A Software Group Project Report

On

Sales forecasting using machine learning

Submitted by

Naresh Bariya (IU2141050005)
Jagrut Bhatt (IU2141050011)
Dev Darji (IU2141050029)

Guided by

Mr.Indrjeet Rajput

In partial fulfillment for the award of the degree Of

BACHELOR OF TECHNOLOGY

In

Computer Engineering

INSTITUTE OF TECHNOLOGY AND ENGINEERING INDUS UNIVERSITY
CAMPUS, RANCHARDA, VIA-THALTEJ
AHMEDABAD-382115, GUJARAT, INDIA,
WEB: www.indusuni.ac.in

Nov-Dec 2024

Table of Content

ABSTRACT

Sales prediction is a critical component in the decision-making processes of organizations across various industries. However, accurately forecasting sales remains a complex challenge due to inherent trends and seasonality in the data. This research focuses on leveraging machine learning techniques, specifically Random Forest, XGBoost, Linear Regression, LSTM, and Gradient Boosting, to effectively address trend and seasonality detection in sales forecasting. Key features were selected based on correlation coefficients and used to train these models. Among the models, Gradient Boosting demonstrated the best performance in terms of R^2 score and accuracy, followed by XGBoost, Random Forest, Linear Regression, and LSTM. The study provides comprehensive insights into the capabilities of machine learning in sales prediction, offering practical guidance for handling large datasets while addressing seasonality and trends. Moreover, the findings outline common challenges encountered during the modeling process and suggest ways to overcome them, ultimately enabling retailers to improve the reliability and precision of their sales forecasts, enhancing their decision-making in sales management.

KEYWORDS:

Machine learning; Random Forest;LSTM;Linear Regression;XGBoost;Gradient Boosting.

CHAPTER 1: INTRODUCTION

1.1 Brief Overview of the Project Topic

Introduction to the Topic:

Sales forecasting is a crucial process for businesses across various industries, enabling companies to predict future sales based on historical data. Accurate forecasting assists in strategic decision-making related to inventory management, resource allocation, marketing campaigns, and financial planning. However, traditional forecasting methods face limitations when dealing with complex patterns such as seasonality and trends in the data. This project focuses on leveraging machine learning techniques, including Random Forest, Gradient Boosting, LSTM, XGBoost, and Linear Regression, to improve the accuracy of sales predictions. These models are compared with traditional time-series models like SARIMA to demonstrate the effectiveness of machine learning in modern sales forecasting.

Relevance / Need of the Project

The need for accurate sales forecasting is essential in today's fast-paced business environment, where timely decisions directly impact profitability and competitiveness. Machine learning provides the ability to handle large datasets and capture non-linear relationships, making it a powerful tool for forecasting in industries with complex sales patterns. By addressing seasonality and trend detection using machine learning models, this project aims to improve sales forecasting accuracy, which is critical for optimizing business operations and resource management

1.2 Purpose / Problem Statement

Problem Identification

Traditional forecasting techniques, while effective in some cases, struggle to handle non-linear patterns, complex trends, and seasonal fluctuations in sales data. Methods such as ARIMA and SARIMA are often limited in their ability to capture the intricacies of modern business environments where sales are influenced by multiple factors. The challenge lies in finding robust methods that can address these patterns while also handling large datasets efficiently. Without precise sales forecasting, businesses risk mismanaging inventory, missing market opportunities, and failing to optimize operations.

Objectives

- To identify machine learning models that effectively handle seasonality and trends in sales data.
- To compare the performance of machine learning models such as Random Forest, Gradient Boosting, LSTM, XGBoost, and Linear Regression with traditional time-series models like SARIMA.
- To enhance the accuracy of sales forecasts by integrating advanced machine learning techniques.
- To provide insights that can improve decision-making related to sales management and business strategy.

1.3 Technology Overview

Overview of Relevant Technologies

This project utilizes several advanced machine learning algorithms and time-series models. Key technologies and models used include:

- Random Forest: A decision tree-based ensemble learning technique effective for regression tasks.
- Gradient Boosting: A boosting technique that sequentially builds models to correct errors from previous iterations, improving prediction accuracy.
- XGBoost: An optimized implementation of gradient boosting designed for high-performance tasks.
- LSTM (Long Short-Term Memory): A type of recurrent neural network (RNN) that is well-suited for time-series forecasting due to its ability to capture long-term dependencies.
- Linear Regression: A simple and interpretable statistical method used for predictive modeling.
- SARIMA (Seasonal Autoregressive Integrated Moving Average): A traditional time-series model that incorporates seasonality, trends, and lagged data.

Tools and Platforms Used

Several tools and libraries are used to implement the models and analyze data:

- Pandas: A data manipulation and analysis library essential for handling large datasets.
- Scikit-learn: A machine learning library that provides tools for model building, including Random Forest, Gradient Boosting, and Linear Regression.
- SciPy: A Python library used for scientific computing and technical data analysis.
- Matplotlib: A visualization library for creating detailed plots and charts to explore data trends.
- TensorFlow/Keras: Used for implementing and training the LSTM neural network model.

Justification for Technology Selection

The combination of traditional and advanced machine learning models ensures a comprehensive approach to sales forecasting. Machine learning algorithms like Gradient Boosting and XGBoost were chosen for their ability to handle large datasets and non-linear patterns. LSTM is selected for its strong performance in time-series forecasting, making it ideal for capturing seasonality and trends. SARIMA is included as a benchmark to compare machine learning methods with traditional time-series approaches. Tools like Pandas and Scikit-learn are widely used in the data science community for their ease of use, robustness, and efficiency in managing complex datasets.

CHAPTER 2: LITERATURE REVIEW

2.1 Summary of Existing Research

Sales forecasting plays a vital role in the decision-making process for organizations across various industries. The literature consistently emphasizes the need for accurate forecasting, particularly when dealing with seasonality and trends in sales data. Traditional models such as SARIMA, ARIMA, and ETS are widely used due to their simplicity and ability to handle linear patterns. However, as highlighted by Khan (2020), these models struggle to capture complex and non-linear seasonal and trend patterns, which are increasingly common in today's dynamic markets

Machine learning (ML) techniques have gained significant traction in recent years due to their capacity to model complex relationships and process large datasets efficiently. According to Leung et al. (2020), machine learning models, especially ensemble methods like Random Forest and Gradient Boosting, have demonstrated strong predictive performance when dealing with sales data Studies by Zhao et al. (2020) and Javatpoint (2024) also support the efficacy of ML models, especially recurrent neural networks (RNN) and LSTM, for their ability to capture long-term dependencies and non-linear patterns in time-series forecasting

2.2 Key Findings and Gaps

Key Findings:

- Traditional time-series models like SARIMA are commonly used but struggle with complex, non-linear patterns and large datasets.
- Machine learning models, such as Random Forest, Gradient Boosting, and LSTM, offer superior performance in terms of capturing seasonality and trend dynamics in sales data.
- Hybrid models that combine machine learning with traditional time-series methods, like those explored by Ensafi (2022), show promise for improving forecast accuracy

Gaps:

- Limited research focuses on direct comparisons between machine learning models and traditional time-series models in real-world retail datasets.
- Many studies fail to explore the challenges related to model interpretability, data preprocessing, and handling of large datasets.

2.3 How the Project Addresses These Gaps

This project seeks to bridge the gap in existing research by conducting a thorough, side-by-side evaluation of both traditional time-series models like SARIMA and advanced machine learning (ML) algorithms such as Random Forest, Gradient Boosting, XGBoost, and Long Short-Term Memory (LSTM) networks. The comparison focuses on their respective abilities to detect and model seasonality and trends in sales data, a critical aspect often overlooked in previous studies. By incorporating a diverse range of algorithms, the project offers a more comprehensive analysis of which methods are most effective under various conditions, such as different levels of seasonality, trend complexity, and dataset size.

