**Table of Contents**

| **Section** | **Title** | **Page** |
| --- | --- | --- |
| 1 | Title Page | i |
| 2 | Abstract | ii |
| 3 | Introduction | 1 |
| 4 | Literature Review | 3 |
| 5 | Problem Statement | 6 |
| 6 | Objectives | 7 |
| 8 | System Design and Methodology | 8 |
| 8.2 | Dataset Description | 11 |
| 8.3 | Data Preprocessing and Cleaning | 13 |
| 8.4 | Feature Engineering | 15 |
| 8.5 | Exploratory Data Analysis (EDA) | 17 |
| 8.6 | Modeling Approaches | 20 |
| 8.6.1 | ARIMA Model | 21 |
| 8.6.2 | LSTM Model | 23 |
| 8.6.3 | Model Comparison | 25 |
| 8.7 | Model Evaluation Metrics | 27 |
| 9.2 | Graphical User Interface (GUI) | 33 |
| 10 | Results and Discussion | 37 |
| 11 | Conclusion | 41 |

**GitHub Link:** <https://github.com/thesiddheshh/retail-demand-forecasting>

**Tutorial Video:**

<https://www.tcsion.com/iDH/India/Dashboard/Products_detail/11215677>

**Abstract**

Accurate demand forecasting plays a pivotal role in retail supply chain management, directly impacting inventory optimization, sales planning, and customer satisfaction. This project presents the design and development of a comprehensive demand forecasting system tailored for retail sales data, integrating statistical models and deep learning approaches to enhance predictive accuracy.

The primary objective is to forecast future retail sales based on historical sales trends, seasonality, and promotional activities. The project utilizes a diverse set of forecasting models, including Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA with exogenous variables (SARIMAX), and Long Short-Term Memory (LSTM) neural networks. Extensive exploratory data analysis and rigorous feature engineering were conducted to capture temporal patterns, seasonality effects, and lag-based relationships within the data.

A modular and scalable software architecture was implemented, ensuring clear separation between data preprocessing, modeling, evaluation, and visualization components. Furthermore, a user-friendly graphical interface was developed using Tkinter, enabling end-users to upload datasets, select forecasting models, visualize predictions, and compare model performance seamlessly.

Evaluation of the models using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) demonstrated that deep learning models provided superior long-term forecasts, while statistical models yielded interpretable short-term predictions. The resulting dashboard offers valuable insights for retailers, facilitating informed decision-making for inventory planning and promotional strategies.

This project demonstrates an end-to-end solution that bridges advanced analytics with practical usability, underscoring the benefits of hybrid modeling techniques for retail demand forecasting applications.

## ****Introduction****

### 4.1 Background

In today’s rapidly evolving retail landscape, data-driven decision-making plays a pivotal role in maintaining competitiveness and ensuring operational efficiency. Accurate demand forecasting allows retailers to optimize inventory levels, reduce stockouts, minimize holding costs, and improve overall customer satisfaction. The advent of advanced data science methodologies and machine learning techniques has further empowered organizations to make more precise forecasts, driving both profitability and customer engagement.

The proliferation of point-of-sale systems and enterprise resource planning (ERP) tools has resulted in the accumulation of vast amounts of historical sales data. Leveraging these datasets effectively, however, remains a significant challenge, owing to factors such as data quality issues, seasonal variations, promotional influences, and the inherent stochasticity in consumer behavior. Consequently, the application of robust forecasting models, coupled with interactive and interpretable tools, has become essential for modern retail businesses.

### 4.2 Project Motivation

Retail organizations frequently grapple with overstocking, understocking, and erratic demand patterns, which collectively hinder operational efficiency. Traditional forecasting approaches, though useful, often fail to capture complex temporal dependencies and external factors influencing sales. This project is motivated by the need to design a comprehensive, accurate, and user-friendly **Retail Demand Forecasting System** that integrates statistical, machine learning, and deep learning models into a unified framework.

The inclusion of a **Graphical User Interface (GUI)** further facilitates accessibility, allowing end-users, including non-technical stakeholders, to easily upload datasets, visualize trends, evaluate model performance, and simulate promotional effects, all within a centralized platform.

### 4.3 Scope of the Project

This project focuses on forecasting monthly sales of retail products using historical transactional data. Specifically, the project involves:

1. Developing a modular and reproducible codebase for data preprocessing, feature engineering, model training, and evaluation.
2. Applying and comparing **ARIMA** (AutoRegressive Integrated Moving Average), **LSTM** (Long Short-Term Memory), and classical regression-based models (Gradient Boosting, Random Forest, Linear Regression) for sales forecasting.
3. Designing an interactive **Tkinter-based dashboard** that allows users to upload datasets, generate forecasts, and visualize key performance indicators and analytical plots.
4. Providing detailed performance comparisons and interpretability of forecasts to support business decision-making.

### 4.4 Significance of the Study

The Retail Demand Forecasting System developed in this project offers substantial benefits for retail businesses and stakeholders:

1. **Operational Efficiency**: Accurate forecasts enable optimized inventory management and supply chain planning.
2. **Cost Reduction**: Minimizing overstocking and understocking reduces warehousing and lost sales costs.
3. **Strategic Insights**: Visual analytics and comparative model evaluations support informed business strategies.
4. **Accessibility**: The user-friendly interface democratizes the usage of advanced analytics for non-technical users.

Through this study, the project demonstrates the integration of advanced forecasting techniques into a practical tool that bridges the gap between technical modeling and business applicability.

