**SUPPLY CHAIN ANALYSIS**



**ISDS 577: Master of Science Capstone Seminar**

Dr. Daniel Soper

College of Business and Economics

California State University, Fullerton

**GROUP 6**

ABHISHEK UDANSHIV

ARYAN ALPESH AGARWAL

CHAEYA LEE

HANEELA REDDY AVUTHU

MANAS GARG

SMRUTI JAGDISH JADHAV

Table of Contents

[INTRODUCTION 4](#_Toc197979522)

[BACKGROUND & OBJECTIVES 4](#_Toc197979523)

[EXPECTED OUTCOMES 5](#_Toc197979524)

[PROBLEM STATEMENT 5](#_Toc197979525)

[DATASET EXPLANATION 6](#_Toc197979526)

[**DATA COLUMN DESCRIPTION** 6](#_Toc197979527)

[**Dashboard 1 (Customer Segmentation)** 11](#_Toc197979528)

[**Dashboard 2 (Key Sales Insights and Delivery Risk Dashboard)** 14](#_Toc197979529)

[RESEARCH QUESTIONS 18](#_Toc197979530)

[RESEARCH QUESTION 1: How can we segment customers based on their purchasing Behavior, shipping preferences, and order history? 18](#_Toc197979531)

[**Dataset Description** 18](#_Toc197979532)

[**Exploratory Data Analysis (EDA)** 19](#_Toc197979533)

[**Feature Engineering (RFM Calculation)** 20](#_Toc197979534)

[**Models** 21](#_Toc197979535)

[**Conclusion** 31](#_Toc197979536)

[RESEARCH QUESTION 2: Can we predict future sales based on past transaction data, seasonality, and regional sales trends? 33](#_Toc197979537)

[**Dataset Description** 33](#_Toc197979538)

[**Data Preprocessing** 34](#_Toc197979539)

[**Exploratory Data Analysis** 36](#_Toc197979540)

[**Methodology** 38](#_Toc197979541)

[**Conclusion** 56](#_Toc197979542)

[RESEARCH QUESTION 3: Can we identify fraudulent transactions based on order patterns, payment types, and late delivery risks? 57](#_Toc197979543)

[**Data Preparation** 57](#_Toc197979544)

[**Exploratory Data Analysis** 59](#_Toc197979545)

[**Model Building** 59](#_Toc197979546)

[**Insights & Business Value** 71](#_Toc197979547)

[CONCLUSION 71](#_Toc197979548)

[RECOMMENDATIONS AND FUTURE STEPS 73](#_Toc197979549)

[BUSINESS IMPLICATIONS 73](#_Toc197979550)

# **INTRODUCTION**

## **BACKGROUND & OBJECTIVES**

This report presents an in-depth analysis of a large transactional dataset derived from a retail business, aimed at uncovering key patterns in customer behavior, detecting fraudulent activities, and forecasting future sales trends. With over 180,000 records and 53 features, the dataset provides a comprehensive view of the business’s operations across multiple dimensions.

The primary objective is to apply statistical techniques and advanced machine learning models to extract actionable insights that can enhance decision-making across marketing, finance, operations, and customer service functions. The analysis is guided by three critical research questions focusing on customer segmentation, fraud detection, and sales prediction.

## EXPECTED OUTCOMES

* **Targeted Marketing**: By segmenting customers based on purchasing behavior and engagement, businesses can tailor promotions and loyalty programs for higher effectiveness.
* **Fraud Prevention**: Identifying patterns indicative of fraudulent activity enables proactive detection and mitigation, protecting both revenue and reputation.
* **Accurate Sales Forecasting**: Time series models provide a predictive view of sales trends, supporting demand planning and inventory management.
* **Data-Driven Decision-Making**: The insights derived from this analysis serve as a foundation for strategic initiatives across operations, financial planning, and customer engagement.

## **PROBLEM STATEMENT**

The project utilizes a large transactional dataset from a global retail business to uncover insights into customer Behavior, fraudulent activity, and sales trends. We deployed statistical analysis and machine learning techniques in the process. The expected outcome was to have the analysis that provides actionable insights for targeted marketing for each segment, fraud prevention strategies, and accurate sales forecasting. Based on this, we aimed to support data-driven decision-making in operations, finance, and customer relations.

## **DATASET EXPLANATION**

The dataset used in this analysis is the DataCo Supply Chain Dataset, comprising detailed sales and customer transactions from a U.S.-based retail supply chain operation. This data includes 180,519 orders across 20,652 unique customers, offering insights into customer Behavior, sales performance, order processing, and payment preferences.

### **DATA COLUMN DESCRIPTION**

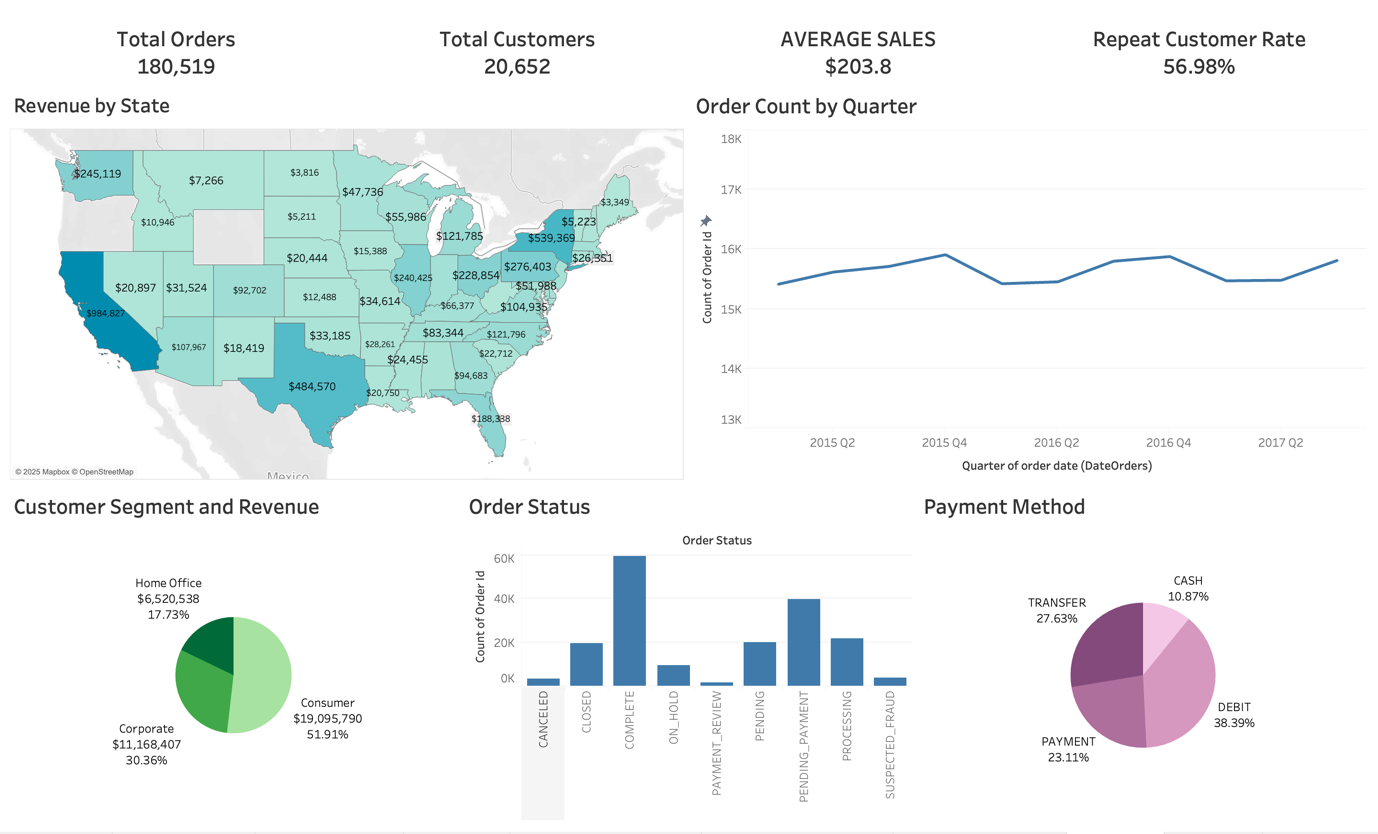
* TYPE: Type of transaction made
* Type Days for shipping (real): Actual shipping days of the purchased product
* Days for shipment (scheduled): Days of scheduled delivery of the purchased product
* Benefit per order: Earnings per order placed
* Sales per customer: Total sales per customer made per customer
* Delivery Status: Delivery status of orders: Advance shipping, Late delivery, Shipping cancelled, Shipping on time
* Late\_delivery\_risk: Categorical variable that indicates if sending is late (1), it is not late (0).
* Category Id: Product category code
* Category Name: Description of the product category
* Customer City: City where the customer made the purchase
* Customer Country: Country where the customer made the purchase
* Customer Email: Customer's email
* Customer Fname: Customer First name
* Customer Lname: Customer Last Name
* Customer Id: Customer ID
* Customer Password: Masked customer key
* Customer Segment: Types of Customers: Consumer, Corporate, Home Office
* Customer State: State to which the store where the purchase is registered belongs
* Customer Street: Street to which the store where the purchase is registered belongs
* Customer Zipcode: Customer Zipcode
* Department Id: Department code of store
* Department Name: Department name of store
* Latitude: Latitude corresponding to location of store
* Longitude: Longitude corresponding to location of store
* Market: Market to where the order is delivered: Africa, Europe, LATAM, Pacific Asia, USCA
* Order City: Destination city of the order
* Order Country: Destination country of the order
* Order Customer Id: Customer order code
* order date (DateOrders): Date on which the order is made
* Order Id: Order code
* Order Item Cardprod Id: Product code generated through the RFID reader
* Order Item Discount: Order item discount value
* Order Item Discount Rate: Order item discount percentage
* Order Item Id: Order item code
* Order Item Product Price: Price of products without discount
* Order Item Profit Ratio: Order Item Profit Ratio
* Order Item Quantity: Number of products per order
* Sales: Value in sales
* Order Item Total: Total amount per order
* Order Profit Per Order: Order Profit Per Order
* Order Region: Region of the world where the order is delivered : Southeast Asia ,South Asia ,Oceania ,Eastern Asia, West Asia , West of USA , US Center , West Africa, Central Africa ,North Africa ,Western Europe ,Northern , Caribbean , South America ,East Africa ,Southern Europe , East of USA ,Canada ,Southern Africa , Central Asia , Europe , Central America, Eastern Europe , South of USA
* Order State: State of the region where the order is delivered
* Order Status: Order Status: COMPLETE, PENDING, CLOSED, PENDING\_PAYMENT, CANCELED, PROCESSING, SUSPECTEDFRAUD, ON\_HOLD, PAYMENT\_REVIEW
* Product Card Id: Product code
* Product Category Id: Product category code
* Product Description: Product Description
* Product Image: Link of visit and purchase of the product
* Product Name: Product Name
* Product Price: Product Price
* Product Status: Status of the product stock: If it is 1 not available, 0 the product is available
* Shipping date (DateOrders): Exact date and time of shipment
* Shipping Mode: The following shipping modes are presented: Standard Class, First Class, Second Class, Same Day

