Optimizing Supply Chains with

Sales Forecasting and Delivery Risk Insights

SOWMIYA

GOWTHAMAN SANTHI

730032380

2024

ACKNOWLEDGEMENT

I am deeply grateful to my supervisor for his extremely valuable guidance and support regarding this research. Of course, special thanks to my family and friends because they encouraged me through everything. Finally, thanks to the professors and colleagues who participated in this project by contributing their knowledge and skills.

EXECUTIVE SUMMARY

Imagine a tool that predicts the future-not in a crystal ball, but in the fast-moving world of supply chain logistics. Late deliveries can disrupt businesses and erode customer trust, ultimately nibbling away at profitability. My project offers a solution to this pressing challenge: an AI-powered system that forecasts the upcoming years sales and profit also predicts the risk of late delivery before it even happens, thus allowing businesses to take proactive measures and stay ahead of potential delays.

The core of modern supply chain management remains finding a balance between efficiency and uncertainty. With increasing customer expectations and developing complexity in global supply chains, the capability to anticipate disruptions has turned into a competitive advantage. Equipping supply chain managers with real-time insight for smarter on-the-fly decision-making is what this project targets. By applying advanced machine learning techniques in combination with a rich dataset from the supply chain domain, the system converts this historical order, shipment, and product information into actionable predictions.

But the real revolution comes in making such powerful insights accessible and user-friendly. Enter the web-based interface: a simple, intuitive tool that allows the decision-maker to enter in some key order information and instantly receive predictions regarding the likelihood of an order facing delays. The tool enables one to do everything from switching shipping modes to prioritize orders at high risk to allocating resources, thus enabling proactive decisions, minimizing disruptions, and ensuring maximum customer satisfaction.
It is beyond data analysis; it's about converting data into foresight. Today, no company operating in a fast-evolving market can afford to react to the problem after it occurred; it has to anticipate the problem. This predictive tool allows the business to proactively minimize risks before they occur by assuring that the right resources are in the right place at the right time.

The objective of this project is to harness the power of predictive analytics to develop smarter and more business-efficient strategies by embedding AI at the core of supply chain operations. This wasn't just about shaving seconds off delivery times but mapping a way to streamline the entire logistics ecosystem, reducing waste and making for a resilient supply chain. With the ability to anticipate the unexpected, this will be the game-changer for companies in most industries in this age of big data, allowing them to stay competitive in the global marketplace.
And with delays inevitable in this world, there the project proved that preparation makes all the difference between chaos and success. This project demonstrates the practical value of AI in solving real-world challenges, where potential risks are converted into opportunities for efficiency and excellence by providing an easy-to-use, real-time prediction tool.

TABLE OF CONTENTS

LIST OF FIGURES

4. Analysis

LIST OF TABLES

CHAPTER 1

INTRODUCTION

INTRODUCTION:

The increasing complexity of global supply chains, driven by rapid technological advancements and shifting consumer demands, has placed unprecedented pressure on businesses to maintain efficient operations. Supplier Relationship Management (SRM) plays a critical role in ensuring these operations run smoothly by mitigating risks, reducing inefficiencies, and maintaining a seamless flow of goods. However, traditional SRM methods often fall short in addressing the growing intricacies of today's data-driven supply chain environments. This project aims to bridge that gap by leveraging Machine Learning (ML) models to enhance SRM, with a focus on forecasting sales and profits and predicting late delivery risks.

At the core of this study lies the hypothesis that data-driven models can offer superior insights into supply chain dynamics, compared to traditional methods. This dissertation-style research explores the application of time series models such as Holt-Winters, SARIMA, and Prophet for forecasting sales and profits, alongside classification models like Random Forest, SVM, and XGBoost for predicting late delivery risks. The objective is not only to forecast supply chain performance but to identify patterns and relationships that can be leveraged to optimize supplier relationships and improve decision-making processes.

The project is based on a comprehensive dataset available on Kaggle, which includes detailed information on order processing, delivery times, customer segments, and product categories. By conducting an Exploratory Data Analysis (EDA), key insights into the factors contributing to late deliveries and supply chain inefficiencies are identified. These insights will inform the application of ML models, enabling a detailed evaluation of their effectiveness in real-world SRM contexts.