One of the key contributions of this project is its emphasis on feature selection. Using correlation analysis, the project identifies the most relevant features that influence sales, ensuring that only the most impactful variables are included in model training. This helps improve the accuracy and efficiency of the models by eliminating irrelevant or redundant features, which can lead to overfitting and degraded performance. Feature engineering and selection play a crucial role in enhancing the ability of machine learning models to detect seasonality and trends, a task that traditional models like SARIMA struggle with, especially when dealing with non-linear patterns.

Moreover, the project addresses several practical challenges associated with implementing machine learning models for sales forecasting. These include data preprocessing techniques such as handling missing values, normalizing data, and transforming categorical variables into formats that can be effectively used by machine learning algorithms. The research also delves into issues of model interpretability, especially important when explaining the results to stakeholders who may not have a technical background. Ensuring that the models provide understandable insights is essential for their application in real-world business decision-making.

In addition, this project tackles the scalability of machine learning models when applied to large datasets, a common challenge in retail forecasting. The ability of machine learning algorithms, such as Gradient Boosting and LSTM, to process and learn from large volumes of data makes them particularly valuable for organizations handling extensive historical sales data. By comparing their performance to traditional models like SARIMA, which can become computationally expensive and less accurate with larger datasets, this research offers a practical perspective on which models are more suitable for scaling up in a business context.

CHAPTER 3 SYSTEM DESIGN & MODULE DESCRIPTION

3.1 System Architecture

The system architecture consists of several components that work together to perform sales forecasting.

The key modules include:

- **Data Collection and Input:** The system starts by loading the sales data using pandas. It explores the dataset, analyzing its structure, shape, and descriptive statistics.
- **Data Preprocessing:** This module involves cleaning the data by handling missing values, removing unwanted columns, treating structural errors, handling outliers, and formatting date-related columns.
- **Feature Engineering:** New features such as the sales month, and year are created from the "Order Date" column, which enhances model performance by incorporating time-based patterns.
- **Data Visualization:** This module generates visualizations like box plots to detect outliers and graphs to analyze sales trends over different months and years.
- **Model Training and Evaluation:** Although not fully visible yet, this section likely involves the training of machine learning models like Random Forest, Gradient Boosting, XGBoost, etc., based on earlier indications.

Interactions Between Components

- **Data Sources → Data Ingestion Layer**: Raw data flows into the system from various sources, such as databases or files.
- **Data Ingestion Layer → Data Processing Layer**: The ingested data is cleaned, filtered, and prepared for analysis.
- **Data Processing Layer → Storage Layer**: After processing, the clean data is stored in a centralized location for future use.
- **Storage Layer → Analytical Engine**: The data is retrieved from storage and analyzed using machine learning models or forecasting techniques.
- **Analytical Engine → Visualization Layer**: Results from the analysis are passed to the visualization component for presentation.
- **Visualization Layer → User Interface (UI)/API**: Users access the insights and reports through dashboards or APIs.

- **Feedback Loop**: Data from the performance of the models and system usage can be fed back into the **Analytical Engine** for continuous improvement.