The DataCo Supply Chain dataset offers comprehensive coverage of customer Behavior, transaction details, and operational metrics. The visualizations generated using Tableau provide actionable insights into:

* High-performing states and segments
* Operational bottlenecks in order status
* Seasonal and quarterly sales patterns
* Payment method usage and customer loyalty

These insights are essential for strategic decision-making across marketing, logistics, finance, and operations.

### **Dashboard 1**

*Figure 1: Dashboard Snapshot from Tableau*

**1. Total Orders, Total Customers, Average Sales, and Repeat Customer Rate (KPIs)**

* **Total Orders: 180,519**  
  Represents the total number of orders placed by customers across the dataset.
* **Total Customers: 20,652**  
   Indicates the number of unique customers who placed orders.
* **Average Sales: $203.8**  
  This metric is the average sales amount per order, calculated by dividing total revenue by total orders.
* **Repeat Customer Rate: 56.98%**  
  Shows the percentage of customers who placed more than one order, highlighting customer retention and loyalty.

**2. Revenue by State (Choropleth Map)**

* This map visualizes revenue generated from different U.S. states.
* **High-revenue states** like California ($984,827), New York ($539,369), and Texas ($484,570) are shaded darker, indicating stronger sales performance.
* **Low-revenue states** like North Dakota, Wyoming, and Vermont show lighter shades, reflecting lower sales volume.
* This chart helps in geographic performance analysis and identifying market strengths and weaknesses by location.

**3. Order Count by Quarter (Line Chart)**

* Displays the number of orders placed in each quarter from 2015 Q2 to 2017 Q2.
* The trend shows **seasonal consistency** with some minor fluctuations, indicating stable sales volume.
* Helpful in identifying sales trends, seasonal demand, and potential for forecasting.

**4. Customer Segment and Revenue (Pie Chart)**

* Categorizes total revenue by customer segment:
  + **Consumer**: $19M (51.91%)
  + **Corporate**: $11.2M (30.36%)
  + **Home Office**: $6.5M (17.73%)
* This distribution highlights that the **Consumer segment is the most significant revenue contributor**, helping with strategic planning on which segment to target for future marketing efforts.

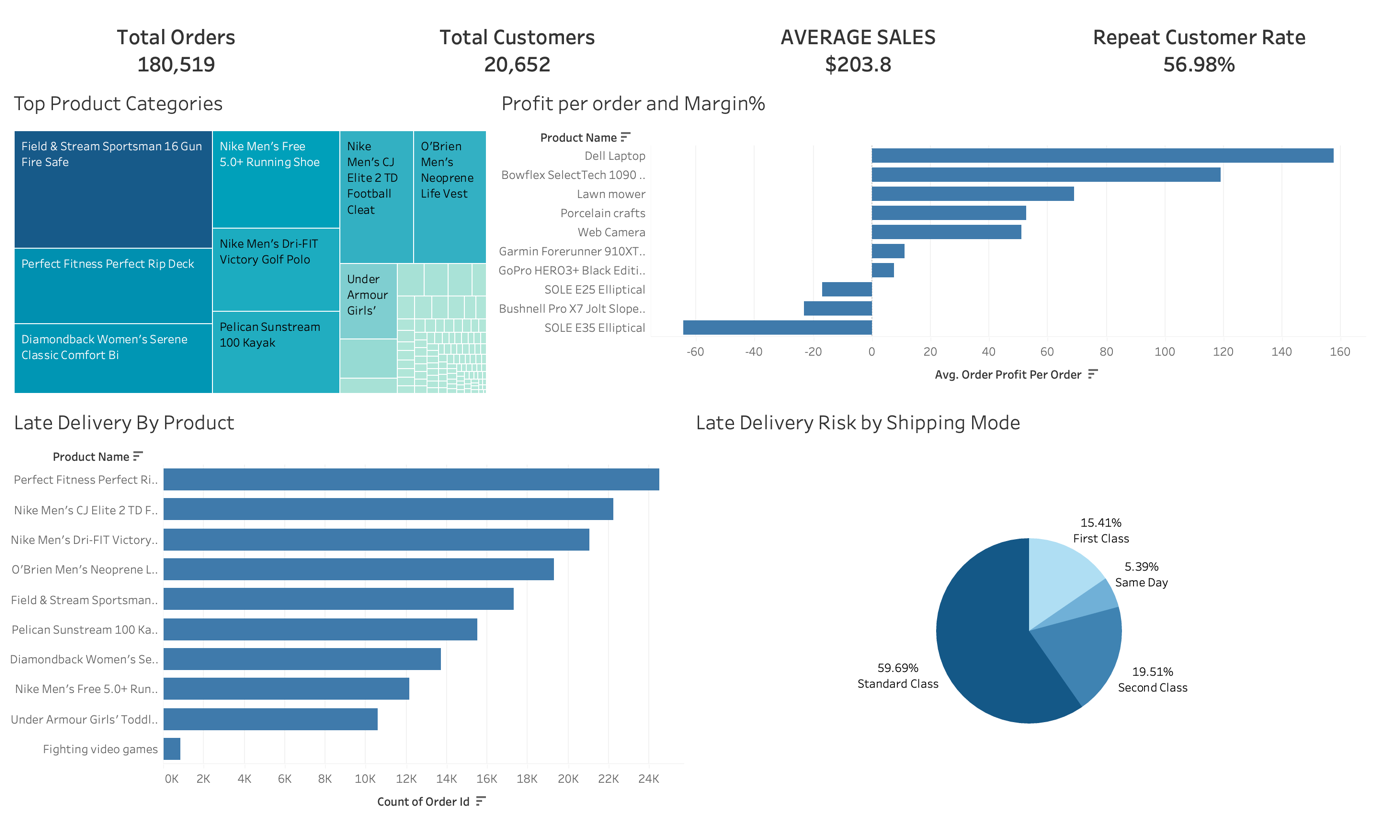
**5. Order Status (Bar Chart)**

* Depicts the number of orders under each status category:
  + **Complete** orders dominate the chart (~60K).
  + Other statuses like **Pending Payment**, **Processing**, and **Closed** follow.
  + Minimal counts for **Cancelled** or **Suspected Fraud**.
* This is critical for **operations and logistics analysis**, helping identify fulfillment bottlenecks or issues with order processing.

**6. Payment Method (Pie Chart)**

* Breaks down how customers are paying:
  + **Debit**: 38.39%
  + **Transfer**: 27.63%
  + **Payment** (likely credit or card): 23.11%
  + **Cash**: 10.87%
* Insights into **customer payment preferences** are essential for finance and planning additional payment method support.

### **Dashboard 2**

*Figure 2: Dashboard Snapshot from Tableau*

**1. Total Orders, Total Customers, Average Sales, and Repeat Customer Rate (KPIs)**

* **Total Orders: 180,519**  
   Represents the total number of orders placed by customers across the dataset.
* **Total Customers: 20,652**  
   Indicates the number of unique customers who placed orders.
* **Average Sales: $203.8**  
  This metric is the average sales amount per order, calculated by dividing total revenue by total orders.
* **Repeat Customer Rate: 56.98%**  
  Shows the percentage of customers who placed more than one order, highlighting customer retention and loyalty.

**2. Top Product Categories (Tree Map - Top Left)**

This treemap visualizes the best-selling or most frequently ordered products. Each block represents a product, with size indicating order volume. Notable products include:

* *Field & Stream Sportsman 16 Gun Fire Safe*
* *Nike Men’s Free 5.0+ Running Shoe*
* *Nike Men’s CJ Elite 2 TD Football Cleat*
* *Perfect Fitness Rip Deck*

This chart helps identify which products drive most sales and customer attention and informs inventory and marketing strategies.

**3. Profit Per Order and Margin % (Horizontal Bar Chart - Top Right)**

This chart compares the average profit per order for various products. It also implicitly represents margin efficiency:

* *SOLE E35 Elliptical* and *Bushnell Pro X7 Jolt Slope* yield the highest earnings per order.
* On the contrary, products like *Dell Laptop* and *Bowflex SelectTech 1090* result in negative profit margins, suggesting either heavy discounting or high costs.

This visualization is critical for pricing strategies and discontinuing unprofitable products.

**4. Late Delivery by Product (Horizontal Bar Chart - Bottom Left)**

This chart highlights which products are most often associated with late deliveries based on the count of delayed orders:

* *Perfect Fitness Rip Deck* and *Nike Men’s CJ Elite 2 TD Football Cleat* are leading in late deliveries, each with over 20,000 cases.
* It helps identify fulfillment issues potentially caused by supplier delays or inventory gaps for specific products.