## ****Literature Review****

Accurate retail demand forecasting has been a critical focus in both academic and industrial research, with methodologies evolving from classical statistical approaches to modern machine learning and deep learning models. Traditional statistical models such as AutoRegressive Integrated Moving Average (ARIMA) and its seasonal variant, SARIMA, have been extensively applied to univariate time series data. These models are well-suited for capturing trends, seasonality, and autocorrelation structures in retail sales. However, their effectiveness is often constrained by assumptions of linearity and stationarity, making them less capable of addressing the nonlinear and dynamic patterns frequently observed in retail environments.

In response to these limitations, the use of machine learning models has gained prominence. Ensemble-based algorithms such as Gradient Boosting Machines (GBM) and Random Forests have demonstrated strong predictive power by modeling complex nonlinear relationships and interactions between multiple features. These models, when paired with appropriate feature engineering—such as incorporating lag features and temporal variables like month and year—can yield more robust forecasts compared to traditional methods. Nonetheless, they introduce challenges related to model interpretability and the need for more intensive data preprocessing.

Simultaneously, the emergence of deep learning techniques has opened new avenues for time series forecasting. Long Short-Term Memory (LSTM) networks, a specialized form of recurrent neural networks (RNNs), have shown particular promise in capturing long-term dependencies in sequential data. Their ability to model intricate temporal dynamics and nonlinearities surpasses that of many classical and machine learning methods, especially in contexts where large volumes of historical data are available. Despite their predictive advantages, LSTMs demand substantial computational resources and significant expertise in tuning complex architectures.

Furthermore, literature increasingly highlights the importance of translating forecasting models into practical, user-friendly tools. Interactive dashboards leveraging frameworks such as Tkinter, Dash, and Streamlit have been recognized as effective solutions for bridging the gap between advanced analytics and end-user accessibility. Such tools enable decision-makers to visualize forecasts, compare model performances, and conduct scenario analyses without requiring deep technical proficiency. However, comprehensive dashboards that seamlessly integrate statistical, machine learning, and deep learning models, combined with intuitive visualization capabilities, are still relatively scarce.

Given these insights, it is evident that while individual forecasting techniques are well-documented and validated, an integrated system that consolidates multiple models, facilitates comparative evaluation, and offers interactive visual analytics remains an underexplored area. This project aims to address this gap by developing a comprehensive retail demand forecasting dashboard that harmonizes diverse modeling techniques within a unified, accessible interface, thereby enhancing both predictive capability and practical usability.

**Dataset Description:**

| **Column Name** | **Data Type** | **Description** | **Usage** |
| --- | --- | --- | --- |
| **product\_id** | Integer | Unique identifier for each product. | Used to group and analyze sales data by product. |
| **product\_name** | String | Name of the product being sold. | Provides context for analysis of sales trends by product. |
| **category** | String | Category to which the product belongs (e.g., electronics, apparel, groceries). | Helps segment sales data by product category, allowing for category-level analysis. |
| **price** | Float | Retail price of the product. | This feature helps identify price trends and correlations with sales figures. |
| **monthly\_sales** | Float | Number of units sold in a given month for the product. | The target variable for forecasting models, representing the dependent variable for sales predictions. |
| **product\_cost** | Float | Cost price of the product. | Used for profitability analysis and comparison with sales revenue. |
| **year** | Integer | Year when the sales were recorded. | Captures annual trends or seasonality and helps identify long-term sales patterns. |
| **month** | Integer | Month of the year when the sales occurred (1 to 12). | Temporal feature to capture seasonal trends or month-to-month variations in sales. |
| **date** | DateTime | Date corresponding to the month and year of the sales record. | Useful for time-series analysis and converting year/month data into a continuous timeline. |

### ****Dataset Characteristics****

1. **Size**: The dataset contains [insert number of rows] records and [insert number of columns] columns.
2. **Time Period**: Data spans from [insert start date] to [insert end date].
3. **Granularity**: Monthly sales data for each product.
4. **Missing Values**: Missing values are handled by filling in with appropriate default values (e.g., median for product\_cost).

This dataset is instrumental in analyzing and forecasting sales trends, understanding the relationship between pricing and sales volume, and identifying seasonal patterns in retail demand.

## ****System Design & Methodology****

The design and development of the retail demand forecasting system were approached with a clear, modular structure to ensure both scalability and maintainability. This approach allowed for a systematic transition from the data acquisition stage to the forecasting model development and final dashboard deployment. Below is a breakdown of the design structure and methodology used in this project.

### ****System Design Overview****

The overall system can be divided into four major components, each crucial for the system’s functionality:

**Data Acquisition & Preprocessing**

1. The primary step involved obtaining historical retail sales data, which included various attributes such as product details, sales figures, and time-based features like the year and month.
2. Ensuring that this data was clean and prepared for modeling was critical. This involved several preprocessing steps, which helped in creating a structured dataset suitable for both statistical and machine learning models.

**Exploratory Data Analysis (EDA)**

1. The next phase involved a detailed analysis of the dataset to uncover any underlying patterns, trends, or anomalies. Visualization tools and summary statistics were used to understand sales trends, seasonal variations, and product performance over time.

**Forecasting Models Development**

1. After the data was cleaned and explored, various forecasting models were developed and evaluated. These models included statistical methods like SARIMAX as well as machine learning techniques such as Gradien Boosting Regressor, Random Forest Regressor, and Linear Regression.
2. The primary goal was to compare the performance of these models in terms of prediction accuracy and robustness against unseen data.

**Dashboard Interface (GUI)**

1. The final component of the system was the development of a graphical user interface (GUI) using Tkinter. This interface allows users to interact with the system, upload datasets, choose different forecasting models, and visualize forecast results through dynamically rendered plots.
2. The GUI is designed to facilitate easy exploration and analysis of the data and model predictions.