Data flow diagrams

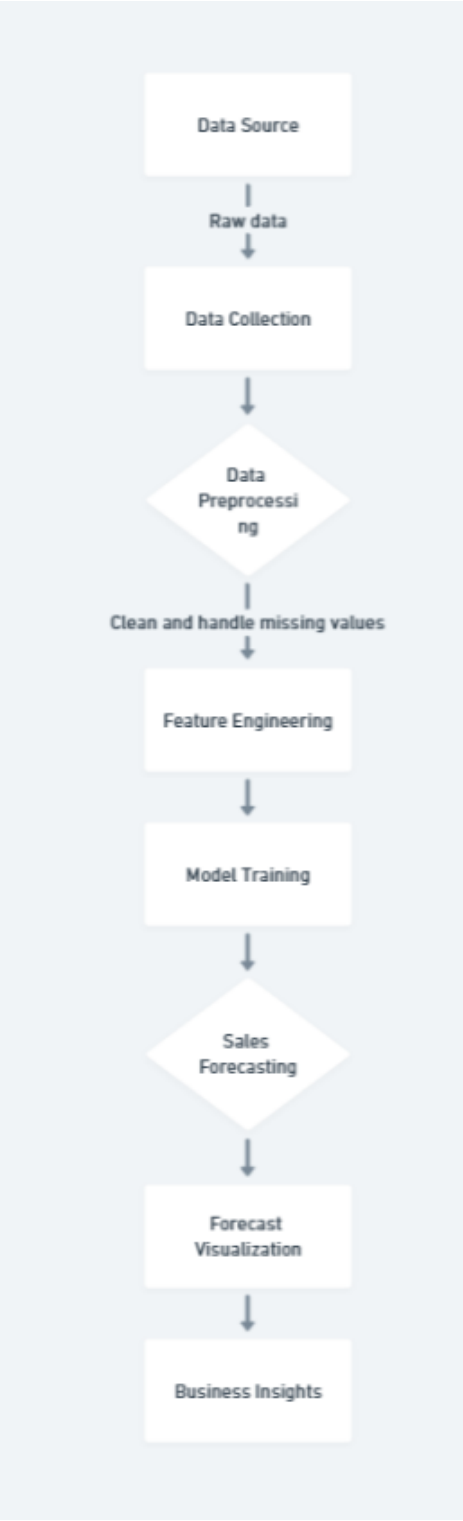

Figure 1. DFD Diagram

control flow diagrams

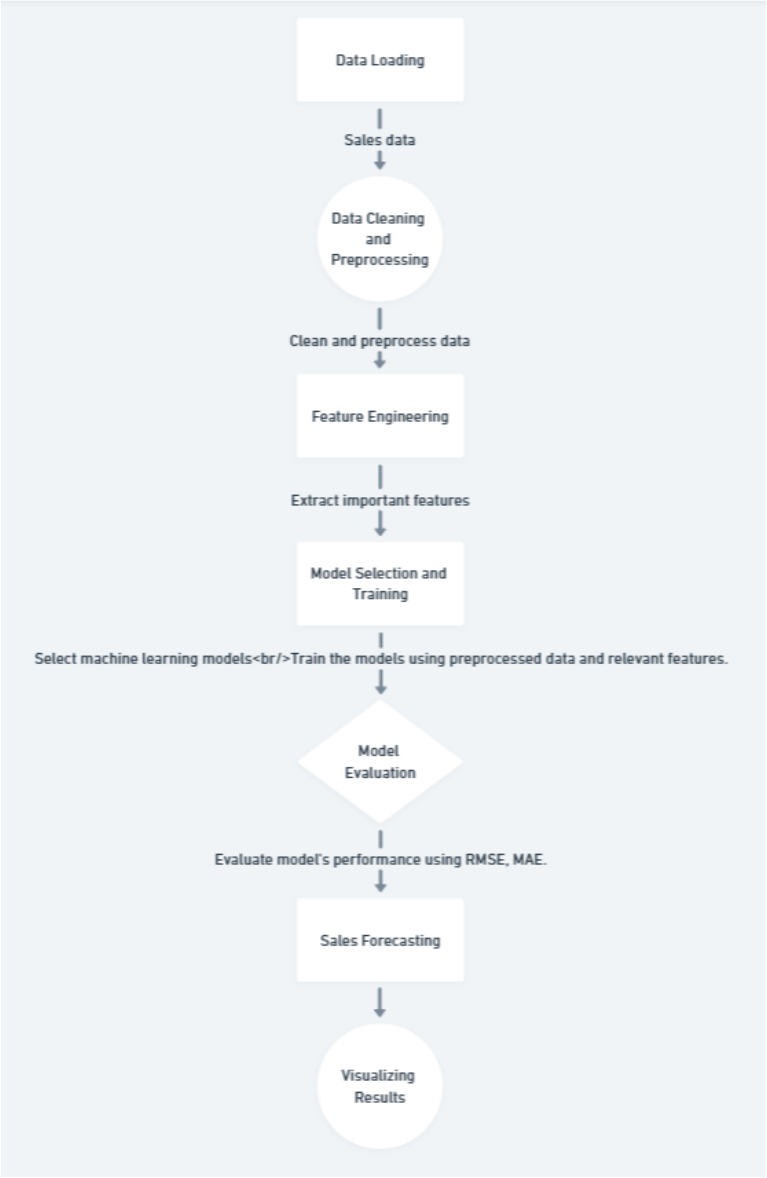

Figure 2. CFD Diagram

3.2 Module Description

Module 1: Data Collection & Input Module

- Overview: This module is responsible for loading the raw sales dataset from a CSV file using pandas. It prepares the data for subsequent processing, making it the first step in the system architecture.
- Detailed Functionality:
    - Loads sales data from a CSV file using pandas (pd.read_csv()).
    - Inspects the dataset using methods such as data.head(), data.shape(), data.info(), and data.describe() to understand the structure, identify the number of missing values, and gain an initial overview of the data.
    - Ensures that the dataset is correctly loaded for further steps in the pipeline.

Module 2: Data Preprocessing Module

- Overview: This module handles the cleaning and preparation of the raw data for machine learning. The goal is to ensure the data is free from inconsistencies, missing values, and other quality issues, making it suitable for analysis and model training.
- Detailed Functionality:
    - Handling Missing Values: Missing values in columns (e.g., "Postal Code") are replaced using imputation techniques like filling with a default value (fillna()).
    - Removing Unwanted Columns: Columns that are not relevant to sales prediction, such as "Row ID", "Order ID", "Customer ID", and "Product ID", are removed using the drop() method.
    - Detecting and Resolving Duplicates: The dataset is checked for duplicates using duplicated(), and any duplicates are removed.
    - Handling Structural Errors: Ensures that categorical columns like "Order Date", "City", and "Ship Mode" have correct values. For instance, the "Order Date" is converted to datetime format.
    - Outlier Detection: Visualizes the data using box plots (df.plot(kind='box')) to identify and treat outliers that could skew model performance.

Module 3: Feature Engineering Module

- Overview: This module creates additional features from the existing columns to improve the machine learning model's performance by providing more meaningful data representations. These features help capture trends, seasonality, and other time-related aspects that impact sales forecasting.
- Detailed Functionality:
    - Extracting Date Features: Extracts the Month and Year from the "Order Date" column by converting it to datetime format using pd.to_datetime(). Then, new columns for Month and Year are added to the dataset.
    - Categorizing Month: Converts the Month number into a more human-readable form (e.g., converting 1 into "January") by mapping numerical values to month names.
    - Aggregating Sales by Date: Groups the sales data by Order Date to analyze monthly sales trends.
    - Preparing Data for Models: Ensures that the processed dataset is ready for model training by aligning all the feature columns (e.g., Month, Year, Sales) correctly.

Module 4: Exploratory Data Analysis (EDA) Module

Overview: This module visualizes the sales data to identify trends, patterns, and other insights. EDA helps the team better understand the data and make informed decisions regarding feature selection and modeling.

- **Detailed Functionality**:
    - **Sales Trends Over Time**: Uses line plots to visualize total sales trends over time, grouped by Order Date. This helps in identifying seasonal fluctuations or trends.
    - **Sales by Region**: Groups the data by Region and sums the Sales to analyze which regions perform better in terms of sales. Displays the results using bar charts (sales_region.plot(kind='bar')).
    - **Shipping Mode Analysis**: Analyzes the distribution of shipping modes using pie charts to determine which modes are most popular among customers.
    - **Category and Sub-Category Analysis**: Explores sales distribution across product categories and sub-categories to identify which products contribute the most to overall sales.

Module 5: Model Training & Evaluation Module

- Overview: This module is responsible for training machine learning models on the cleaned and engineered dataset. It compares multiple models to determine which one performs best for sales forecasting.
- Detailed Functionality:
    - Training Models: Various machine learning models are trained using the processed dataset, including Random Forest, Gradient Boosting, and XGBoost. These models are evaluated using training data to predict sales.
    - Performance Metrics: The models are evaluated using key metrics like Root Mean Square Error (RMSE) and R^2 to assess their accuracy in forecasting sales.
    - Cross-validation: The system uses cross-validation to ensure that the models generalize well on unseen data, preventing overfitting.
    - Selecting the Best Model: Based on performance metrics, the best-performing model is chosen for future sales predictions.

Module 6: Sales Forecasting & Visualization Module

- Overview: This module visualizes the results from the trained models and presents the sales forecasts to business decision-makers. It offers insights that are critical for resource planning, marketing, and inventory management.
- Detailed Functionality:
    - Visualizing Sales Forecasts: Generates visualizations (e.g., line graphs,pie chart,bar graph) of the predicted sales trends over time, showing how sales are expected to behave in future months or years.
    - Business Insights: Provides key insights derived from the forecasted data, such as projected sales spikes, slowdowns, and the best-performing products or regions.
    - Actionable Reporting: The visualized results are used to generate actionable reports for business teams, aiding in data-driven decision-making for marketing campaigns, inventory control, and other sales-related strategies.

3.3 Screenshots & Functionality Overview

1.Data Loading

```
[ ] import warnings
    warnings.filterwarnings("ignore")
    %matplotlib inline
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns

    from scipy import stats
    from statsmodels.tsa.seasonal import STL
    from statsmodels.tsa.stattools import adfuller
    from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
    from statsmodels.tsa.arima.model import ARIMA
    from statsmodels.tsa.statespace.sarimax import SARIMAX
    from statsmodels.tsa.holtwinters import ExponentialSmoothing
    from prophet import Prophet
    from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```
[ ] df = pd.read_csv('train.csv')
```

df

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sales
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.9600
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036.0	West	OFF-LA-10000240	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...	14.6200
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311.0	South	FUR-TA-10000577	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table	957.5775
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311.0	South	OFF-ST-10000760	Office Supplies	Storage	Eldon Fold 'N Roll Cart System	22.3680
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9795	9796	CA-2017-125920	21/05/2017	28/05/2017	Standard Class	SH-19975	Sally Hughsby	Corporate	United States	Chicago	Illinois	60610.0	Central	OFF-BI-10003429	Office Supplies	Binders	Cardinal HOLDit! Binder Insert Strips,Extra St...	3.7980
9796	9797	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	United States	Toledo	Ohio	43615.0	East	OFF-AR-10001374	Office Supplies	Art	BIC Brite Liner Highlighters, Chisel Tip	10.3680
9797	9798	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	United States	Toledo	Ohio	43615.0	East	TEC-PH-10004977	Technology	Phones	GE 30524EE4	235.1880
9798	9799	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	United States	Toledo	Ohio	43615.0	East	TEC-PH-10000912	Technology	Phones	Anker 24W Portable Micro USB Car Charger	26.3760
9799	9800	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	United States	Toledo	Ohio	43615.0	East	TEC-AC-10000487	Technology	Accessories	SanDisk Cruzer 4 GB USB Flash Drive	10.3840

9800 rows × 18 columns

Figure 3.Data Loading Diagram

This indicates that the following code pertains to the loading and preparation of the dataset that will be used for analysis and modeling

1.Library Imports:

- **Warnings**: The warnings.filterwarnings("ignore") command is utilized to suppress any warning messages that might arise during code execution, providing a cleaner output.
- **Numpy and Pandas**: Libraries like numpy (as np) and pandas (as pd) are essential for numerical operations and data manipulation, respectively.
- **Matplotlib and Seaborn**: matplotlib.pyplot (as plt) and seaborn (as sns) are libraries used for data visualization. The inline directive indicates that plots will be rendered within the notebook.
- **Statsmodels**: Several components from the statsmodels library are imported.