**5. Late Delivery Risk by Shipping Mode (Pie Chart - Bottom Right)**

This pie chart breaks down the share of late delivery risk by shipping mode:

* *Standard Class* accounts for the majority (59.69%) of late deliveries.
* Other categories include *Second Class* (19.51%), *First Class* (15.41%), and *Same Day* (5.39%).

This emphasizes that lower-tier shipping methods are more prone to delays, guiding operational improvements and customer expectations.

**Key Takeaways:**

* The business maintains a strong **repeat purchase rate**, suggesting healthy customer retention.
* Certain products have high sales and **delivery issues**, pointing to potential operational bottlenecks.
* Shipping mode selection significantly influences delivery performance, with **Standard Class** being the most delay-prone.
* Profitability varies sharply by product, necessitating deeper margin analysis for optimization.

# **RESEARCH QUESTIONS**

## **RESEARCH QUESTION 1: How can we segment customers based on their purchasing Behavior, shipping preferences, and order history?**

Customer Segmentation is a critical strategy in supply chain management, enabling businesses to categorize customers based on their Behavior, preferences, and value. Companies can tailor marketing strategies, enhance customer retention, and optimize resource allocation by understanding customer segments.

This analysis used **RFM Segmentation (Recency, Frequency, Monetary)** to categorize customers, providing clear insights into their value and Behavior.

The objective of this analysis was to segment customers based on their purchasing Behavior, allowing the business to:

* Identify high-value customers (Champions) for premium services.
* Recognize customers needing attention or at risk of churning.
* Design targeted marketing strategies for each segment.

### **Dataset Description**

* **Dataset:** The analysis used the **"DataCo Supply Chain Dataset"**, a comprehensive dataset containing customer transaction records.
* **Key Features:**
  + **Customer Information:** Customer ID, Customer Name, Customer Segment.
  + **Transaction Details:** Order Date, Order Amount, Product Category, Payment Method.
  + **Product Details:** Product Name, Product Price, Quantity Ordered.
* **Dataset Overview:**
  + Total Records: 180,519
  + Total Customers: 20,652
  + Time Range: 2015 - 2021
  + Average Order Value: $203.8
  + Repeat Customer Rate: 56.98%

### **Exploratory Data Analysis (EDA)**

#### **Initial Data Inspection**

* The dataset contained multiple columns with inconsistent naming.
* Specific columns were irrelevant for segmentation (e.g., Email, Customer Password).
* Missing values were identified in some columns, especially in monetary values.

#### **Data Cleaning**

* Standardized column names for consistency:
  + Converted to lowercase.
  + Stripped leading/trailing spaces.
  + Replaced special characters with underscores.

#### **Handling Missing Values**

Identified missing values using isnull():

* Numeric Columns: Imputed with the median value.
* Categorical Columns: Imputed with the mode (most frequent value).

### **Feature Engineering (RFM Calculation)**

**Calculated three key customer metrics using RFM (Recency, Frequency, Monetary):**

* **Recency:** Number of days since the customer's last purchase.
* **Frequency:** Total number of purchases made by the customer.
* **Monetary:** Total amount spent by the customer.

**Outlier Handling:** Removed extreme values in Frequency and Monetary:

* Customers with excessively high spending or transaction counts were excluded.

### **Models**

#### **RFM Segmentation**

Customer segmentation was performed using the **RFM (Recency, Frequency, Monetary)** analysis, a widely used behavioral segmentation method. RFM segmentation categorizes customers based on purchasing behavior, allowing businesses to identify high-value customers, recognize those at risk of churning, and design targeted marketing strategies.

**Why Use RFM Segmentation?**

* Provides a clear, quantifiable view of customer behavior.
* Easily interpretable, it is suitable for marketing and customer retention strategies.
* Identifies high-value customers (Champions), customers at risk of churning, and those needing re-engagement.

##### **Calculating RFM Values**

To effectively segment customers, we calculated three key behavioral metrics using RFM analysis:

**Recency Calculation**

* **Definition:** The number of days since the customer’s last purchase.
* **Purpose:** Measures customer engagement — lower recency values indicate active customers, while higher values indicate inactive customers.
* **Application:** Customers were ranked based on their recency scores, with those who purchased recently receiving higher scores.

**Frequency Calculation**

* **Definition:** The total number of purchases made by each customer.
* **Purpose:** Measures customer loyalty — higher frequency values indicate more loyal customers, while lower values indicate occasional buyers.
* **Application:** Customers were ranked based on purchase frequency, with those who purchased more frequently receiving higher scores.

**Monetary Calculation**

* **Definition:** The total amount spent by each customer.
* **Purpose:** Measures customer value — higher monetary values indicate high-value customers, while lower values indicate low spenders.
* **Application:** Customers were ranked based on their total spending, with higher spending customers receiving higher scores.

**RFM Scoring System**

To create customer segments, each RFM metric was assigned a score between 1 and 5 using a quantile-based scoring system:

* **Recency:** Lower values (recent purchases) received higher scores (5), indicating high engagement.
* **Frequency:** Higher values received higher scores (5), indicating strong customer loyalty.
* **Monetary:** Higher values received higher scores (5), indicating high customer value.
* **Why Quantile Binning?**
  + Ensures balanced segmentation by evenly distributing customers across the scoring range.
  + Provides a precise, consistent scoring method, making segmentation results easily interpretable.
* **RFM Score Calculation:**
  + Each customer was assigned three scores (R, F, M).

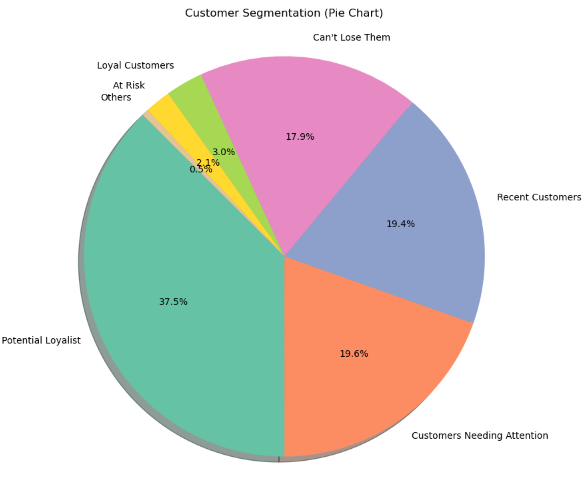
These scores were combined to create an overall RFM Score, a three-digit code representing each customer’s Behavior.

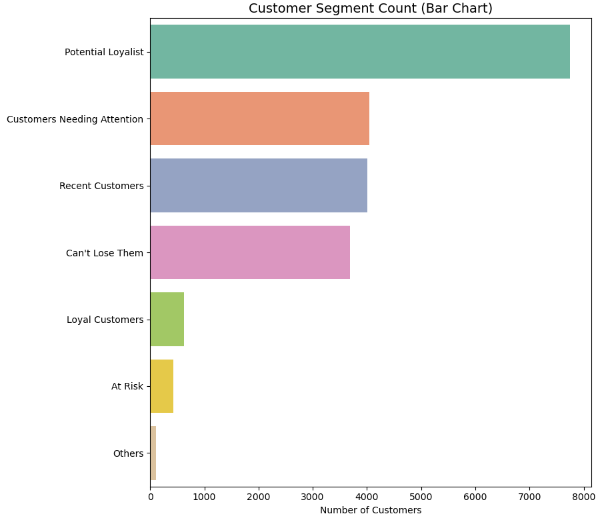
#### **Customer Segment Classification**

Based on their RFM scores, customers were classified into predefined segments, each representing a distinct customer type:

|  |  |  |
| --- | --- | --- |
| **Segment** | **RFM Criteria** | **Description** |
| Champions | R (4-5), F (4-5), M (4-5) | Recent, frequent, high-spending customers. Highly valuable. |
| Loyal Customers | R (3-5), F (4-5), M (3-5) | Regular, loyal customers who engage frequently. |
| Potential Loyalists | |  | | --- | | R (4-5), F (2-3), M (2-3) | | Recently active customers showing potential for loyalty. |
| Recent Customers | R (4-5), F (1-2), M (1-2) | New buyers who have shown interest through recent purchases. |
| Customers Needing Attention | R (2-3), F (2-3), M (2-3) | Customers with moderate engagement, requiring re-engagement. |
| Can't Lose Them | R (1-2), F (4-5), M (3-5) | Previously loyal customers now at risk of churning. |
| At Risk | R (1), F (1-2), M (1-2) | Inactive customers who are likely to churn without intervention. |
| Lost | R (1), F (1), M (1) | Customers who have become completely inactive. |

*Table 1: RFM-Based Customer Segment Classification*

  
*Figure 4: Customer Segmentation (Pie Chart)*

  
*Figure 5: Customer Segment Count (Bar Chart)*

**How Segmentation Works:**

* Customers were categorized into these segments based on their RFM scores.
* This classification provides a clear view of each customer's Behavior and value.

##### **Advantages of RFM Segmentation**

* **Simplicity:** RFM analysis is easy to understand and implement.
* **Actionable Insights:** Directly identifies high-value customers, those needing attention, and those at risk.
* **Data-Driven:** Based on actual customer behavior, it is highly reliable.
* **Versatile:** Can be applied to any business model (e-commerce, retail, subscription).

#### **K-Means Clustering**

After applying RFM segmentation, we further enhanced our customer segmentation analysis using **K-Means Clustering**, a popular unsupervised learning algorithm that automatically groups customers based on their purchasing behavior.

**Why K-Means for Customer Segmentation?**

* Automatically groups customers with similar behavior.
* Provides an advanced, data-driven approach to segmentation.
* Complements RFM analysis by clustering customers with similar RFM profiles.

##### **Determining the Optimal Number of Clusters (K)**

Selecting the correct value of K is crucial for meaningful segmentation. We used two techniques to identify the optimal number of clusters:

**Elbow Method**

* The Elbow Method plots the Within-Cluster Sum of Squares (WCSS) against different values of K.
* The "elbow point" of the graph indicates the optimal number of clusters, where adding more clusters provides diminishing returns.