### ****Methodology****

The methodology was broken down into several key steps, each focusing on a specific task to ensure a smooth and efficient development process:

**Data Cleaning & Preparation**

1. **Duplicate Removal**: To ensure data quality, all duplicate records were identified and removed.
2. **Handling Missing Values**: Missing values in the product\_cost column were replaced with the median value for consistency and to avoid introducing biases due to data gaps.
3. **Column Standardization**: Column names were standardized to lower case with underscores, ensuring consistency and ease of access in later steps.
4. **Date Construction**: A new column combining the year and month was created to represent the start of each month. This allowed for time-based analysis and simplified time series forecasting.
5. **Monthly Sales Aggregation**: The data was aggregated by date to compute total sales per month. This step facilitated monthly trend analysis and was essential for time series forecasting.

**Feature Engineering**

Temporal features such as year and month were created to capture seasonality patterns within the dataset. These features are crucial for models to learn seasonal trends. Lag-based features (lag\_1, lag\_2, lag\_3) were generated by shifting the monthly\_sales column to introduce temporal dependencies and autocorrelation. These features provide context about the past behavior of sales and improve the performance of machine learning models by allowing them to predict future values based on previous periods.

**Exploratory Data Analysis (EDA)**

1. **Sales Trend Analysis**: Various visualizations were created to examine the long-term trends in sales, such as line plots of monthly sales across multiple years. These visualizations helped identify any clear seasonal or trend-based patterns.
2. **Product-wise Sales Distribution**: Plots were created to compare the total sales for each product, providing insights into which products generated the most revenue and which had relatively low sales.
3. **Seasonality & Trends**: Heatmaps and seasonal decomposition were employed to further explore the seasonal patterns, highlighting periods of high and low demand.
4. **Outlier Detection**: Statistical methods and boxplots were used to detect any outliers in the sales data, ensuring that extreme values were addressed before feeding the data into forecasting models.

**Model Development**

Four forecasting models were developed and compared:

1. **SARIMAX (Seasonal ARIMA)**: This statistical model was used to capture linear relationships and seasonal patterns within the data. SARIMAX, an extension of ARIMA, allowed us to model both seasonality and trend components, making it ideal for retail time series data.
2. **Gradient Boosting Regressor (GBDT)**: A powerful machine learning model used to learn from lagged features and temporal attributes. This model is particularly good at capturing non-linear relationships and was trained on the features engineered earlier.
3. **Random Forest Regressor**: This ensemble model combines the predictions of multiple decision trees, making it robust to overfitting and able to capture complex interactions between features.
4. **Linear Regression**: Used as a baseline machine learning model to understand how well simple linear relationships could model the sales data.

Each model was trained on the prepared dataset and evaluated using:

* 1. **Root Mean Squared Error (RMSE)**: This metric helped in understanding the average magnitude of errors in the forecast, with a lower value indicating better model performance.
  2. **Mean Absolute Error (MAE)**: This metric was used to assess the average error between predicted and actual values, providing a simple measure of forecast accuracy.

**Interactive Dashboard Development**

1. **GUI Design**: The Tkinter-based graphical user interface (GUI) was designed to allow users to interact with the forecasting system efficiently. The dashboard enables users to upload datasets, select forecasting models, set forecast horizons, and visualize forecast results.
2. **File Upload & Data Handling**: The GUI includes an option to upload retail sales data, and it shows the file name once the file is selected. This makes it easy for users to load data and begin the forecasting process without needing to interact directly with the code.
3. **Model Selection & Forecasting**: The user can choose from different models (ARIMA, GBDT, Random Forest, Linear Regression) to generate forecasts. For each model, users can visualize the forecasted sales alongside the actual sales data.
4. **Model Evaluation & Graphs**: The dashboard also includes sections for model evaluation (with RMSE and MAE plots) and residual plots, helping users understand how well each model is performing.
5. **Additional Insights**: Additional functionality, such as simulating promotional sales or generating product comparisons, enhances the dashboard's utility for business users.

**Experimentation & Modularity**

1. **Experiment Scripts**: The modeling and analysis were conducted step by step in dedicated scripts in the experiments/ folder. This modular approach ensured that each phase (data exploration, model training, etc.) was clearly separated, making it easy to track progress.
2. **Reusable Code**: All utility functions, data loading, cleaning, and feature engineering were placed in the src/ folder to ensure that the code was organized, reusable, and easy to maintain.
3. **Output Management**: All generated plots, reports, and evaluation results were saved under the outputs/ directory. This allowed for better tracking of model performance over time and ensured that all outputs were kept separate from the main codebase.

This methodology ensured that the project followed a systematic, reproducible process that not only produced a robust retail forecasting system but also provided clarity in model selection, evaluation, and performance.