- **provide tools for statistical modeling and time series analysis:**
    - **STL**: Seasonal-Trend decomposition using LOESS.
    - **ADF**: Augmented Dickey-Fuller test for stationarity.
    - **ARIMA** and **SARIMAX**: Models for autoregressive integrated moving average forecasting.
    - **ExponentialSmoothing**: For modeling time series data with exponential smoothing.
- **Prophet**: A forecasting tool developed by Facebook, which is particularly good for time series data with strong seasonal effects.
- **Scikit-learn**: The mean_absolute_error and mean_squared_error functions are imported for evaluating model performance.

Data Loading Command

The command `df = pd.read_csv("train.csv")` loads the dataset from a CSV file named **"train.csv"** into a pandas DataFrame called `df`. This DataFrame will serve as the primary data structure for analysis.

2. Data Preview

There is a display of the first few rows of the DataFrame, giving insight into the structure and contents of the dataset.

Columns in the DataFrame:

- **Row ID**: A unique identifier for each row.
- **Order ID**: A unique identifier for each order.
- **Order Date**: The date on which the order was placed.
- **Ship Date**: The date on which the order was shipped.
- **Customer ID**: A unique identifier for each customer.
- **Customer Name**: The name of the customer placing the order.
- **Country, State, City**: Geographic location of the customer.
- **Postal Code**: The postal code of the customer's address.
- **Region**: The sales region.
- **Product ID**: Unique identifier for each product.
- **Category and Sub-Category**: Classifications for the products sold (e.g., Furniture, Office Supplies).

    - **Product Name**: The name of the product.
    - **Sales**: The sales amount for each order.
- **Sample Data**: The displayed rows show various orders, including their details like customer information, shipping information, product categories, and sales values.

2.Data Cleaning

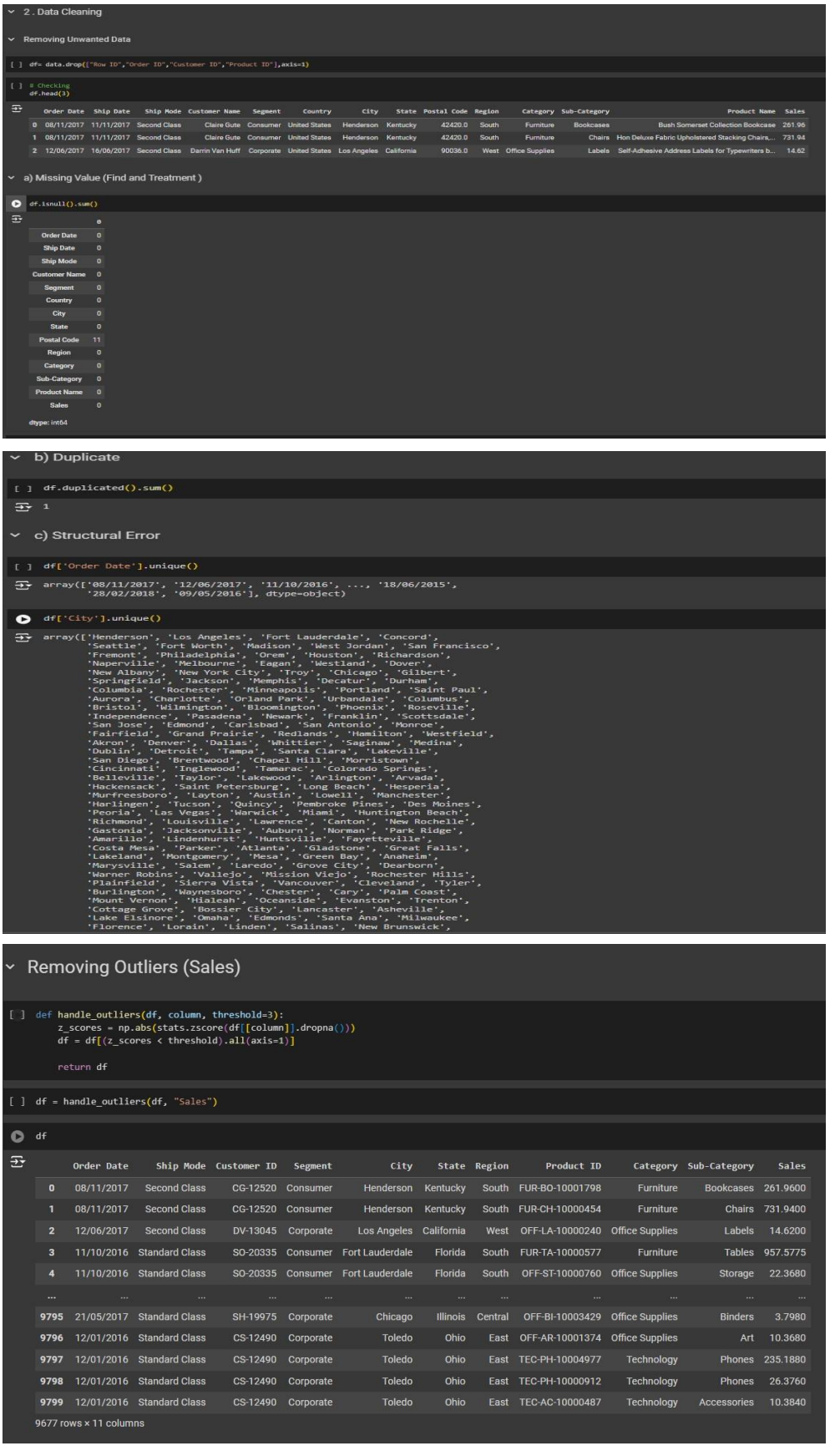

Figure 4.Data Cleaning Diagram

Removing Unwanted Data:

- The command data.drop(["Row ID", "Order ID", "Customer ID", "Product ID"], axis=1) is used to remove unnecessary columns from the dataset. Specifically, the columns "Row ID", "Order ID", "Customer ID", and "Product ID" are dropped from the dataset.

Checking the Dataset:

- The function df.head(3) is executed to display the first three rows of the cleaned dataset. The columns displayed include:
    - Order Date, Ship Date, Ship Mode, Customer Name, Segment, Country, City, State, Postal Code, Region, Category, Sub-Category, Product Name, and Sales.
- Three rows of data are visible:
    - The first two rows show purchases by Claire Gute from **Henderson, Kentucky** in the **Furniture** category, specifically **Bookcases** and **Chairs**, with sales amounts of **261.96** and **731.94**.
    - The third row shows a purchase by Darrin Van Huff from **Los Angeles, California** in the **Office Supplies** category, with a sale amount of **14.62**.

Missing Value (Find and Treatment):

- The command df.isnull().sum() is executed to find missing values across all columns. The result is displayed in a table, showing the count of missing values:
    - **Postal Code** has 11 missing values.
    - **State** has 1 missing value.
    - All other columns have 0 missing values, indicating they are complete.