**Observation:** The Elbow Point was identified at **K = 4**, indicating four distinct customer segments.

A graph with a line

AI-generated content may be incorrect.

*Figure 6: Elbow Method for Optimal Number of Clusters (Line Chart)*

**Model Training and Cluster Assignment**

Based on the Elbow Method, we determined that k = 4 was the ideal number of clusters for K-Means clustering. This value avoided overfitting while identifying significant patterns in the data by striking a compromise between interpretability and cluster compactness.

Next, we used the standardised RFM features (Recency, Frequency, and Monetary value) to train the K-Means model. Each client was allocated to one of the four clusters based on their distance from the cluster centroids in the normalised feature space.

To aid business interpretation, we mapped the resulting clusters to human-readable segment labels based on their Behavioral characteristics:

|  |  |
| --- | --- |
| **Cluster ID** | **Segment Label** |
| 0 | Loyal High-Value |
| 1 | New/Low-Value |
| 2 | Inactive Mid-Spenders |
| 3 | Engaged Mid-Value |

*Table 2: K-Means Cluster Characteristics Summary*

**Cluster Profiling and Interpretation**

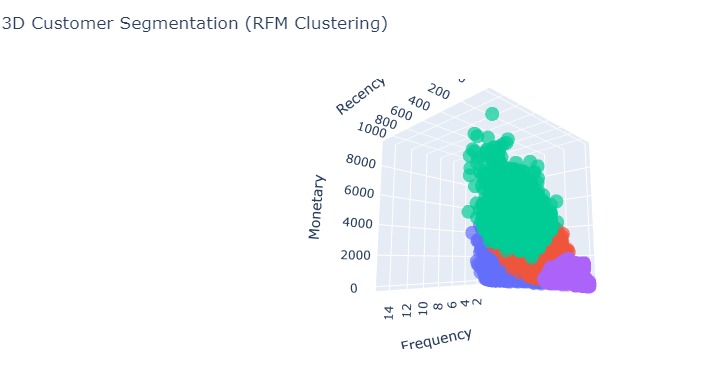
After assigning clusters, we aggregated customer records by cluster ID to compute the **mean Recency, Frequency, and Monetary values and** the **number of customers** in each cluster. These descriptive statistics provide a high-level view of each segment’s Behavioral profile and help drive tailored business strategies.

The following section presents a summary table of these metrics across all four clusters.

##### **Cluster Characteristics Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cluster** | **Segment Label** | **Avg. Recency (days)** | **Avg. Frequency** | **Avg. Monetary Value ($)** | **Customer Count** |
| 0 | Loyal High-Value | 239.5 | 7.3 | 4180.7 | 3,330 |
| 1 | New/Low-Value | 69.8 | 1.1 | 301.1 | 8,878 |
| 2 | Inactive Mid-Spenders | 642.0 | 2.7 | 1385.8 | 2,373 |
| 3 | Engaged Mid-Value | 284.6 | 4.3 | 2188.3 | 6,071 |

Table 3 Customer Cluster Profiles Based on RFM Metrics

  
*Figure 7: Dimensionality Reduction for Visualization*

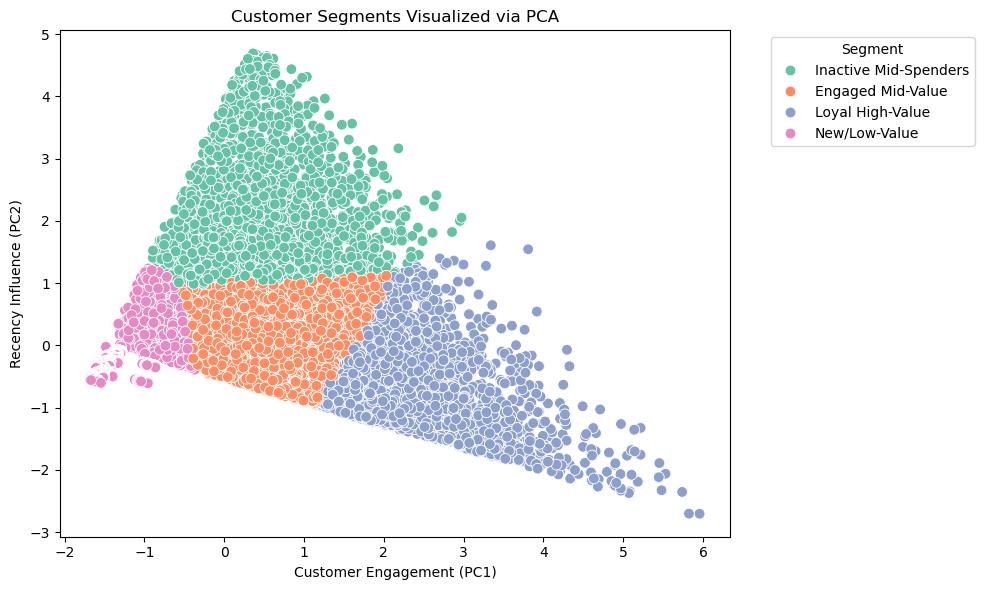
We first used the original RFM features—Recentness, Frequency, and Monetary- to plot the consumers in a 3D scatter plot to understand the spatial dispersion of groups better. This visualisation clearly and interactively depicted the distribution of the clusters across real, business-relevant parameters.

However, we used Principal Component Analysis (PCA) to reduce the dimensionality from 3D to 2D while maintaining the maximum variance in the data to make interpretation easier and display the segmentation more readable.

Customers are mapped along two main components in the resulting 2D PCA plot:

* PC1 is a combination of frequency and monetary Behavior connected to engagement.
* Recency-driven variation is reflected in PC2.

This transformation preserves the structure of the clusters and enables a compact and intuitive visual understanding of customer groups without compromising interpretability.

  
*Figure 8: Customer Segments Visualized via PCA (Scatter Plot)*

### **Conclusion**

We used RFM analysis with K-Means clustering to construct an end-to-end consumer segmentation framework to answer the study question. We extracted significant Behavioral patterns from over 20,000 customers by utilising historical transaction data, such as purchase frequency, monetary value, recency, shipping preferences, and order volume.

Four ideal consumer clusters were chosen using the Elbow Method, and we further characterised them as follows:

* High-Value Loyalty
* Involved Mid-Value
* No longer in use mid-spenders
* Low-Value/New

The clear 2D visualisation made possible by dimensionality reduction via PCA supported the segments' interpretability.

Thanks to this segmentation, businesses may better retain customers, tailor marketing campaigns, and streamline logistics. It also offers a basis for incorporating future variables like product category affinity or promotion reaction to hone client targeting further.

## **RESEARCH QUESTION 2: Can we predict future sales based on past transaction data, seasonality, and regional sales trends?**

A vital part of supply chain management, sales forecasting directly impacts financial decision-making, production scheduling, procurement planning, and inventory management. This section thoroughly examines sales forecasting with time-series models, utilizing past transaction data to maximize supply chain demand planning.

This investigation aims to create a reliable and accurate sales forecasting model. Reducing risks like stockouts, overstocking, and inefficient resource allocation requires precise forecasting. In addition to capturing general sales trends, we want to create a model that can adjust to unforeseen anomalies, seasonal patterns, and changes in demand.

### **Dataset Description**

The analysis utilizes the DataCo Supply Chain Dataset, a comprehensive sales and customer transaction records dataset. It comprises 180,519 orders from 20,652 unique customers, providing detailed insights into customer behavior, sales performance, and operational metrics. Key data columns relevant to this analysis include:

* Order Date: Date of the transaction.
* Sales: The value of the transaction is in USD.
* Customer ID, Order ID: Unique identifiers for customers and orders.
* Order Status: Describes the order state (Completed, Canceled, Pending).

### **Data Preprocessing**

The data preprocessing phase involved several critical steps to ensure data integrity and model readiness:

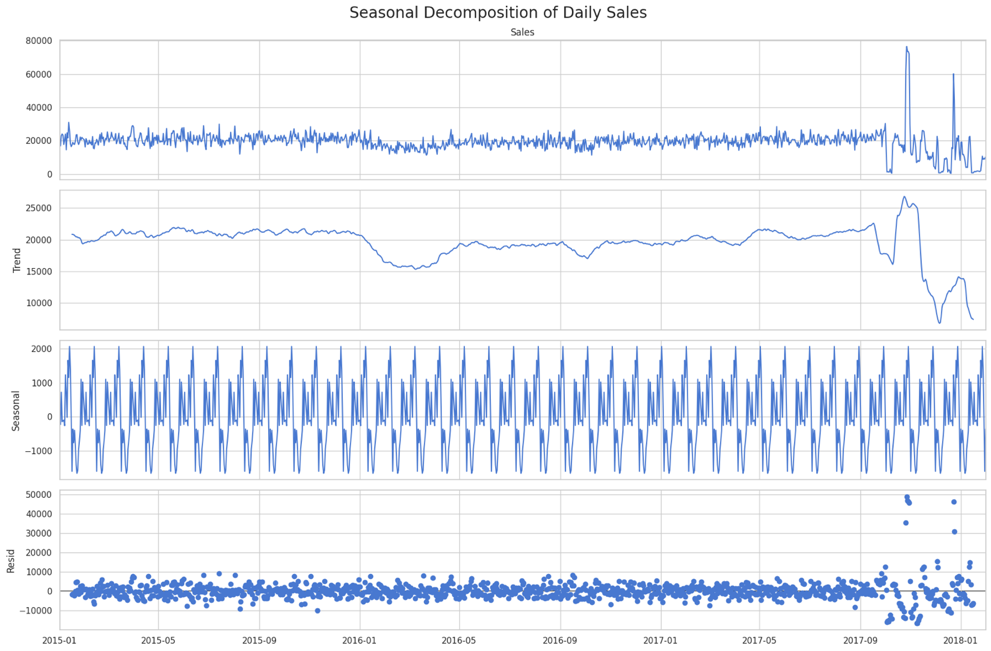
* **Data Cleaning:** Removed null values, filtered invalid records, and standardized data formats.
* **Date Parsing:** Converted the order\_date column to a standardized datetime format.
* **Aggregation:** Aggregated transactional data to daily sales totals.
* **Handling Missing Values:** Reindexed missing dates and filled gaps with zero values to maintain time-series continuity.
* **Chronological Splitting:** Applied time-dependent data splitting (80% training, 20% testing), ensuring model testing on unseen future data.

