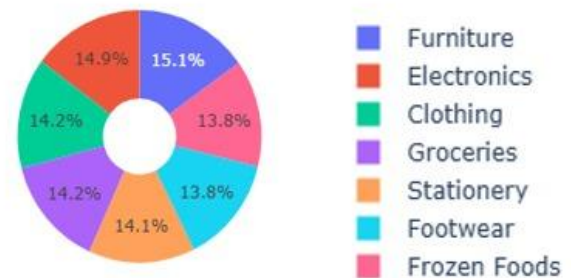


Fig. 7. Sales Distribution by Category

Fig.7. The pie chart indicates the percentage of sales by categories. Each category and its respective percentage is indicated by a segment as part of total sales. Furniture takes up the largest with 15.1%, followed by Electronics with 14.9%, then Clothing and Groceries each with 14.2%.

Stationery contributes 14.1%, with Footwear and Frozen foods each contributing 13.8%. All in all, the distribution is quite even, with each category contributing almost equally to overall sales. This indicates a diversified interest from consumers in all areas, with one category not ruling the market.

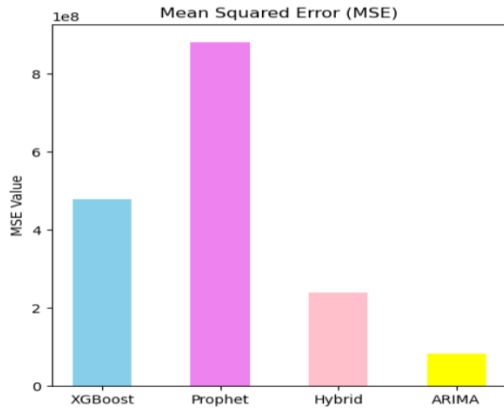


Fig.8. Comparison of MSE

Fig.8. The bar chart titled "mean squared error (MSE)" depicts the prediction accuracy of the four models, via., XGBOOST, prophet, hybrid, and ARIMA. The y-axis represents MSE values, and lower values indicate that the model is working correctly.

Of the models, prophet has the greatest MSE, meaning that it is least accurate in this experiment. XGBOOST is superior but still possesses a relatively high error. Hybrid model possesses a huge improvement with a lesser MSE, which means that model combination can enhance the accuracy of prediction. Arima possesses the smallest MSE among all, meaning that it predicted the most accurately for this dataset.

Generally, the chart highlights how the hybrid and ARIMA models are better in terms of performance compared to the individual models in terms of forecasting accuracy, and the ARIMA is better based on the MSE as a measure.

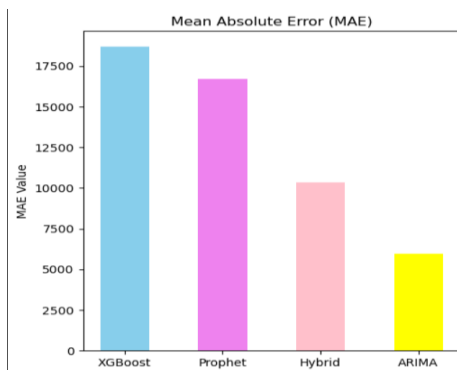


Fig.9. Comparison of MAE

Fig.9. The bar chart labelled "Mean Absolute Error (MAE)" is used to compare the prediction accuracy of the four models, i.e., XGBOOST, Prophet, Hybrid, and ARIMA. The y-axis is for MAE values, and the smaller the value, the better the model.

XGBOOST has the greatest MAE, i.e., it produced the highest average errors of the models. Prophet has less MAE. Hybrid indicates a significant improvement, i.e., improved prediction accuracy compared to individual XGBOOST and Prophet models. ARIMA is the best with the least MAE, i.e., maximum accuracy in terms of average error.

This similarity suggests that although Hybrid model does report improvements based on individual components, ARIMA offers most accurate estimations overall by using the MAE measure.

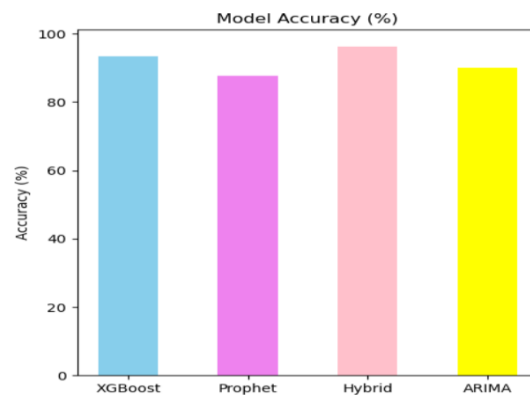


Fig.10. Comparison of Accuracy

Fig.10. Bar chart "Model Accuracy (%)" compares the prediction accuracy of the four models: XGBoost, Prophet, Hybrid, and ARIMA. Y-axis is accuracy percentage, where a higher the value corresponds to better performance.