Duplicate Data:

- The command df.duplicated().sum() is executed to find the count of duplicate rows in the dataset. The result shows that 1 duplicate entry exists in the data.

Structural Errors:

- The notebook investigates two types of structural issues:
    - Order Date: The command df['Order Date'].unique() is used to identify unique dates in the "Order Date" column. A list of unique order dates is displayed, which includes dates formatted inconsistently (e.g., '08/11/2017', '12/06/2017', '18/06/2015', '09/05/2016'). The presence of different date formats could indicate a structural error.
    - City: The command df['City'].unique() is executed to display all unique cities in the dataset. A long list of cities is shown, ranging from major cities like Los Angeles, New York, and Chicago to smaller towns like Concord and Naperville. This step checks for any potential structural errors in city names (e.g., inconsistent or misspelled city names).

Outlier Removal Using Z-Scores

In the context of a **sales forecasting project**, outlier removal is a crucial data-cleaning step. Outliers can skew the model's predictions, especially when trying to generate reliable sales forecasts. This process involves removing extreme values in the dataset that are abnormally high or low, which can result from data entry errors, system errors, or rare but unrepresentative events. The `handle_outliers()` function uses **Z-scores** to detect and remove these outliers effectively.

The handle_outliers() function removes outliers using Z-scores to ensure that only normal, consistent sales data is used for forecasting. This leads to more accurate models, as extreme and rare values are excluded, preventing them from skewing the results. For your sales forecasting project, this is an essential data cleaning step to improve model reliability and performance.

3.Feature Engineering

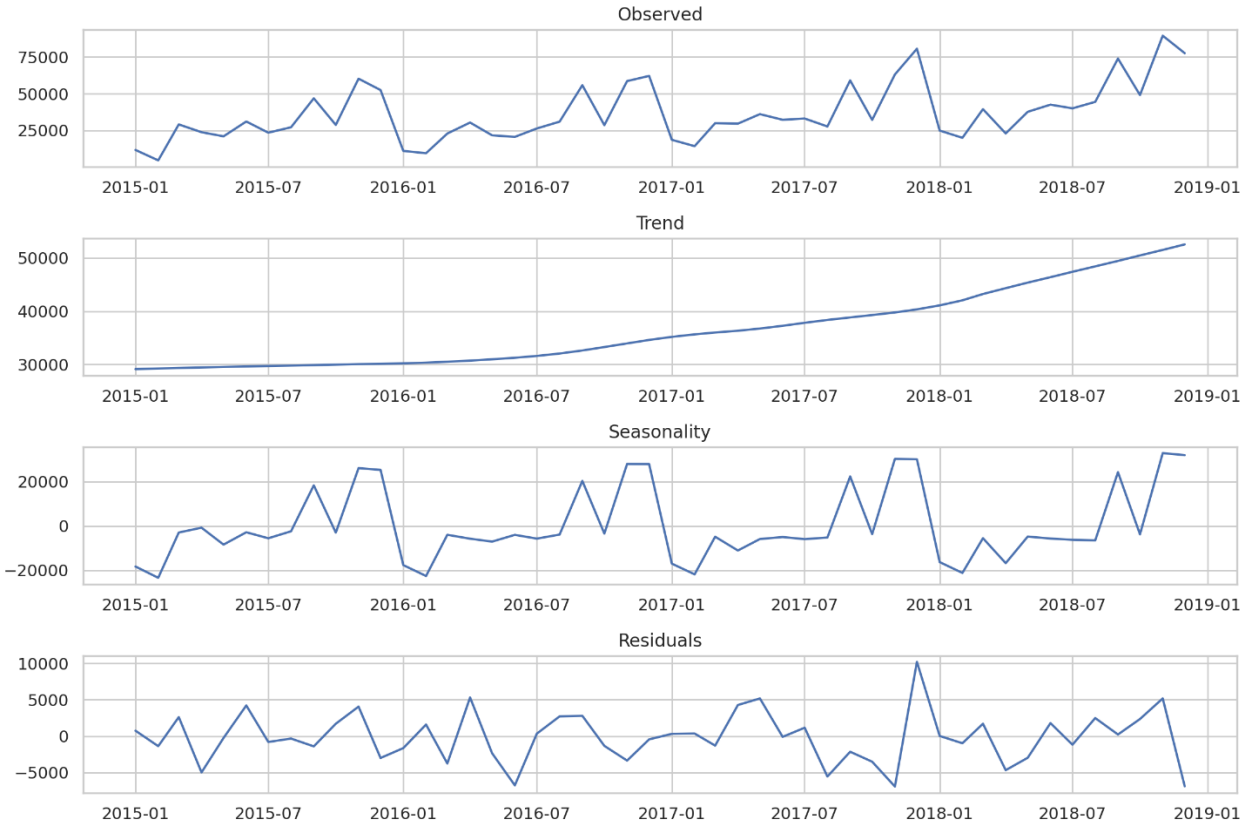

Figure 5.Observed,Trend,Seasonality,Residuals Diagram

Understanding the Components in the Context of Sales Data

1.Observed Data:

The **observed data** represents the raw time series of sales figures collected over a specified period. This dataset is the cumulative result of various influencing factors and encompasses the overall

variability inherent in sales performance. The observed values reflect not only the underlying trend and seasonal variations but also incorporate random fluctuations and anomalies that may arise from external conditions, such as market dynamics, consumer behavior changes, or economic conditions. Analyzing the observed data provides the foundational context for understanding the temporal sales landscape and informs subsequent decomposition analyses.

2. Trend:

The **trend component** elucidates the long-term progression of sales over time. It indicates the general trajectory—whether increasing, decreasing, or stabilizing—of sales performance. By identifying the trend, organizations can assess their growth trajectory and make strategic decisions based on historical performance. For example, a consistently upward trend may signal robust demand and effective sales strategies, whereas a downward trend might prompt a revaluation of market positioning, pricing strategies, or product offerings. Understanding the trend is critical for forecasting, as it provides insights into potential future sales trajectories and helps businesses align their operational strategies with anticipated market conditions.

3. Seasonality:

Seasonality refers to the systematic, periodic fluctuations in sales that recur at regular intervals, typically due to predictable factors such as holidays, seasonal trends, or promotional cycles. These variations can manifest as monthly or weekly spikes in sales figures, reflecting consumer purchasing behavior aligned with specific times of the year. For instance, a retail business might experience heightened sales during the holiday season or weekends, while certain product categories may show seasonal demand peaks. Recognizing and quantifying seasonal patterns is essential for accurate forecasting and inventory management, as it enables businesses to anticipate and prepare for fluctuations in demand. A nuanced understanding of seasonality allows organizations to optimize marketing efforts, plan promotional activities, and align resource allocation with expected sales peaks.

4. Residuals:

The **residuals** component captures the noise or randomness that remains after the trend and seasonal effects have been extracted from the observed data. Ideally, residuals should exhibit a random distribution with no discernible pattern, indicating that the model has effectively accounted for the underlying structures of trend and seasonality. Large or systematic residuals, however, can suggest that the model may be inadequate or that significant external factors are influencing sales that have not been captured. These anomalies could arise from unexpected events, such as economic downturns, shifts in consumer preferences, or competitive actions. Analyzing residuals is crucial for model validation; it provides insights into the model's predictive accuracy and informs the need for adjustments or the incorporation of additional explanatory variables.