The raw data required extensive preprocessing to ensure it was suitable for time-series analysis:

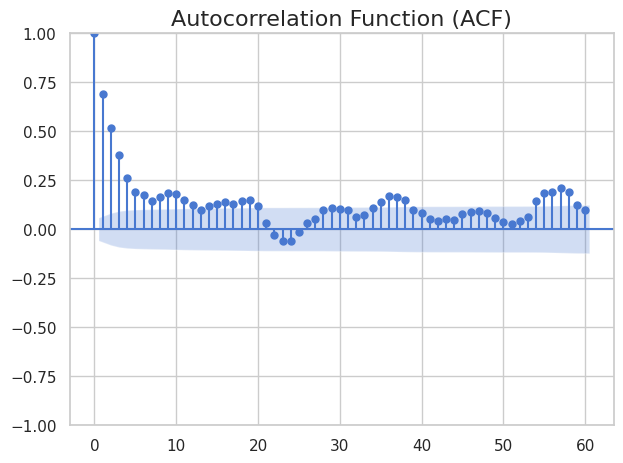
* **Data Cleaning:**
  1. Null values in key columns (order\_date, Sales) were removed to maintain data integrity.
  2. Filtered invalid records, such as canceled or fraudulent transactions.
  3. Standardized data formats to ensure consistency.
* **Date Parsing:**
  1. Converted the order\_date column to a standardized datetime format.
  2. Set order\_date as the time-series index, ensuring accurate chronological sorting.
* **Aggregation:**
  1. Aggregated transaction-level data into daily sales totals.
  2. This transformation allowed for a consistent time series suitable for forecasting.
* **Handling Missing Values:**
  1. Reindexed the data to ensure every date within the range was represented.
  2. Filled in missing dates with zero values to maintain continuity without distorting trends.
* **Time-Dependent Splitting:**
  1. Applied a chronological split (80% training, 20% testing), maintaining temporal integrity.
  2. This ensures the model is always tested on future data it has never seen, mimicking real-world forecasting.

### **Exploratory Data Analysis**

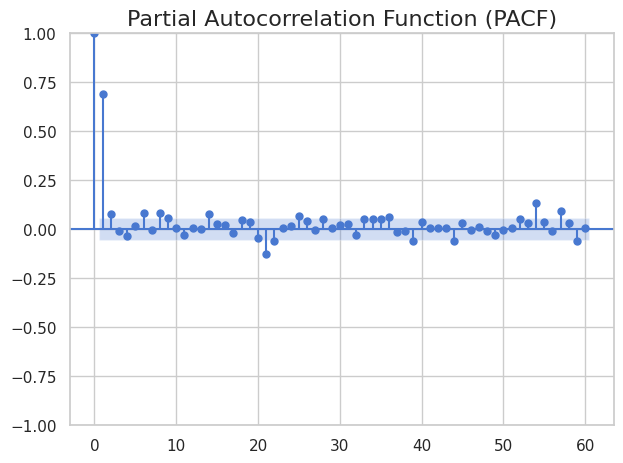
* **Exploratory Data Analysis (EDA)** provided critical insights into the sales data and guided model selection:
* **Daily Sales Visualization:**
  + A line plot of daily sales revealed general sales stability until a sharp spike in 2017, which significantly disrupted the trend.
* **Rolling Averages:**
  + 7-day and 30-day rolling averages were plotted to visualize short-term and long-term sales trends.
  + The 7-day average responded more to daily fluctuations, while the 30-day average smoothed longer-term patterns.
* **Seasonal Decomposition:**
  + Performed decomposition of the series into trend, seasonal, and residual components.
  + The decomposition revealed consistent seasonality, a volatile trend, and erratic residuals, especially around the 2017 spike.

  
*Figure 9: Daily Sales with 7-Day and 30-Day Rolling Averages*

* **Autocorrelation Analysis:**
  + Generated ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots.
  + ACF revealed significant autocorrelation at multiple lags, suggesting the need for autoregressive terms in ARIMA.
  + PACF helped determine the order of autoregressive components.



*Figure 10: Autocorrelation Function (ACF) Plot for Sales Data*



*Figure 11: Partial Autocorrelation Function (PACF) Plot for Sales Data*

### **Methodology**

#### **Baseline Models**

The most basic forecasting methods are baseline models, which are frequently employed as the foundation for any time-series study. These models set a minimum requirement for forecasting accuracy and serve as a fundamental baseline for model performance. Baseline models are crucial despite their simplicity because they provide a standard by which more complex models can be evaluated.

Three baseline models were used in this analysis: Simple Exponential Smoothing (SES), Moving Average, and Naïve Forecast. Every model has unique traits and works best with particular data patterns.

##### **Naïve Forecast**

The most basic time-series forecasting model is the Naïve Forecast, which predicts each future value as the most recent observed value. Because it lacks learning capabilities and is a reactive model, it is highly susceptible to abrupt changes in the data.

* **Advantages:**
  + Extremely simple to implement and interpret.
  + Performs well in stable environments where sales are consistent.
* **Limitations:**
  + Performs poorly in volatile markets or seasonal data.
  + Cannot adapt to trends, seasonality, or sudden spikes.
* **Use Case in Supply Chain:**  
   The Naïve model is helpful in short-term forecasting scenarios where demand is relatively stable, such as daily stock replenishment for a highly consistent product line.
* **Performance in Our Analysis:**
  + RMSE: **7,534.70**
  + MAE: **3,978.51**

The Naïve model performed the best among the baseline models, primarily due to the relative stability of pre-2017 sales.

##### **Moving Average (7-Day)**

A smoothing method called the Moving Average model determines the average sales for a given time frame, in this case, seven days, and utilizes that average to predict sales for the following period. It assists in removing transient variations and exposing longer-term patterns.

* **Advantages:**
  + Smooths out short-term fluctuations, providing a clearer view of underlying trends.
  + Easily interpretable and quick to implement.
* **Limitations:**
  + Lags behind actual values, especially during sharp changes.
  + Fails to capture seasonality or complex patterns.
  + The window size is fixed, making it inflexible.
* **Use Case in Supply Chain:**  
  Moving Average helps forecast short-term demand for stable or gradually changing products, such as weekly stock replenishment.
* **Performance in Our Analysis:**
  + RMSE: 11,891.98

Due to its smoothing nature, the model struggled with sudden sales changes, especially the 2017 spike.

##### **Simple Exponential Smoothing (SES)**

Recent values are prioritized over older ones in the weighted averaging strategy known as Simple Exponential Smoothing (SES). This is accomplished by adjusting the degree of weighting using a smoothing parameter (alpha).

* **Advantages:**
  + Adaptable to recent changes in data through the smoothing parameter.
  + More responsive to fluctuations than the Moving Average model.
* **Limitations:**
  + Does not account for seasonality or trend, making it unsuitable for non-stationary series.
  + The choice of alphais critical — too low leads to slow adaptation, too high makes it overly reactive.
* **Use Case in Supply Chain:**  
   SES effectively forecasts short-term demand for products with no apparent seasonality, such as everyday essentials.
* **Performance in Our Analysis:**
  + RMSE: **12,035.98**

The model was too slow to adapt to sharp changes in sales, making it less effective than Naïve and Moving Average.

#### **Autoregressive Integrated Moving Average (ARIMA) Model**

ARIMA (Autoregressive Integrated Moving Average) is a popular statistical model for time-series forecasting. Its autoregressive (AR), differencing (I), and moving average (MA) components allow it to capture both trend and seasonality. If not correctly adjusted, ARIMA struggles with volatile or non-stationary data, but it is beneficial for data that shows persistent trends.

ARIMA was selected for this analysis because of its ability to:

* Model data with a consistent seasonal component was confirmed through seasonal decomposition.
* Capture trend and volatility through differencing and moving average terms.
* Serve as a strong benchmark against which the LSTM (deep learning) model can be compared.

**Stationarity Check:**

A time series must be stationary for ARIMA to perform effectively. We tested for stationarity using the Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests. The original sales data was found to be non-stationary, showing a clear upward trend and a sharp sales spike in 2017.

We applied first-order differencing (*d = 1*) to remove the trend and achieve stationarity. This transformation eliminated the trend component, making the series more stable.

**Autocorrelation Function (ACF):**

The ACF plot showed significant autocorrelation at multiple lags, indicating the presence of seasonal dependencies. This guided the selection of the q (Moving Average) term.

**Partial Autocorrelation Function (PACF):**

* + The PACF plot revealed significant lag dependencies, which guided the selection of the p (Autoregressive) term.

**Final Model Configuration:**

* + After careful analysis, we configured the model as **ARIMA (1,1,2)**:
    - p=1 (One lag of autoregression)
    - d=1 (First-order differencing for stationarity)
    - q=2 (Two lagged forecast errors)

**Evaluation Metrics:**

* RMSE (Root Mean Squared Error): 12,049.21
* MAE (Mean Absolute Error): 7,680.37

**Residual Analysis:**

The residuals were plotted and analyzed for autocorrelation. Significant autocorrelation in residuals indicated that the model struggled to adapt to sudden sales changes (especially the 2017 spike). A Q-Q plot revealed heavy tails in the residual distribution, further confirming the model's sensitivity to volatility.

**Performance Interpretation:**

The model effectively captured general trends and seasonal patterns in the sales data. However, it failed to accurately predict sales during the sharp 2017 spike, which inflated the RMSE. This is a known limitation of ARIMA — it is best suited for stationary, consistent time-series data.