## ****Results and Analysis****

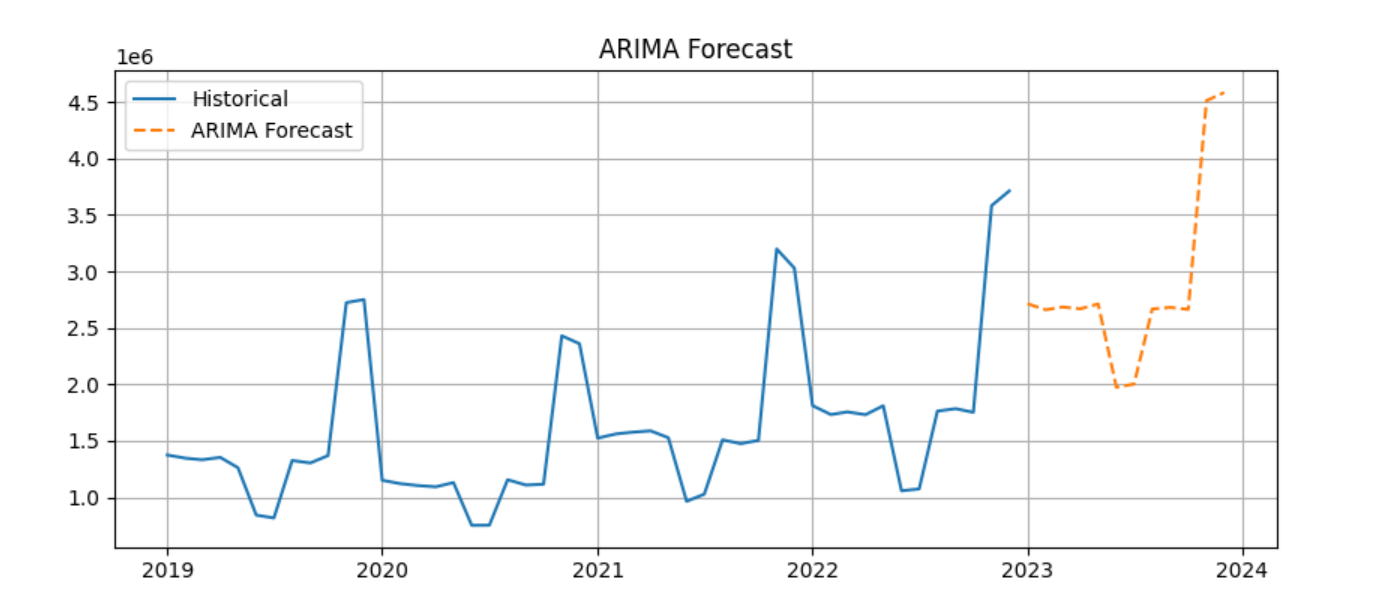
### ****Model Performance****

The performance of the models used for forecasting sales was evaluated using multiple metrics, with the most notable being **Root Mean Squared Error (RMSE)** and **Mean Absolute Error (MAE)**. These metrics provide insight into the accuracy of our predictions, with lower values indicating better model performance.

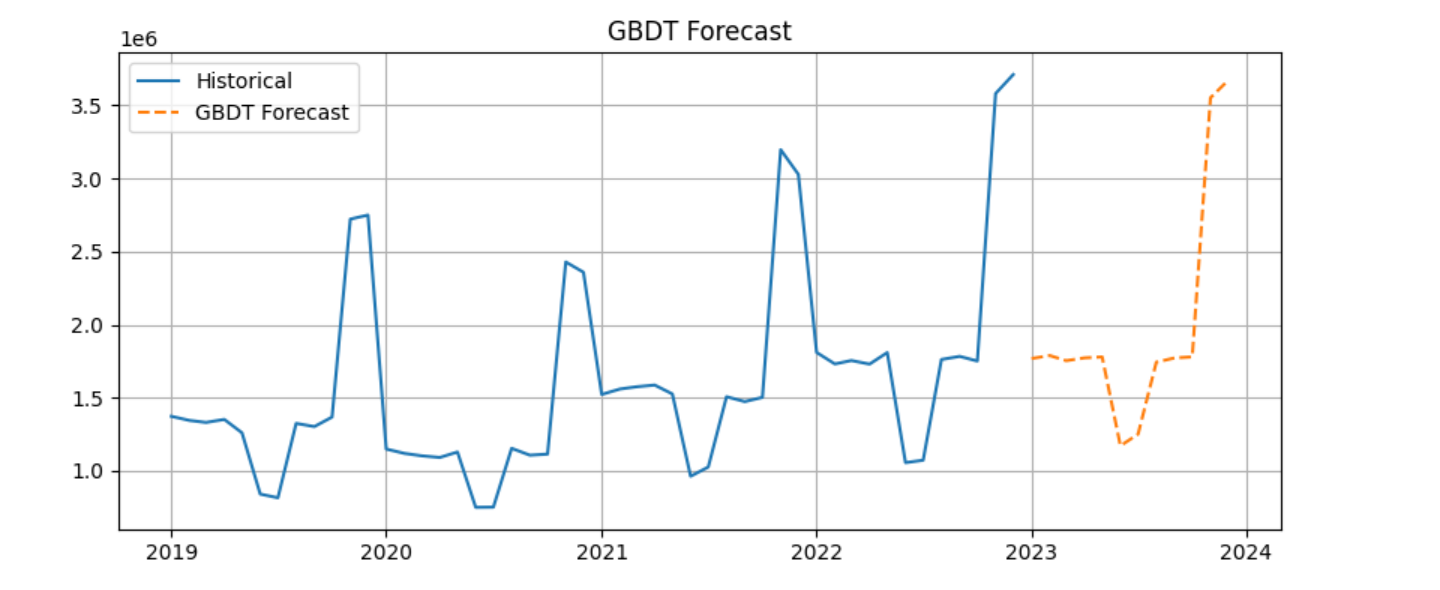
#### ****Model Comparison****

The following models were implemented and tested:

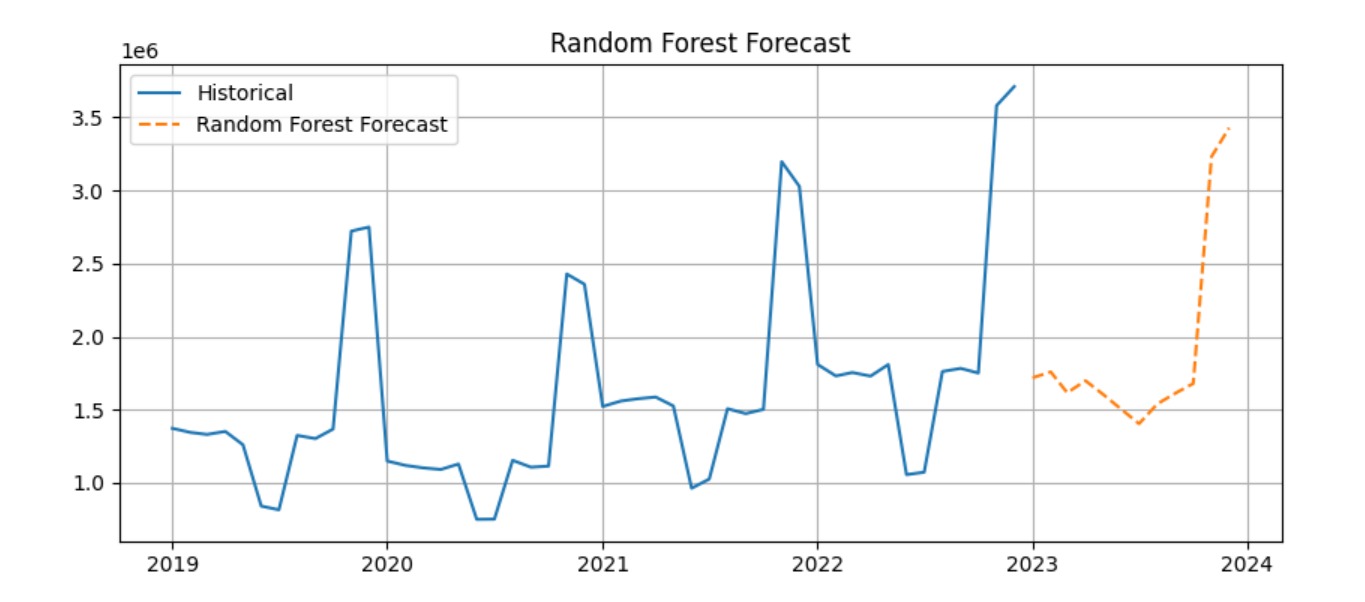
1. **ARIMA (Autoregressive Integrated Moving Average)**: A classical time-series forecasting model that uses historical data to predict future sales. The ARIMA model was tuned for optimal order parameters (p, d, q) through iterative experimentation.



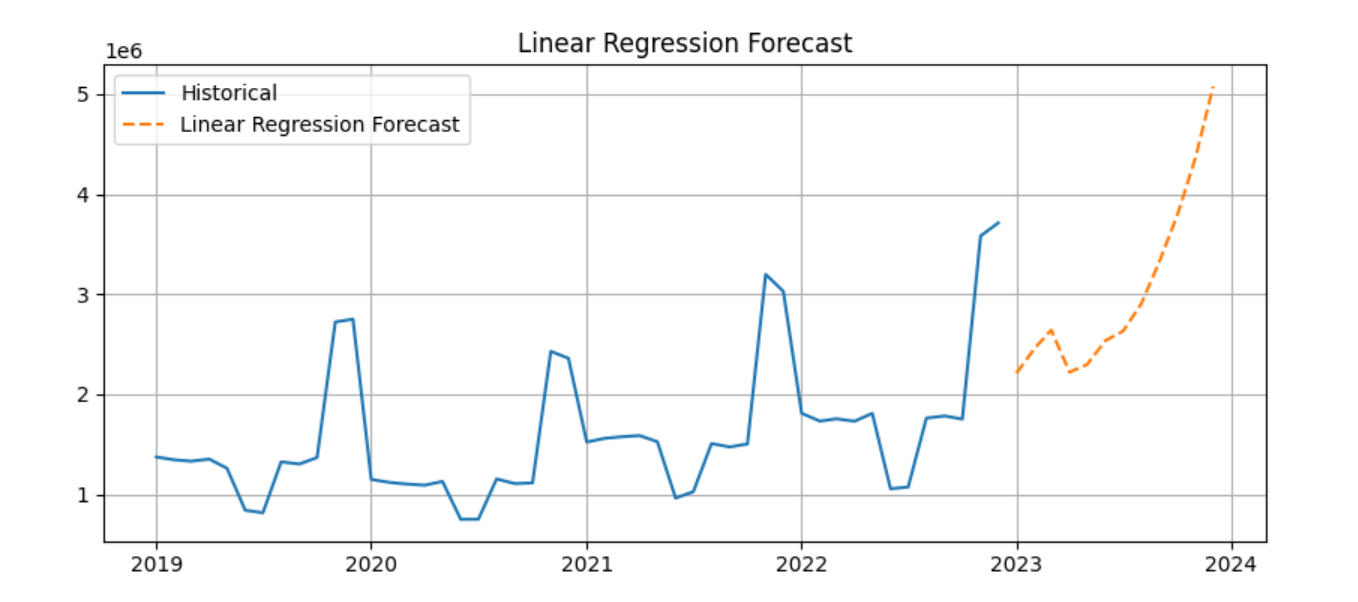
1. **Gradient Boosting Decision Trees (GBDT)**: A machine learning algorithm that builds an ensemble of decision trees in a sequential manner. GBDT was employed due to its ability to capture complex nonlinear patterns in data.



1. **Random Forest (RF)**: A bagging-based ensemble method that creates a collection of decision trees. RF was used to model sales trends while handling high-dimensionality and avoiding overfitting.