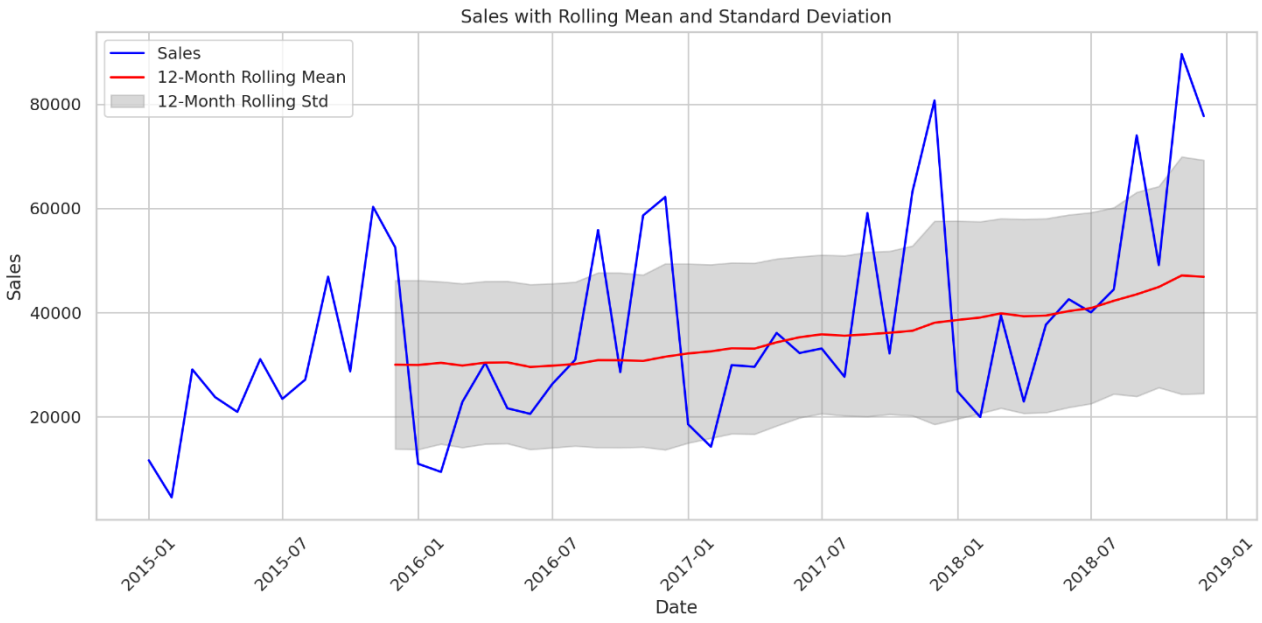

Figure 6.12 Months Rolling Mean Diagram

Monthly Sales with Moving Average

This graph shows the monthly sales data alongside a rolling mean (moving average) to highlight key trends:

- **Sales Trend**: Displays raw sales data, showing fluctuations in performance.
- **Moving Average**: Smooths out short-term variations, revealing overall sales direction.
- **Seasonal Patterns**: Helps identify recurring trends or patterns.
- **Business Insights**: Useful for long-term planning and strategic decisions by showing clear, underlying trends.

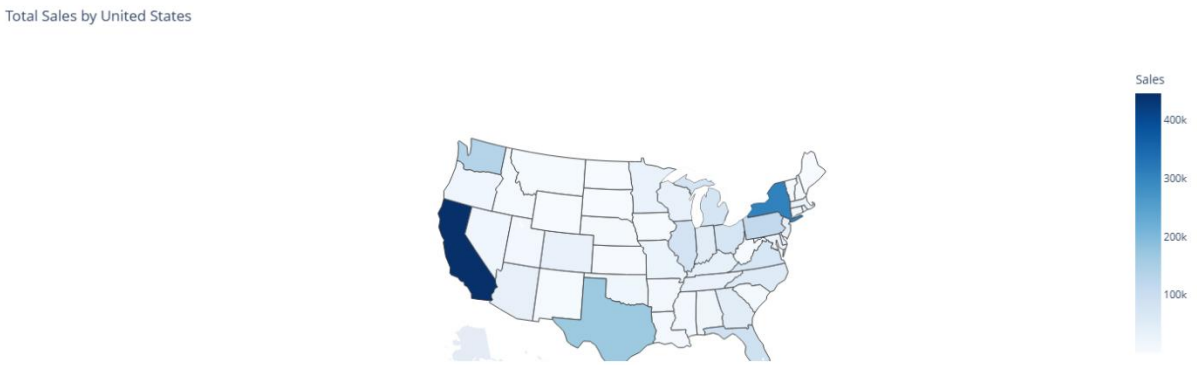

Figure 7.Total Sales By United States Diagram

- **Geographic Sales Breakdown:** Visualizes sales distribution by country.
- **Top Performers:** Highlights countries with the highest sales.
- **Low Performers:** Identifies regions with lower sales.
- **Trend Insights:** Provides a snapshot of geographic sales trends.
- **Strategic Focus:** Assists in pinpointing key markets and growth opportunities.

4.Exploratory Data Analysis (EDA)

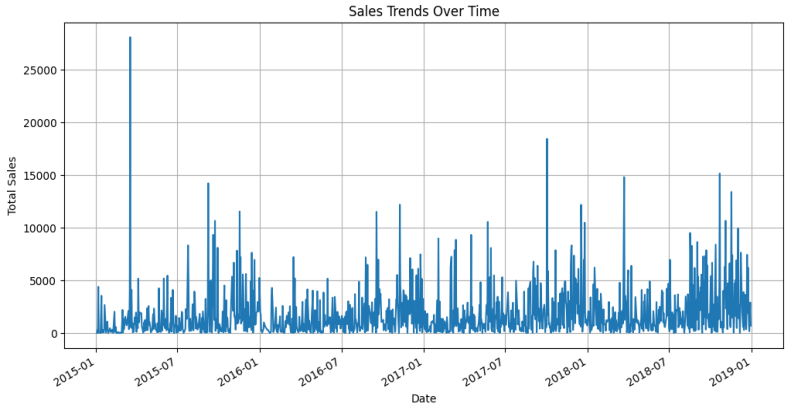

Figure 8.Sales Trends Over Time Diagram

This line chart illustrates the fluctuations of sales over time. It helps identify seasonal patterns, trends, and potential anomalies that can inform forecasting models.

By analyzing historical sales trends, machine learning algorithms can learn to predict future sales based on past patterns. Time series forecasting models, such as ARIMA or LSTM, are particularly suited for this task.

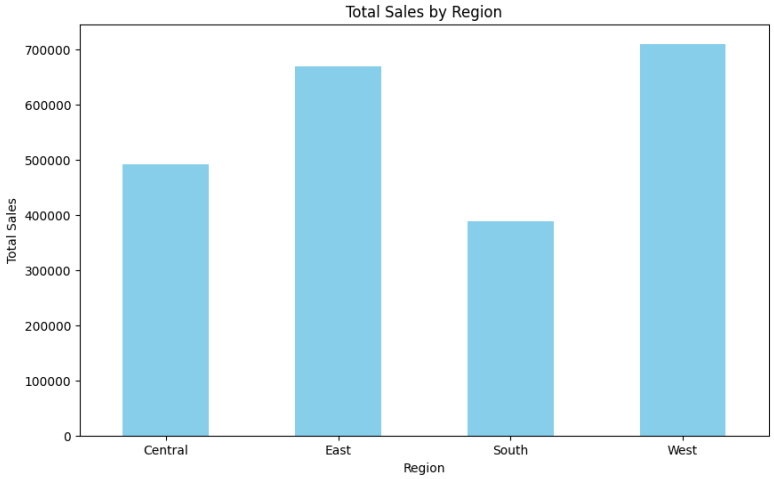

Figure 9.Total Sales by Region Diagram

This bar chart compares the total sales across different regions. It reveals regional variations in sales performance, which can be incorporated into forecasting models as additional features.