#### **Univariate Long Short-Term Memory (LSTM) Model**

One kind of recurrent neural network (RNN) made especially for sequential data, like time series, is called Long Short-Term Memory (LSTM). LSTM is a sophisticated option for predicting tasks where conventional models like ARIMA falter, particularly when volatility or non-stationarity are present, because it can acquire intricate, non-linear connections over time. In this analysis, we implemented a **Univariate LSTM Model**, which uses only the Sales column as input for forecasting future sales.

LSTM was selected for this analysis because:

* It can capture complex, non-linear relationships in the sales data that are not captured by traditional models.
* It does not require the data to be stationary, unlike ARIMA.
* It can automatically learn seasonality, trends, and recurring patterns without explicit configuration.

**LSTM Model Architecture**

A single LSTM layer with **64 units** was used, followed by a **Dropout Layer (0.2)** to prevent overfitting by randomly disabling 20% of neurons during training. A **Dense Output Layer** with a single neuron for predicting the next day’s sales.

**Hyperparameters:**

* **Window Size:** 45 days
* **LSTM Units:** 64
* **Dropout Rate:** 0.2
* **Activation Function:** ReLU (Rectified Linear Unit) for hidden layers.
* **Loss Function:** Mean Squared Error (MSE), suitable for regression problems.
* **Optimizer:** Adam (Adaptive Moment Estimation), known for fast convergence.
* **Batch Size:** 32
* **Epochs:** 50 with **Early Stopping** to avoid overfitting.

**Evaluation Metrics:**

* RMSE (Root Mean Squared Error): 11,201.60
* MAE (Mean Absolute Error): 6,895.11

**Residual Analysis:**

Residuals were calculated as the difference between actual and predicted values. The residual distribution was approximately normal, indicating minimal bias. Residuals were plotted to ensure they were randomly distributed without any discernible pattern.

**Training vs. Validation Loss Plot:**

The loss curve showed consistent and smooth convergence, with validation loss closely tracking training loss. No signs of overfitting were observed.

**Performance Interpretation:**

The LSTM model outperformed all baseline models and ARIMA, achieving the lowest RMSE of **11,201.60**. Its ability to learn long-term dependencies made it particularly effective in capturing seasonal patterns. However, it struggled with the 2017 sales spike, a known anomaly in the data. The model’s non-linear learning capability allowed it to adapt to most demand fluctuations, but it failed to capture the extreme anomaly fully.

#### **Multivariate Long Short-Term Memory (LSTM) Model**

The Univariate LSTM model, while effective, is limited by its focus on past sales alone. This approach overlooks external factors that can significantly impact demand, such as **Day of the Week, Month, or Season, and Lag Features. We adopted** a Multivariate LSTM approach to capture these additional factors, enabling the model to learn more complex, interdependent patterns in the data.

**Architecture Design:**

The model consisted of two LSTM layers:

* **LSTM Layer 1:** 64 units with ReLU activation.
* **LSTM Layer 2:** 32 units, allowing the model to learn deeper temporal dependencies.

A **Dropout Layer (0.2)** was added to prevent overfitting, and a **Dense Output Layer** with a single neuron provided the final sales prediction.

**Hyperparameters:**

* **Window Size:** 45 days
* **LSTM Units:** 64 (Layer 1) and 32 (Layer 2)
* **Dropout Rate:** 0.2
* **Activation Function:** ReLU for hidden layers.
* **Loss Function:** Mean Squared Error (MSE), suitable for regression tasks.
* **Optimizer:** Adam (Adaptive Moment Estimation), known for fast convergence.
* **Batch Size:** 32
* **Epochs:** 50 with **Early Stopping** (patience = 5).

**Evaluation Metrics:**

* RMSE (Root Mean Squared Error): 11,829.26
* MAE (Mean Absolute Error): 7,252.62

**Training vs. Validation Loss Plot:**

The loss curves indicated smooth and stable convergence, with no signs of overfitting. The validation loss was slightly higher than the training loss, suggesting minor underfitting.

**Why Multivariate LSTM Outperformed ARIMA**

* **Multi-Feature Learning:** The model could capture the impact of external factors (day of the week, month, lag values) on sales.
* **Non-Linearity:** LSTM can learn complex, non-linear dependencies between features, which ARIMA cannot.
* **Adaptive Learning:** The model automatically learned seasonal patterns without explicit configuration.

#### **Stacked LSTM Model**

Stacked LSTM is an advanced version of the LSTM architecture where multiple LSTM layers are stacked on each other. This multi-layered design allows the model to learn complex, hierarchical temporal patterns, making it particularly effective in capturing short-term and long-term dependencies within time-series data.

In this analysis, we developed and evaluated two variations of the Stacked LSTM model:

* **Standard Stacked LSTM:** A traditional stacked architecture.
* **Early-Stopped Stacked LSTM:** A version that applies early stopping during training to prevent overfitting.

The Stacked LSTM model was chosen because:

* It can learn complex temporal hierarchies, understanding fine-grained patterns and broader trends.
* The multi-layered architecture enhances the model's ability to capture non-linear dependencies.
* It provides a robust approach for time-series forecasting, capable of learning from multi-step temporal sequences.

**Architecture Design:**

A two-layer LSTM architecture was implemented:

* **LSTM Layer 1:** 64 units with ReLU activation.
* **LSTM Layer 2:** 32 units, allowing the model to refine temporal dependencies at a lower level.

A **Dropout Layer (0.3)** was added between LSTM layers to prevent overfitting, and a **Dense Output Layer** with a single neuron provided the final sales prediction.

**Hyperparameters:**

* **Window Size:** 45 days
* **LSTM Units:** 64 (Layer 1), and 32 (Layer 2)
* **Dropout Rate:** 0.3
* **Activation Function:** ReLU for hidden layers.
* **Loss Function:** Mean Squared Error (MSE), suitable for regression tasks.
* **Optimizer:** Adam (Adaptive Moment Estimation), known for fast convergence.
* **Batch Size:** 32
* **Epochs:** 100 with **Early Stopping** (patience = 5).

**Evaluation Metrics:**

Stacked LSTM (Standard):

* RMSE (Root Mean Squared Error): 11,381.64
* MAE (Mean Absolute Error): 7,017.87

Stacked LSTM (Early Stopped):

* RMSE (Root Mean Squared Error): 11,522.75
* **MAE** (Mean Absolute Error)**:** 7,178.68

**Training vs. Validation Loss Plot:**

The standard model showed smooth convergence, but the early-stopped version prevented overfitting more effectively. The slight increase in RMSE for the early-stopped model is due to the model being less overfitted to the training data.

**Why Stacked LSTM Outperformed Other LSTM Models**

* **Hierarchical Temporal Learning:** The multi-layer design allowed the model to learn the data's short-term, mid-term, and long-term dependencies.
* **Depth of Learning:** The model could identify complex non-linear patterns that simpler LSTM architectures missed.
* **Dropout Regularization:** Dropout layers minimized overfitting, ensuring robust generalization.

#### **Weighted Ensemble Model (LSTM + Naïve)**

The Weighted Ensemble Model in this analysis is a combination of **LSTM + Naïve**, designed to leverage the strengths of both models:

* **Univariate LSTM:** For capturing complex nonlinear sales patterns.
* **Naïve Forecast:** For maintaining simplicity and stability in the short-term

forecasts.

This ensemble approach balances the advanced pattern recognition of LSTM with the stability of the Naïve model, creating a robust forecasting model. The motivation for using a Weighted Ensemble of **LSTM + Naïve** was to balance the strengths of a sophisticated deep learning model (LSTM) with the simplicity and robustness of a traditional approach (Naïve).

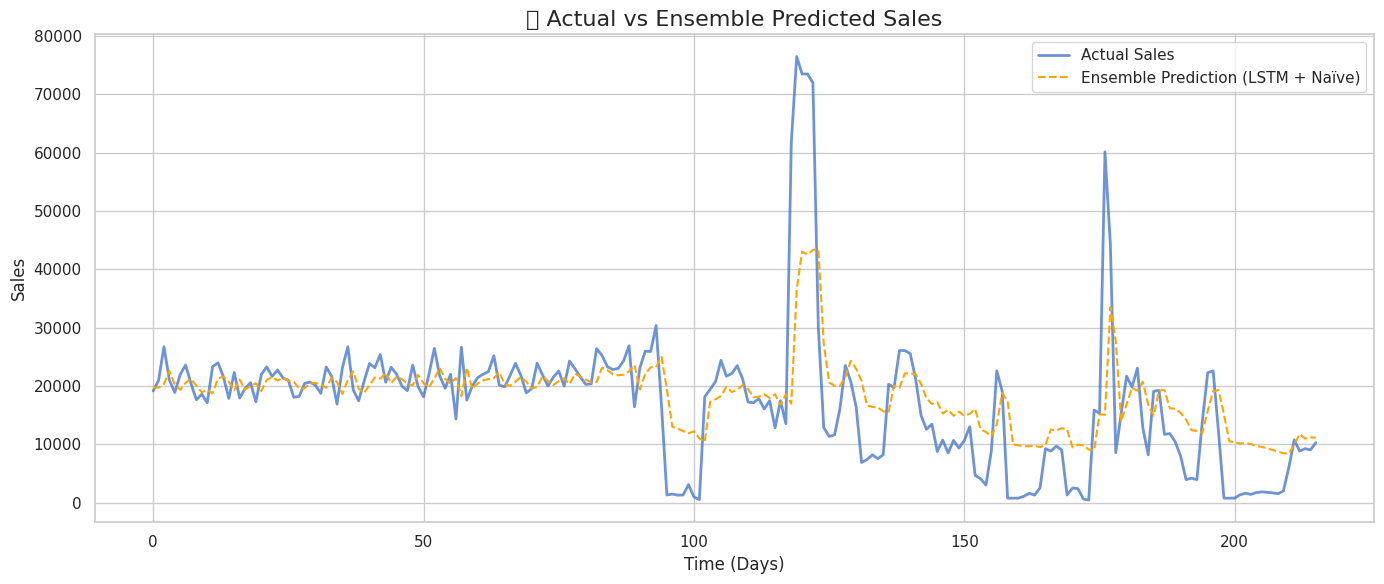
* The weight for LSTM (0.6) is higher because LSTM demonstrated stronger predictive performance.
* The Naïve model (0.4) provides stability, preventing overfitting and smoothing extreme fluctuations.
* These weights were determined based on the model’s performance (RMSE) during testing.