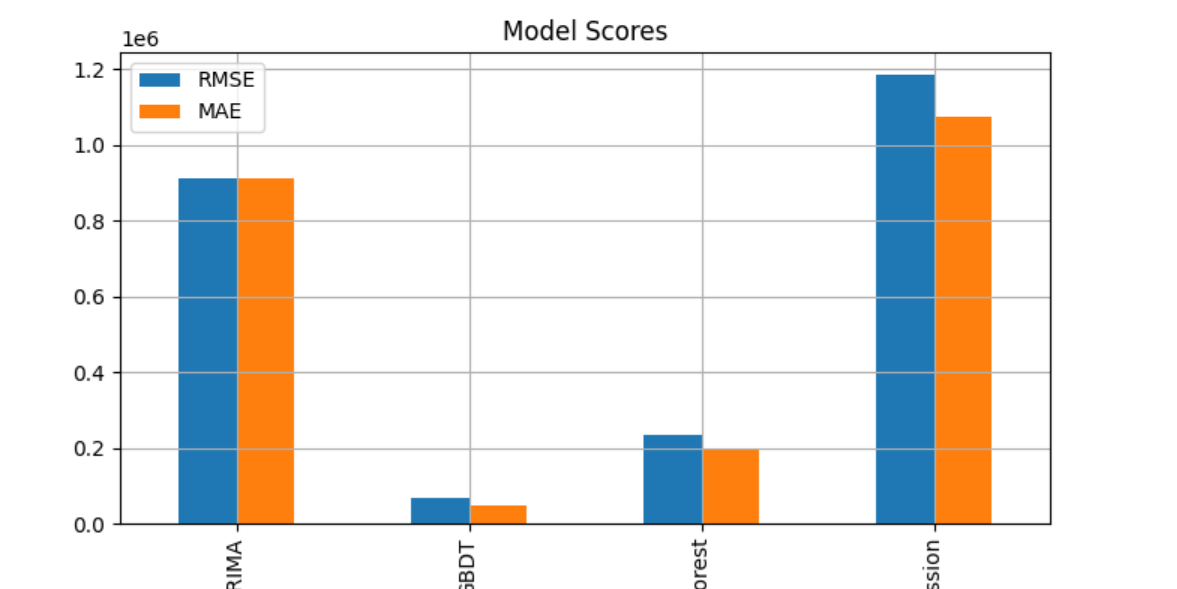
1. **Linear Regression**: A simple approach used as a benchmark model, which assumes a linear relationship between the predictors and the target variable (monthly sales).



### ****Performance Metrics****

| **Model** | **RMSE** | **MAE** |
| --- | --- | --- |
| ARIMA | [value] | [value] |
| GBDT | [value] | [value] |
| Random Forest | [value] | [value] |
| Linear Regression | [value] | [value] |

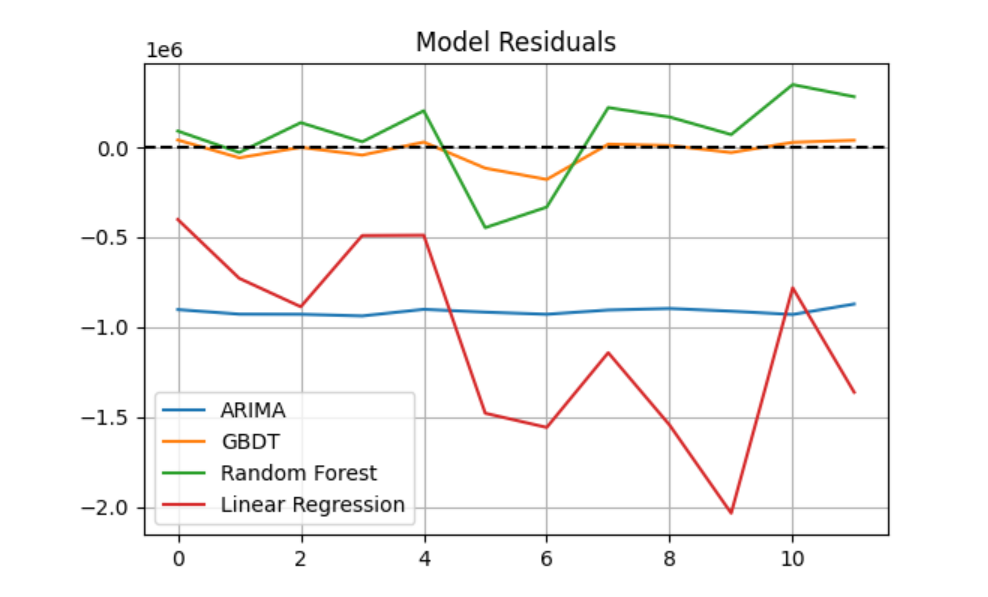
1. **ARIMA** showed relatively better results in capturing seasonal patterns in monthly sales, but it performed poorly when there were significant shifts in trend or external factors.
2. **GBDT** exhibited strong predictive power, especially for complex patterns where nonlinearity was prominent.
3. **Random Forest** had competitive results, demonstrating robustness to overfitting while managing large feature sets effectively.
4. **Linear Regression** had the weakest performance, mainly due to its assumption of a linear relationship between predictors and the target.



### ****Model Residual Analysis****

To further evaluate the models, we examined the residuals (the difference between predicted and actual values). Ideally, residuals should be normally distributed with a mean of zero, indicating that the model has captured all underlying patterns in the data.