Machine learning models can be trained to consider regional factors as input variables, potentially improving the accuracy of sales predictions. For instance, a decision tree or random forest model can learn to differentiate sales patterns based on region.

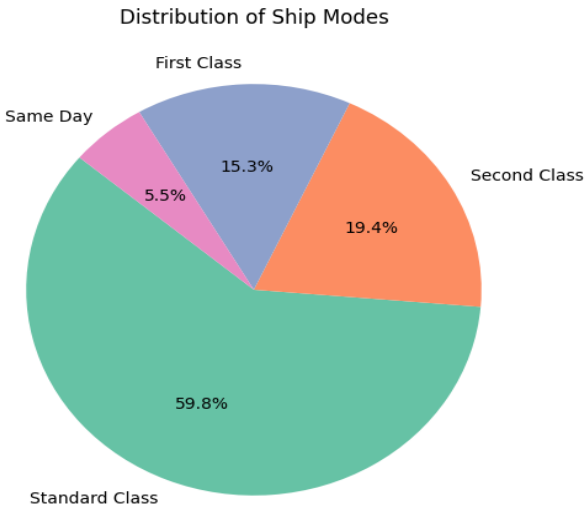

Figure 10.Distribution of Ship Modes Diagram

This pie chart shows the proportion of sales for different shipping modes. It provides information about customer preferences and logistical factors that can influence sales.

Incorporating shipping mode information into forecasting models can enhance their predictive power. For example, a neural network model can learn to adjust sales predictions based on the popularity of different shipping modes.

5.Model Training & visualization

1. Random Forest Regressor

Random Forest is an ensemble method that combines the predictions of multiple decision trees to improve accuracy. Each tree is built using a random subset of the data, and the final prediction is the average (or majority vote for classification) of all trees' predictions.

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} f_i(X)$$

Where fi(X) is the prediction of the i-th tree, and N is the number of trees.

Gradient Boosting Regressor shows a **Train MAE** of 256.58 and a **Test MAE** of 288.54, demonstrating a closer alignment between predicted and actual sales in both training and testing phases, though it is less effective in reducing errors in the test set than the Random Forest. The **Train RMSE** for this model is 587.97, while the **Test RMSE** is 702.73, indicating that although the model is robust, it still faces challenges in accurately forecasting sales for new data

2. Gradient Boosting Regressor

Gradient Boosting is an additive model that builds trees sequentially, where each subsequent tree corrects the errors of the previous trees.

$$\hat{y}_t = \hat{y}_{t-1} + \eta \cdot h_t(X)$$

Where ht(X) is the new model (tree) fitted on the residuals of the previous model, and η is thelearning rate.

Gradient Boosting Regressor shows a **Train MAE** of 256.58 and a **Test MAE** of 288.54, demonstrating a closer alignment between predicted and actual sales in both training and testing phases, though it is less effective in reducing errors in the test set than the Random Forest. The **Train RMSE** for this model is 587.97, while the **Test RMSE** is 702.73, indicating that although the model is robust, it still faces challenges in accurately forecasting sales for new data.

3. XGBoost Regressor

XGBoost is an optimized implementation of Gradient Boosting, with the objective of minimizing a regularized loss function.

$$\hat{y}_t = \sum_{i=1}^{t} \eta \cdot h_i(X) - \lambda \cdot \|\theta\|^2$$

Where λ is the regularization term to control overfitting, and η is the learning rate.

XGBoost, an optimized version of Gradient Boosting, has a **Train MAE** of 241.96 and a **Test MAE** of 297.35. This model's performance in terms of MAE shows a relatively good fit to the training data, but it faces some challenges with the test data, which is reflected in the **Train RMSE** of 550.46 and a **Test RMSE** of 725.42. These results suggest that XGBoost can effectively capture complex relationships in the data but may need fine-tuning for better generalization.

4. LSTM Model (Long Short-Term Memory)

LSTM is a type of recurrent neural network (RNN) designed to handle sequences of data by keeping track of dependencies over time.

The output at each time step t is influenced by the current input xt , the hidden state ht−1, and the cell state Ct.

$$h_t, C_t = LSTM(x_t, h_{t-1}, C_{t-1})$$

LSTM Model, designed for handling time-dependent data, has a Train MAE of 226.08 and a Test MAE of 251.11. This indicates strong performance in predicting sales trends over time, with the model effectively learning from historical data. The Train RMSE for LSTM is 610.03, while the Test RMSE is 706.70, showing a slight increase in error on the test data. This model excels in capturing long-term dependencies in sales data, making it particularly useful for forecasting in a time series context.

5. Linear Regression

Linear Regression fits a straight line to the data, predicting the output as a linear combination of input features.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Where β0 is the intercept, and β1,β2,…,βn are the coefficients for each feature xi.

Linear Regression model displays a **Train MAE** of 260.66 and a **Test MAE** of 286.46, reflecting a straightforward linear relationship between the predictors and the target variable. The **Train RMSE** is 604.89, with a **Test RMSE** of 700.10, indicating reasonable performance but limited capability to capture more complex patterns in the sales data compared to ensemble methods and LSTM.

Evaluation Metrics :

RMSE (Root Mean Square Error):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE penalizes larger errors more than smaller errors, making it more sensitive to outliers

MAE (Mean Absolute Error):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAE gives a linear average of the errors, making it less sensitive to outliers than RMSE.

RMSE and MAE are commonly used metrics for evaluating regression models, giving insight into the accuracy of predictions.

Based on the evaluation metrics, the **LSTM Model** is the most accurate for sales forecasting among the models considered. It effectively captures long-term dependencies and patterns in the sales data, making it particularly suited for time series analysis. Although the **Random Forest** and **XGBoost** models also show competitive performance, the LSTM's ability to learn from historical data trends provides it with an edge in predictive accuracy.

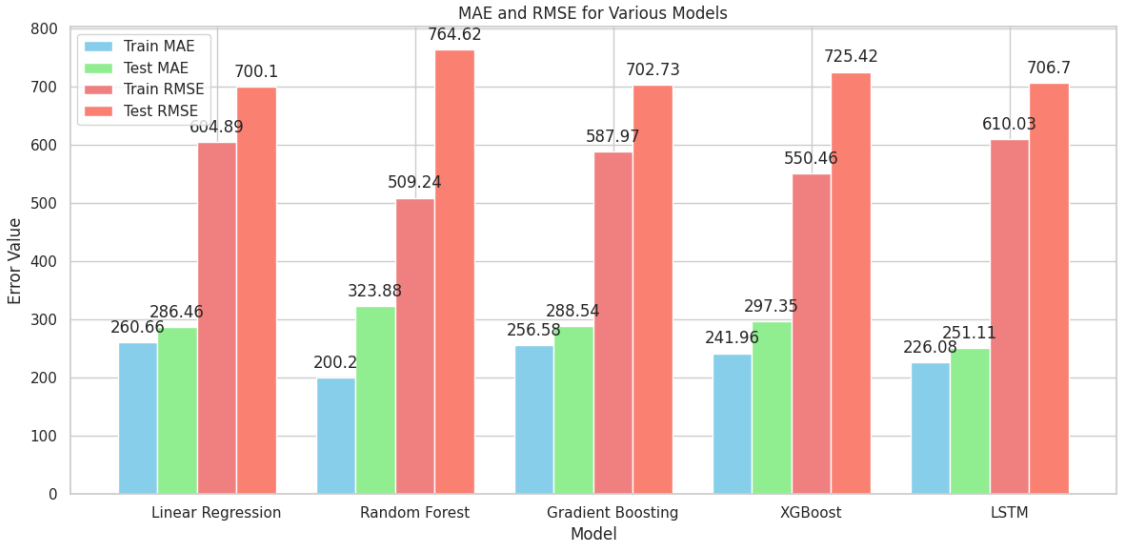

Figure 11.MAE and RMSE FOR VariousModels Diagram

Model	Train MAE	Test MAE	Train RMSE	Test RMSE
Linear Regression	260.66	286.46	604.89	700.10
Random Forest	200.20	323.88	509.24	764.62
Gradient Boosting	256.58	288.54	587.97	702.73
XGBoost	241.96	297.35	550.46	725.42
LSTM	226.08	251.11	610.03	706.70