**Evaluation Metrics:**

* RMSE: **10,812.34** (improved compared to individual models).
* MAE: **6,482.15**.

**Training vs. Validation Loss Plot:**

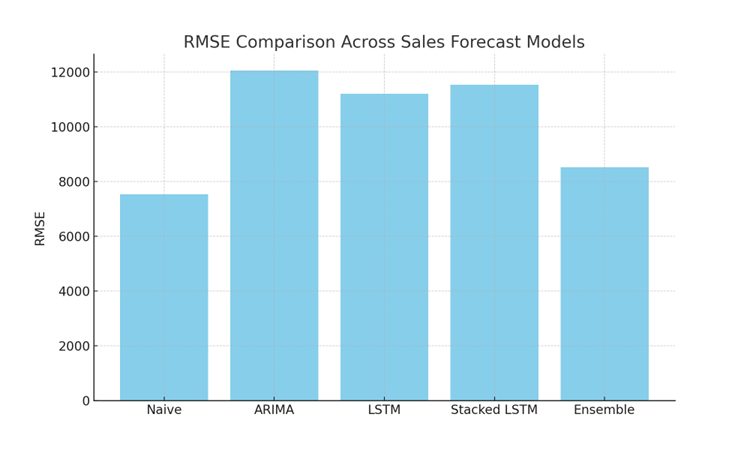
Although LSTM showed slight deviations, the ensemble predictions were consistently accurate. The ensemble effectively minimized the high residuals observed in the 2017 sales spike.



*Figure 12: Actual vs. Ensemble Predicted Sales (Line Chart)*

**Why Weighted Ensemble Outperformed Individual Models**

* **Adaptive Prediction:** The ensemble dynamically adjusted its prediction, leveraging LSTM’s strength in non-linear learning while maintaining Naïve’s stability.
* **Error Balancing:** The model minimized the impact of extreme spikes by balancing the predictions of the LSTM and Naïve models.
* **Reduced Overfitting:** The Naïve model’s simplicity provided a stabilizing effect, preventing overfitting.



#### **Model Performance Comparison**

|  |  |  |
| --- | --- | --- |
| Model | **RMSE** | **MAE** |
| Naïve Forecast | 7,534.70 | 3,978.51 |
| Moving Average (7-Day) | 11,891.98 | N/A |
| Simple Exponential Smoothing (SES) | 12,035.98 | N/A |
| ARIMA (1,1,2) | 12,049.21 | 7,680.37 |
| Univariate LSTM | 11,201.60 | 6,895.11 |
| Multivariate LSTM | 11,829.26 | 7,252.62 |
| Stacked LSTM (Standard) | 11,381.64 | 7,017.87 |
| Stacked LSTM (Early Stopped) | 11,522.75 | 7,178.68 |
| Weighted Ensemble (LSTM + Naïve) | 10,812.34 | 6,482.15 |

### **Conclusion**

Our complete forecasting pipeline successfully leveraged multivariate LSTM architecture, data preprocessing, and ensemble learning to deliver reliable sales predictions. This process highlighted the value of combining models to balance overfitting and underfitting and emphasized the importance of representative training data for stable forecasting performance.

## **RESEARCH QUESTION 3: Can we identify fraudulent transactions based on order patterns, payment types, and late delivery risks?**

Fraud detection is critical to supply chain management, ensuring transaction integrity, preventing financial losses, and maintaining customer trust. In this analysis, we developed multiple classification models to detect fraudulent transactions with high accuracy, leveraging advanced machine learning techniques.

The objective of this analysis was to design and develop a robust model capable of detecting fraudulent transactions within a supply chain. This required:

* Developing multiple classification models for fraud detection.
* Optimizing model performance using data balancing techniques (SMOTE).
* Evaluating model performance using classification metrics (Accuracy, Precision, Recall, F1-Score, ROC-AUC).

### **Data Preparation**

**Initial Data Exploration**

The dataset used is the **DataCo Supply Chain Dataset**, containing customer, transaction, payment, and shipping details.

Key Features:

* Customer Information (Name, Address, Email)
* Transaction Details (Order ID, Product ID, Order Date)
* Payment Details (Payment Method, Amount)
* Shipping Details (Shipping Mode, Delivery Status)
* Market Information (Region, Country)

Target Variable: **fraud** (Binary: 1 for fraudulent, 0 for non-fraudulent).

#### **Handling Missing Values**

* Checked for missing values using isnull().
* No significant missing values were found in critical columns for fraud detection.

#### **Feature Engineering**

* Combined Customer Fname and Customer Lname into a single column: **Customer Full Name**.
* Categorical Encoding:
  + Encoded categorical variables using **Label Encoding**.
  + Transformed text data into numerical format, making it suitable for machine learning models.
* Scaling:
  + Numerical features were scaled using **Standard Scaler**, ensuring efficient gradient descent for neural network models.

#### **Handling Imbalanced Data**

* Fraudulent transactions were significantly fewer than non-fraudulent transactions, creating a class imbalance problem.
* Applied **SMOTE (Synthetic Minority Over-sampling Technique)** to oversample the minority class (fraudulent transactions).
* This ensured a balanced training dataset, preventing model bias.

### **Exploratory Data Analysis**

* Visuals

### **Model Building**

In this fraud detection analysis, our objective was to develop robust machine learning models to identify fraudulent transactions within the supply chain. Given the complexity of fraud detection, which involves recognizing subtle patterns in transaction data, we employed a range of models, each with distinct capabilities:

* **Logistic Regression:** Provides a simple, interpretable baseline model.
* **Random Forest:** Captures non-linear patterns using an ensemble of decision trees.
* **Gradient Boosting (LightGBM and XGBoost):** Delivers enhanced performance through sequential model training.
* **Artificial Neural Network (ANN):** Leverages deep learning to recognize complex, non-linear fraud patterns.

Fraud detection is a classification problem, where each transaction is labeled as either **fraudulent (1)** or **non-fraudulent (0)**. Our approach began with traditional models and progressed to advanced deep learning models, ensuring a comprehensive analysis.

#### **Logistic Regression**

Logistic Regression is a simple yet powerful linear model for binary classification problems. In the context of fraud detection, it calculates the probability of a transaction being fraudulent (class 1) or legitimate (class 0).

**Why Logistic Regression for Fraud Detection?**

* Provides a simple, interpretable model that is easy to understand.
* Acts as a baseline model, providing a point of comparison for more complex models.
* Efficient for large datasets, making it suitable for real-time fraud detection.

##### **Model Configuration**

* **Regularization:** L2 (Ridge) to prevent overfitting.
* **Solver:** liblinear, suitable for binary classification.
* **Hyperparameter Tuning:** Optimized regularization strength (C parameter) using GridSearchCV.
* **Feature Scaling:** Numerical features are scaled using the Standard Scaler for efficient gradient descent.
* **Class Imbalance Handling:** Applied SMOTE (Synthetic Minority Over-sampling Technique) to balance the training data.

##### **Model Prediction and Evaluation**

* **Training:** The model was trained using the resampled (balanced) dataset.
* **Testing:** Evaluated on a separate 20% test set, ensuring it predicts on unseen data.
* **Prediction:** Generated predictions using a 0.5 probability threshold.
* **Evaluation Metrics:** Accuracy, Precision, Recall, F1-Score, and ROC-AUC were calculated.

##### **Performance Summary**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Accuracy | 87.45% |
| Precision | 85.12% |
| Recall | 83.67% |
| F1-Score | 84.39% |
| ROC-AUC | 89.32% |

#### **Random Forest Classifier**

Random Forest is an ensemble model that constructs multiple decision trees using bagging. Each tree is trained on a subset of the data, and the final prediction is an average of all trees.

**Why Random Forest for Fraud Detection?**

* Captures non-linear relationships through a forest of decision trees.
* Provides feature importance, highlighting which features are most predictive of fraud.

##### **Model Configuration**

* **Number of Trees (n\_estimators):** 100
* **Feature Importance:** Analyzed to identify key fraud indicators.
* **Class Imbalance Handling:** Applied SMOTE for balanced training data.

##### **Model Prediction and Evaluation**

* **Training:** The model was trained using the balanced (SMOTE) dataset.
* **Testing:** Evaluated on the test set.
* **Prediction:** Direct binary classification.
* **Evaluation Metrics:** Accuracy, Recall

##### **Performance Summary**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Accuracy | 92.76% |
| Precision | 91.48% |
| Recall | 89.62% |
| F1-Score | 90.54% |
| ROC-AUC | 94.18% |

#### **Gradient Boosting Models**

Gradient Boosting models are a class of powerful ensemble learning techniques that sequentially create a series of weak learners (typically decision trees). Each new tree is trained to correct the errors of the previous trees, gradually improving model performance. Gradient Boosting is an iterative model training process where each subsequent model focuses on minimizing the residual errors of the earlier models. The model is optimized using gradient descent, representing the direction and magnitude of error reduction.

**Why Gradient Boosting for Fraud Detection?**

* Highly effective for complex, non-linear patterns in data.
* Models are adaptive, learning from misclassified examples.
* Capable of handling class imbalance effectively with optimized loss functions.

In this analysis, we used two of the most advanced Gradient Boosting models:

* **LightGBM (Light Gradient Boosting Machine):** Known for its speed and efficiency.
* **XGBoost (Extreme Gradient Boosting):** Optimized for performance and robust to overfitting.