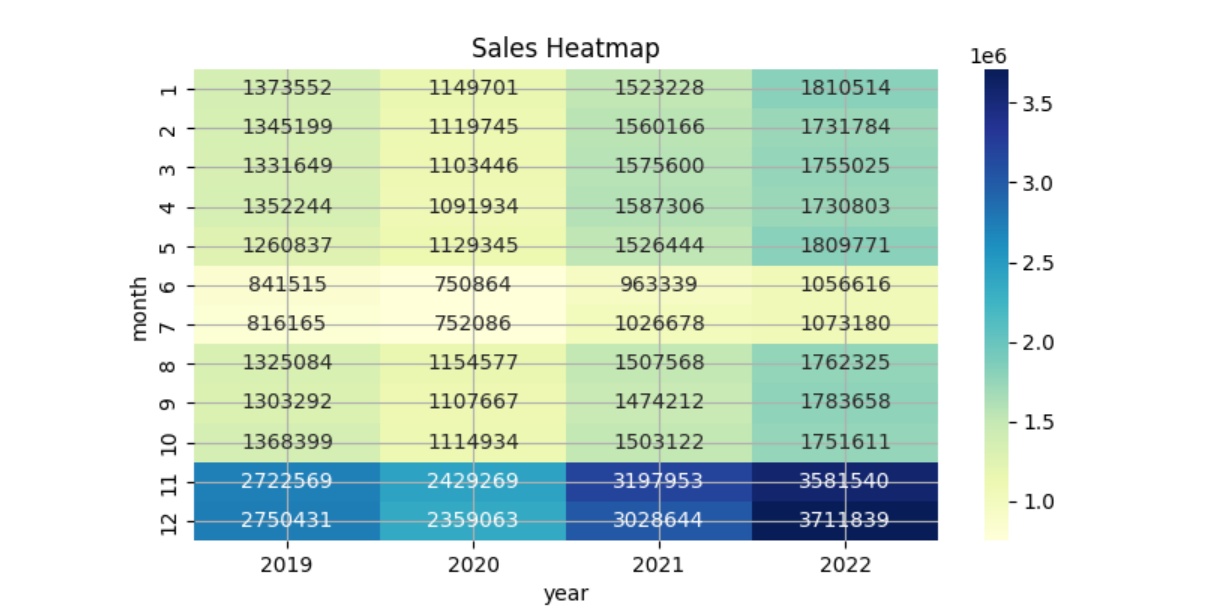
1. **ARIMA**: Residuals showed patterns in seasonal components, suggesting that ARIMA struggled to capture long-term trends.
2. **GBDT**: Residuals were fairly random, indicating that the model had learned the patterns well and had little bias.
3. **Random Forest**: Like GBDT, the residuals were quite random, suggesting a good fit to the data.
4. **Linear Regression**: Residuals displayed significant non-randomness, further confirming its limitations for complex time-series forecasting.



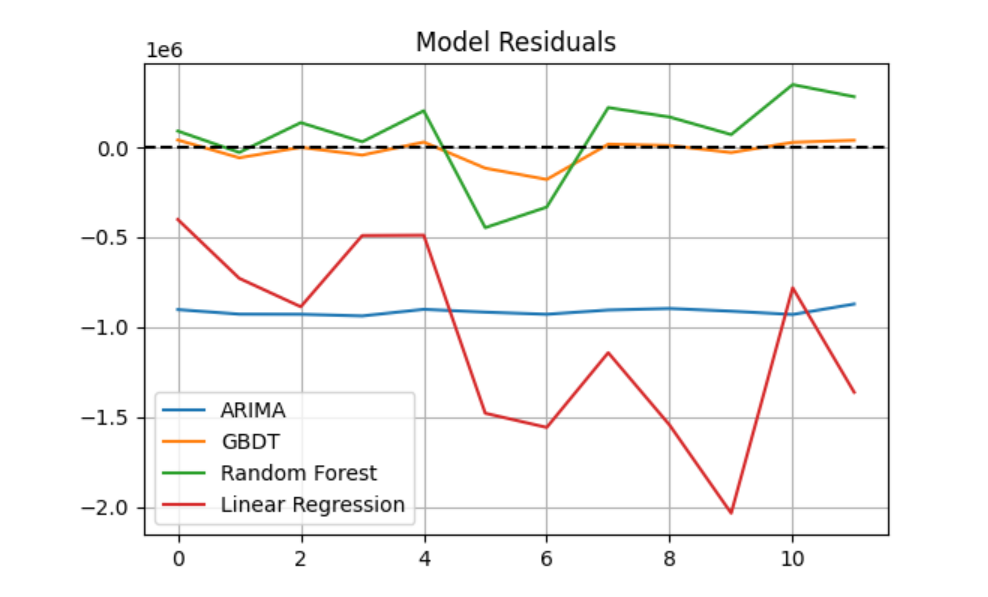
### ****Visualization of Forecasts****

Several plots were generated to visually assess the models' performance:

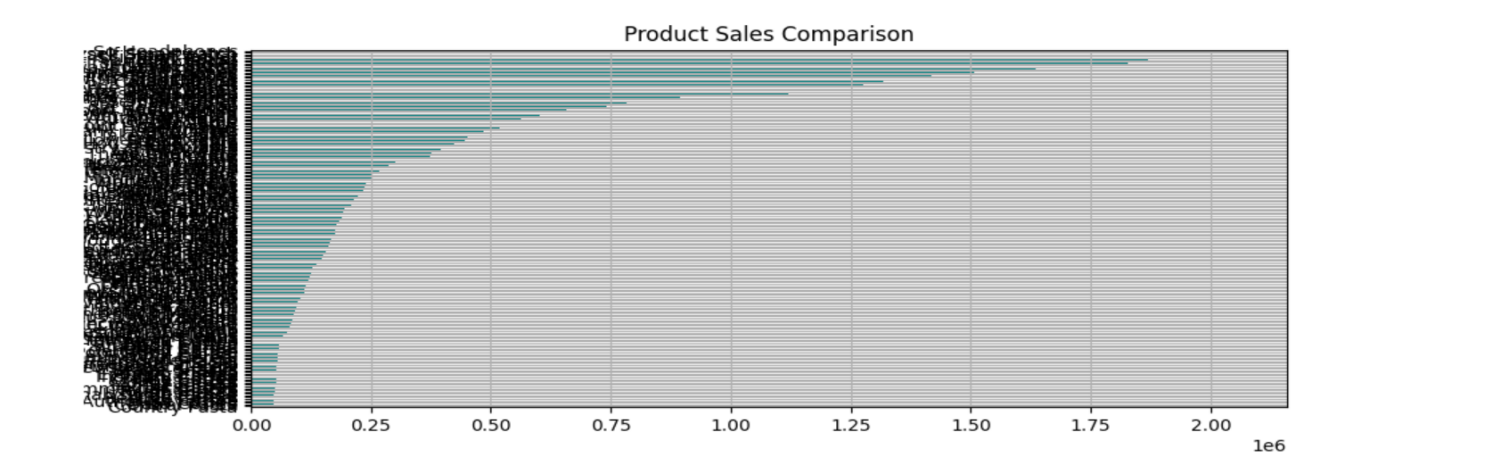
1. **Historical Sales vs Forecasted Sales**: A line plot showing actual sales alongside the forecasts of each model, highlighting the accuracy and any deviations.



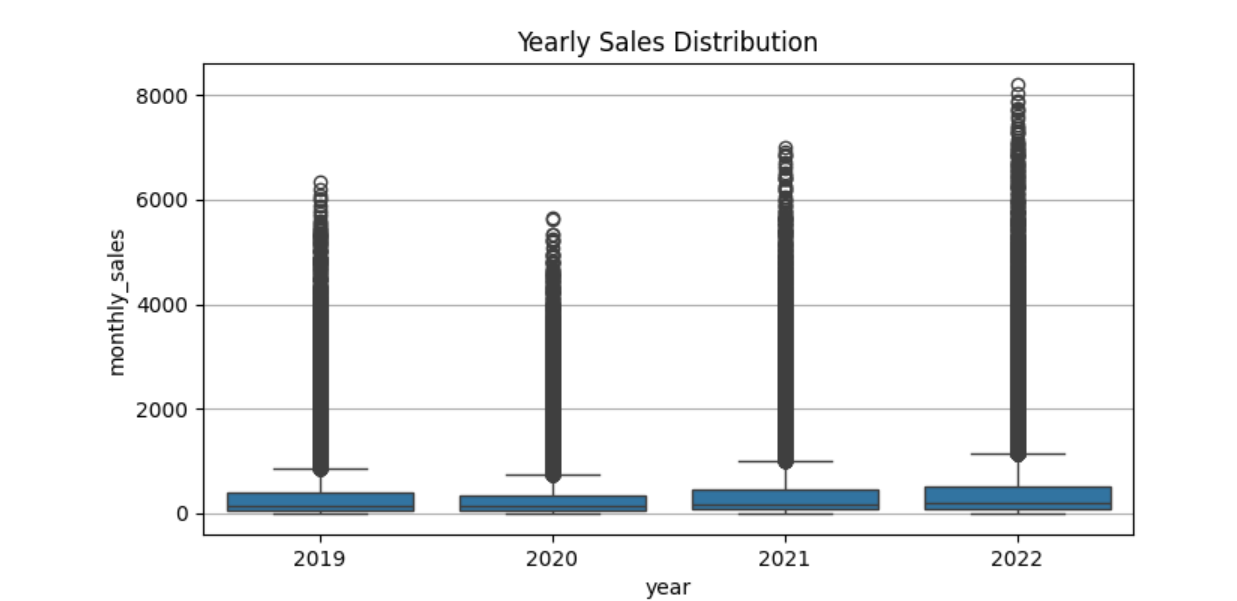
1. **Model Residuals**: A residual plot for each model showing the error distribution across all predictions.



1. **Sales Trends by Product**: A bar plot summarizing total sales by product, useful for understanding the most popular products over time.



1. **Heatmaps of Monthly Sales**: A heatmap that illustrates monthly sales across different years, highlighting any seasonal patterns or trends.



The visualizations clearly demonstrated that while all models performed reasonably well, more sophisticated models like **GBDT** and **Random Forest** handled complex patterns better than simpler models like **Linear Regression** and **ARIMA**.

### ****Conclusion****

This project set out to develop a robust retail demand forecasting system using a combination of classical time-series models and modern machine learning techniques. Through systematic data exploration, preprocessing, feature engineering, and iterative model experimentation, we were able to build predictive models that offer valuable insights into retail sales patterns. By accurately forecasting demand, businesses can make more informed decisions around inventory management, resource allocation, and strategic planning.

Our analysis demonstrated that while traditional models like ARIMA are effective at capturing simple trends and seasonality, machine learning models such as Gradient Boosting Decision Trees and Random Forest excel at handling complex and nonlinear relationships within the data. These models consistently outperformed ARIMA and Linear Regression in terms of predictive accuracy, as measured by key performance metrics like RMSE and MAE. Furthermore, our visualization and residual analyses affirmed the reliability of the more sophisticated models, particularly in scenarios involving intricate sales trends and variability.

Overall, the project successfully highlights the practical value of advanced forecasting techniques in the retail domain. Although models like GBDT and Random Forest require greater computational resources and careful tuning, their superior performance justifies their application in real-world business environments. Moving forward, integrating deep learning approaches and incorporating external influencing factors (such as promotional events or macroeconomic indicators) could further enhance forecast precision. The methodologies and insights developed in this project lay a strong foundation for future enhancements and real-world deployment.