Among the models, Hybrid performs the best at around 95%, just a hair more, and demonstrates the power of a combination model. XGBoost is the second-best at about 93%, followed by ARIMA at about 90%. Prophet performs worst among the four at about 88%.

The above chart demonstrates that Hybrid performs the best overall in making predictions, outperforming single models and demonstrating its potential at prediction tasks.

TABLE II. COMPARISON BETWEEN MODELS

S. No	Models	Metrics		
		<i>Mse</i>	<i>Mae</i>	<i>Accuracy</i>
1	Prophet	1000005723.43	18451.99	86.19%
2	Arima	38823.39	61725.47	91.27%
3	Xgboost	513424458.50	19133.09	93.29%
4	Hybrid	249685423.98	10643.46	96.27%

V. DISCUSSION

To predict daily sales, you've developed a robust hybrid model that combines Prophet, XGBoost, and Linear Regression. The underlying trend and seasonality of the time series were modeled with Prophet, and intricate, nonlinear trends were captured by training XGBoost on the residuals of Prophet's prediction. To further smooth out long-term fluctuations, a linear regression trend component was also used. All three components are combined in the final forecast ($\hat{y}_{corrected}$), which is displayed as a line plot. The evaluation metrics—mean squared error (MSE), mean absolute error (MAE), and an accuracy score based on MAE—show that the model performs well, and this method successfully improves prediction accuracy.

Total transactions by platform was shown in a static bar graph. This is useful information for offer or promotion of a platform, since it shows which platforms the customer is using the most and where most transactions are occurring.

Each bar had titles that indicated the number one seller in each category but could not be read due to rotation and positioning mistakes. You recreated the chart in Plotly, which had far tidier and more interactive form, to display it in tidier and interactive form. This interactive one is more comprehensible and to work with the data by indicating the best seller within each bar with hover function

Lastly, the total sales distribution according to product categories was depicted through a pie chart. The donut-chart view enabled one to easily view what categories comprise most of the overall revenue. It gave a pleasing-looking and useful view

of performance category-wise with hover tooltips revealing structured sales information.

CONCLUSION AND FUTURE SCOPE

This project constructs a future-proof hybrid forecasting system with strength and optimality machine learning (XGBoost) and statistical modeling (Prophet) capabilities. This solution allows companies to make optimal decisions on the amount of inventory, price, and cost planning. When real-time technologies, deep learning, and automations keep shaping up, such a forecasting model can be refined further to transform into an even more intelligent, scalable, and business-centric platform for the years to come

The hybrid forecasting model can be further enhanced by integrating advanced deep learning techniques such as LSTM, Transformer, and CNN-based models to capture complex, non-linear sales patterns. The use of external data sources, such as economic data, customer sentiment, competitor prices, and weather, can enhance model accuracy and adaptability. AutoML and hyperparameter tuning can achieve optimal performance with minimal human effort, and edge computing offers localized forecasting to improve regional demand forecasting.

Explainable AI (XAI) adoption enhances transparency, offering clear explanations of model choices and enhancing stakeholder trust. Together, these developments position the hybrid model as an entirely automated, real-time, and scalable prediction instrument. Outside retail, this model is applicable to health, finance, and supply chains, as well as precise predictions allow better resources planning and managing risks. Incorporating automated models of decision making, the model allows firms to respond quickly to changing market scenarios, more efficiently, at less cost, and with a competitive edge. With the advancement of AI technology, this hybrid system can potentially revolutionize strategic forecasting in various industries.

REFERENCES

- [1] Taylor, S. J., & Letham, B. (2018). Forecasting at Scale. *The American Statistician*, 72(1), 37-45.
- [2] Dankorpho, P. (2024). Sales Forecasting for Retail Business using XGBoost Algorithm. *Journal of Computer Science and Technology Studies*, Al Kindi Publisher, 6(2), 136-141.
- [3] Malik, S., et al. (2024). Sales Forecasting Using Machine Learning Algorithm in the Retail Sector. *Journal of Computing & Biomedical Informatics*, 6(2), 282–294.
- [4] Li, K. (2023). A Sales Prediction Method Based on XGBoost Algorithm Model. *BCP Business & Management*, 36, 367–371.
- [5] Wei, H., & Zeng, Q. (2021). Research on Sales Forecast Based on XGBoost-LSTM Algorithm Model. *Journal of Physics: Conference Series*, 1754, p. 012191.
- [6] Ji, S., et al. (2019). An Application of a Three-Stage XGBoost- Based Model to Sales Forecasting of a Cross-Border E-Commerce Enterprise. *Mathematical Problems in Engineering*, 2019, 1 –12.
- [7] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.
- [8] Makridakis, S., et al. (2018). Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. *PLOS ONE*, 13(3).
- [9] Zhang, G. P. (2003). Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. *Neurocomputing*, 50, 159– 175
- [10] Bandara, K., et al. (2020). Forecasting Across Time Series Databases Using Long Short-Term Memory Networks on Groups of Similar Series: A Clustering Approach. *Expert Systems with Applications*, 140, p. 112896.