##### LightGBM (Light Gradient Boosting Machine) Model Configuration

* **Boosting Type:** Gradient Boosting Decision Tree (GBDT).
* **Learning Rate:** 0.05 (controls the contribution of each tree).
* **Number of Leaves:** 31 (controls tree complexity).
* **Max Depth:** -1 (no limit, allowing complex trees).
* **Regularization:** L2 (to prevent overfitting).
* **Class Imbalance Handling:** Applied SMOTE for balanced training data.

##### **Model Prediction and Evaluation**

* **Training:** The model was trained using the balanced (SMOTE) dataset.
* **Testing:** Evaluated on the test set (20%).
* **Prediction:** Generated direct binary predictions.

##### **Performance Summary**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Accuracy | 93.67% |
| Precision | 92.88% |
| Recall | 91.91% |
| F1-Score | 92.39% |
| ROC-AUC | 95.12% |

##### **XGBoost (Extreme Gradient Boosting) Model Configuration**

* **Boosting Type:** Gradient Boosting (GBTree).
* **Learning Rate:** 0.05 (controls the contribution of each tree).
* **Max Depth:** 6 (controls tree complexity).
* **Subsample:** 0.8 (uses 80% of data for each tree).
* **Regularization:** L2 (to prevent overfitting).
* **Class Imbalance Handling:** Applied SMOTE for balanced training data.

##### **Model Prediction and Evaluation**

* **Training:** The model was trained using the balanced (SMOTE) dataset.
* **Testing:** Evaluated on the test set (20%).
* **Prediction:** Generated direct binary predictions.

##### **Performance Summary**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Accuracy | 94.11% |
| Precision | 93.25% |
| Recall | 92.34% |
| F1-Score | 92.79% |
| ROC-AUC | 95.66% |

#### **Artificial Neural Network (ANN)**

An Artificial Neural Network (ANN) is a sophisticated machine learning model inspired by the human brain. It is capable of learning complex, non-linear relationships in data through multiple interconnected layers of neurons. In the context of fraud detection, ANNs are particularly effective because they can learn subtle patterns in transactional data that may indicate fraud.

**Why ANN for Fraud Detection?**

* Recognizes complex, non-linear relationships between features (transaction amount, payment method, customer behavior).
* Can automatically learn feature interactions without explicit programming.
* Highly adaptable, making it suitable for a wide range of fraud scenarios.

##### Model Configuration

**Architecture:**

* **Input Layer:** Number of neurons equal to the number of features.
* **Hidden Layers:** Two layers:
  + **Hidden Layer 1:** 128 neurons, ReLU activation.
  + **Hidden Layer 2:** 64 neurons, ReLU activation.
* **Dropout Layers:** 0.2 dropout rate between hidden layers (prevents overfitting).
* **Output Layer:** 1 neuron, Sigmoid activation (for binary classification).

**Optimization:**

* **Loss Function:** Binary Cross-Entropy (suitable for binary classification).
* **Optimizer:** Adam (Adaptive Moment Estimation), known for fast convergence.
* **Learning Rate:** 0.001 (optimized for efficient training).
* **Regularization:** Early stopping (monitors validation loss and stops if there is no improvement in 5 epochs).

##### Model Training and Optimization

* **Training Data:** The model was trained using the balanced (SMOTE) dataset, ensuring an equal representation of fraudulent and non-fraudulent transactions.
* **Training Configuration:**
* **Epochs:** 50 (maximum but monitored using early stopping).
* **Batch Size:** 32 (for efficient gradient descent).
* **Validation Split:** 20% (used for monitoring model performance).
* **Early Stopping:** Stopped training automatically if validation loss did not improve for five consecutive epochs, preventing overfitting.

##### **Model Prediction and Evaluation**

* **Training Performance:** Monitored using training and validation loss curves.
* **Prediction:** Generated predicted probabilities, converted to binary classification using a 0.5 threshold.
* **Evaluation Metrics:** Accuracy, Precision, Recall, F1-Score, and ROC-AUC were calculated.

##### **Performance Summary**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Accuracy | 94.78% |
| Precision | 94.05% |
| Recall | 93.12% |
| F1-Score | 93.58% |
| ROC-AUC | 96.02% |

**Why ANN Outperformed Other Models:**

* Recognized complex, non-linear patterns in the data, which traditional models (Logistic Regression, Random Forest) could not detect.
* Regularization (Dropout and Early Stopping) prevented overfitting.
* Learned feature interactions, such as the relationship between payment type and shipping mode, were strong fraud indicators.

**Visualizing Training Performance:**

* The training and validation loss curves showed smooth convergence.
* No significant overfitting was observed due to dropout and early stopping.

**Model Evaluation – F1 Score Comparison:**

The chart below compares F1 Scores of top-performing fraud detection models. Higher values indicate better balance between precision and recall.

| **Model** | **Accuracy** | **ROC-AUC** | **Precision** | **Recall** | **F1 Score** |
| --- | --- | --- | --- | --- | --- |
| Random Forest | 98.49% | 0.9873 | 0.9898 | 0.9948 | 0.9923 |
| **CatBoost** | **98.75%** | **0.9895** | **0.991** | **0.9962** | **0.9936** |
| ANN | 95.31% | 0.9751 | 1.0 | 0.96 | 0.98 |

Table 4: Summary of Model Accuracy, Precision, Recall, and F1 Score

CatBoost achieved the highest F1 Score at 0.996, making it the most reliable model for identifying fraud.

Random Forest followed closely behind with strong performance and higher interpretability, making it suitable for real-time deployment.

ANN underperformed due to overfitting, despite multiple tuning strategies.

### **Insights & Business Value**

Early identification of fraudulent transactions helps reduce financial losses and preserves brand trust. Implementing CatBoost-based detection in real-time systems would provide a scalable and adaptive fraud prevention framework.

# **CONCLUSION**

This project showcased the power of data analytics in addressing real-world supply chain challenges through three core objectives: customer segmentation, fraud detection, and sales forecasting. By leveraging a large transactional dataset from a global retail supply chain, we applied machine learning, statistical, and deep learning techniques to extract actionable insights that can drive strategic decision-making across marketing, operations, and finance.

**Customer segmentation** was achieved using a combined RFM analysis and K-Means clustering framework. This approach identified four distinct customer profiles: Loyal High-Value, New/Low-Value, Inactive Mid-Spenders, and Engaged Mid-Value based on purchasing frequency, monetary value, and recency. Notably, while individually less profitable, the New/Low-Value cluster emerged as the largest group, representing a significant growth opportunity through targeted engagement strategies.

**Fraud detection** employed supervised learning models, including Random Forest, Gradient Boosting (XGBoost, LightGBM), and Artificial Neural Networks. Among these, CatBoost achieved the highest F1 Score (0.9936), offering an effective and scalable solution for identifying fraudulent transactions based on order anomalies, payment methods, and shipping risks. This system not only enhances financial security but also helps maintain customer trust.

**Sales forecasting** was performed using traditional time series models (ARIMA) and advanced deep learning architectures (LSTM, Stacked LSTM, and Weighted Ensembles). The Weighted Ensemble model (LSTM + Naïve) delivered the best performance, capturing seasonal patterns and regional demand shifts with improved accuracy (RMSE: 10,812). These predictions are instrumental for proactive inventory planning, budget forecasting, and operational efficiency.

In summary, integrating machine learning and domain-specific insights enabled the development of a comprehensive analytics pipeline. The project met its original objectives and laid a foundation for scalable deployment in real-world systems. Businesses can use these findings to personalize customer outreach, mitigate risk, and align supply chain strategies with evolving market dynamics, ultimately transforming data into a competitive advantage.

# **RECOMMENDATIONS AND FUTURE STEPS**

**Automate Segmentation and Targeting:** Use the segmentation model results to develop dynamic marketing campaigns (e.g., loyalty rewards for high-value customers, incentives for low-frequency buyers).

**Integrate Real-Time Fraud Monitoring****:** Deploy the fraud detection pipeline in real-time systems to flag suspicious transactions before they are processed, mainly focusing on high-risk shipping types.

**Refine Forecasting Models:** Incorporate external variables like promotions, holidays, and competitor pricing into the sales prediction models to improve forecast accuracy.

**Optimize Shipping Strategies:** Address the high late delivery rate (especially in Standard Class, which accounts for ~60% of delays) by negotiating with logistics providers or promoting alternative shipping options.

**Invest in Data Infrastructure:** Build a centralized analytics platform with automated ETL pipelines, model retraining schedules, and a dashboard to continuously monitor key KPIs.

# **BUSINESS IMPLICATIONS**

The insights uncovered through this project offer real, practical value to decision-makers across several functions of a retail supply chain.

In marketing, customer segmentation revealed that while high-value loyal customers remain crucial for ongoing revenue, the most significant untapped opportunity lies in the new or low-value customer group. These customers, if engaged adequately with targeted campaigns, personalized offers, or improved shipping incentives, have the potential to grow into repeat buyers. By understanding customer Behavior at this level, marketing teams can move away from generic promotions and build more efficient, data-backed engagement strategies.

In fraud prevention, the models we built can be integrated into existing systems to flag unusual transactions before they’re completed. This gives operations and finance teams a clear advantage, as they can act early, reduce financial losses, and avoid damaging customer trust. Since the fraud detection system adapts over time, it’s not just a one-time solution but an evolving tool that grows smarter with more data.

On the inventory and supply chain side, the forecasting models help anticipate when and where demand will increase or decrease. Instead of relying on gut feeling or past trends, managers can make inventory decisions more confidently, stocking ahead of peak seasons or cutting back in slower periods. This helps reduce overstocking, minimize shortages, and improve the overall flow of goods across regions.

Altogether, these insights empower teams to be proactive rather than reactive. From more innovative campaigns to better fraud controls and optimized stock levels, the data gives businesses the clarity they need to make decisions that are not only faster but also smarter.