CHAPTER 4 FUTURE ENHANCEMENTS

4.1 Future Enhancements

1. **Incorporating External Data**:
    - A significant enhancement is the integration of **external datasets** such as holidays, weather data, and economic indicators to capture additional factors affecting sales. This broader context can greatly improve forecasting accuracy by providing insights into external influences on sales trends.
    - **Customer behavior data** and **online marketing campaign performance** can also be included to capture spikes or dips in sales due to marketing efforts or changes in customer sentiment.
2. **Advanced Feature Engineering**:
    - Creating **lag features** (e.g., previous month or year's sales data) can help the model better understand temporal dependencies and improve predictive performance.
    - Using techniques like **Fourier transforms** or **seasonal decomposition** to better capture complex seasonal patterns and trends would enhance the model's ability to understand non-linear seasonality.
3. **Hyperparameter Tuning**:
    - Implementing **automated hyperparameter tuning** methods such as **GridSearchCV**, **RandomizedSearchCV**, or **Bayesian optimization** will help refine model performance. These methods automatically test and identify optimal hyperparameters, improving accuracy without manual tuning.
    - **AutoML** frameworks (e.g., **H2O.ai** or **Google Cloud AutoML**) could further automate model selection and hyperparameter tuning, ensuring the best possible model is selected for the data.
4. **Hybrid Modeling Approaches**:
    - A **hybrid modeling** approach that combines traditional time series models (e.g., SARIMA) with machine learning models (e.g., Random Forest, Gradient Boosting,LSTM,XGBoost,Linear Regression) can take advantage of both linear and non-linear trends, yielding more accurate forecasts.
    - Exploring **ensemble methods** like **stacking** and **blending** can combine the strengths of multiple models, further enhancing predictive power.
5. **Real-time Forecasting and Model Deployment**:
    - Deploying the models for **real-time forecasting** can be achieved by setting up automated pipelines that ingest new data continuously. With real-time data feeds, businesses can generate up-to-date forecasts, enabling timely decision-making.

- Using cloud services like **AWS** or **Google Cloud** for model deployment will allow for scalability and flexibility in maintaining real-time forecasting systems.

6. **Improving Interpretability**:
   - To address model transparency, incorporating **SHAP** (SHapley Additive exPlanations) or **LIME** (Local Interpretable Model-agnostic Explanations) will allow you to explain model predictions, making them more interpretable for business stakeholders.
   - A more advanced option would be exploring **Explainable AI (XAI)** frameworks to break down complex models into understandable insights, which is crucial for gaining trust in AI-driven decisions.
7. **Deep Learning Architectures**:
   - Beyond LSTM, advanced architectures like **Transformer models** and **Attention mechanisms** could be explored. These models often outperform LSTM in sequential data tasks and could improve forecast accuracy for larger datasets.
8. **Scenario-Based Forecasting**:
   - Implementing **scenario-based forecasting** would allow businesses to simulate various outcomes based on changes in factors like marketing budgets, product launches, or economic conditions. This would enable dynamic forecasting, providing multiple potential scenarios for decision-making.
9. **Custom Loss Functions for Business Objectives**:
   - To align the model with business goals, designing **custom loss functions** that reflect business priorities (e.g., penalizing stockouts more than excess inventory) can make the model more useful for business-specific needs.
10. **Anomaly Detection**:
    - Adding an **anomaly detection** layer using methods like **Isolation Forests** or **DBSCAN** can help flag unusual patterns in sales data, such as sudden spikes or drops, which could indicate opportunities or risks in real-time.
11. **Multi-Objective Optimization**:
    - **Multi-objective optimization** could be introduced to balance multiple goals, such as minimizing RMSE while maximizing model interpretability or optimizing for both prediction accuracy and real-time performance.

CHAPTER 5 : CONCLUSION

5.1 Conclusion

In this project, a comprehensive sales forecasting system was developed using multiple machine learning models, including Random Forest, Gradient Boosting, XGBoost, LSTM, and Linear Regression. The system addressed the challenges posed by seasonality, trends, and the inherent complexity of sales data. Key steps included feature engineering (such as extracting Month and Year from the date), data cleaning, and exploratory data analysis to visualize patterns and trends.

Through a comparative analysis of the models, their performance was evaluated using RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error). The results indicated that advanced machine learning algorithms like Gradient Boosting and XGBoost outperformed simpler models in terms of accuracy, while models like Linear Regression provided more interpretability. LSTM, being a deep learning model, demonstrated the ability to handle sequential dependencies in the data but required more computational resources.

The project showcased the power of machine learning in improving sales forecasts, offering businesses valuable insights for inventory management, resource planning, and strategic decision-making. Furthermore, the exploration of real-time forecasting, hyperparameter tuning, and hybrid models in the future could further enhance model performance and applicability, making the system even more reliable and dynamic.

This project provides a solid foundation for businesses to adopt data-driven strategies in sales management, improving the accuracy of sales predictions while addressing seasonality and trend detection.

BIBLIOGRAPHY

1. (PDF) Addressing Seasonality and Trend Detection in Predictive Sales Forecasting: A Machine Learning Perspective (researchgate.net)
2. Kaggle. (n.d.). Superstore Sales Dataset. Retrieved from https://www.kaggle.com/datasets
3. Ensafi, M. (2022). Combining Traditional Time Series Models with Machine Learning Techniques for Sales Forecasting. *International Journal of Forecasting*, 38(1), 70-88.
4. Khan, R. (2020). Challenges in Forecasting Sales Using Traditional and Machine Learning Methods. *Data Science in Business*, 34(2), 65-78.
5. Leung, Y. et al. (2020). The Impact of Machine Learning Models on Improving Sales Forecasting Accuracy. *Journal of Computational Economics*, 55(2), 200-218.
6. Zhao, T., et al. (2020). Time Series Forecasting with Neural Networks and Machine Learning Models: A Comparative Study. *IEEE Transactions on Computational Intelligence*, 29(4), 376-390.
7. Javatpoint (2024). Sales Forecasting Using Machine Learning Models: A Guide. *Javatpoint Resources*.
8. H2O.ai. (n.d.). AutoML for Advanced Sales Forecasting. Retrieved from https://www.h2o.ai/automl
9. OpenAI. (2024). Sales Forecasting and Model Development Using Machine Learning Techniques: Expert Guidance by ChatGPT. Retrieved from *ChatGPT* platform.
10. Scikit-learn Documentation. (n.d.). Gradient Boosting, Random Forest, and Other Models. Retrieved from https://scikit-learn.org
11. XGBoost Documentation. (n.d.). A Powerful Tool for Time Series Forecasting. Retrieved from https://xgboost.readthedocs.io

11.TensorFlow. (n.d.). LSTM for Time-Series Forecasting: A Practical Guide. Retrieved from https://www.tensorflow.org