While AI models offer significant potential benefits for supply chain management, they also present several challenges. Integrating these models into existing systems can be complicated, as it requires smooth data integration from multiple sources and the ability to make real-time decisions. Disruptions in the supply chain—like those caused by pandemics or geopolitical events—add further complications to using predictive analytics. Additionally, there can be pushback from stakeholders due to the perceived complexity and costs of AI implementations. This project tackles these issues by showing how businesses can adopt predictive models gradually, optimizing operations with minimal disruption and enhancing Supplier Relationship Management (SRM) with clear, measurable results

The methodological approach of the project involves developing predictive models that businesses can use to forecast trends, manage risks, and improve operational efficiency. The project demonstrates the transformative potential of predictive analytics in SRM, allowing companies to make more informed decisions and enhance supplier relationships by anticipating risks and responding proactively. The integration of ML models in supply chain management is increasingly becoming a critical tool for businesses seeking to maintain a competitive edge in today’s dynamic global markets.

In conclusion, this study aims to provide a data-driven solution to one of the most pressing challenges in supply chain management today: how to better predict and manage supplier risks and relationships. By doing so, it hopes to contribute to the growing body of research that explores the role of AI and ML in enhancing SRM and supply chain efficiency.

1.1 CONTEXT AND BACKGROUND

Supply chains today have been disrupted on a scale never seen before. Such disruption fostered by the ongoing COVID-19 pandemic, geopolitical tensions, and rapid technological changes accelerated the need for more creative ways of doing SRM that can avail more dependable, data-driven insights. This demand has opened up the scope, therefore, of AI and ML being used as a rewarding solution that could enable businesses in automating decision-making, predicting supplier performance, and managing risks effectively. To that effect, integrating AI and ML into SRM is nothing but timely for businesses that have to keep up with resilience, efficiency of operations, and more supply chain transparency considering the present challenges.

This work leverages structured and unstructured data from the DataCo Global Smart Supply Chain Dataset. Application of AI approaches on this dataset allows actionable insights to be extracted that could be generalized in real-world supply chains, showing how a business can further optimize its SRM strategy for better efficiency and effectiveness.

1.2 RESEARCH OBJECTIVE:

The core objective of this research is to enhance sales forecasting accuracy and minimize late delivery risks within supply chain management by critically evaluating a range of forecasting models. Using DataCo Global's Smart Supply Chain Dataset, this investigation will apply time-series forecasting models alongside machine learning techniques to predict sales, profit, and delivery risk outcomes. The study aims to identify the most effective model for precise sales and profit forecasting, while also examining how improved forecasts can directly reduce delivery delays and optimize overall supply chain operations, ensuring more efficient and reliable business performance.

1.2.1. **Enhance Sales Forecasting Accuracy**: To improve the precision of sales forecasting by evaluating and applying time-series forecasting models and machine learning techniques on historical sales data from DataCo Global's Smart Supply Chain Dataset.

1.2.2. **Minimize Late Delivery Risks**: To leverage advanced predictive models to identify potential delivery delays in advance, enabling proactive measures to mitigate late delivery risks in supply chain management.

1.2.3. **Optimize Supply Chain Operations**: To explore how accurate forecasting of sales and profits can streamline supply chain operations, reduce inefficiencies, and improve overall decision-making processes related to order management and resource allocation.

1.2.4. **Evaluate and Compare Forecasting Models**: To compare the performance of different forecasting models (e.g., SARIMA, Holt-Winters, Prophet) and machine learning models (e.g., Random Forest, XGBoost) to determine the most effective methods for sales, profit, and risk prediction.

1.2.5. **Develop a User-Friendly Predictive Tool**: To create an accessible web-based tool using Flask that allows stakeholders to predict late delivery risks in real-time, based on key order details, thus improving operational efficiency.

1.3 RESEARCH QUESTIONS

1.3.1 Which time-series forecasting model provides the highest accuracy in predicting sales and profits within the supply chain?
The Holt-Winters model provided the highest accuracy for predicting both sales and profits, as measured by Root Mean Square Error (RMSE), making it the most reliable for supply chain forecasting.

1.3.2. What are the key factors influencing delivery delays, and how can predictive models be optimized to account for these factors?

1.3.3 How does integrating machine learning models with time-series forecasting improve the overall efficiency of supply chain operations?

1.3.4 How can a real-time predictive tool, driven by machine learning, enhance decision-making for supply chain managers in mitigating risks related to late deliveries?

These objectives and research questions will guide the project, focusing on the application of machine learning and time-series forecasting to improve supply chain performance.

1.4 WHY THIS TOPIC MATTERS TO SEVERAL STAKEHOLDERS:

i. **Business and Corporation**

For companies managing complex supply chains, AI-driven SRM can transform supplier interactions. AI provides real-time data analysis, predictive insights, and automated procurement processes, enabling smarter decisions in supplier selection, performance evaluation, and risk management. By streamlining operations and enhancing collaboration, AI helps businesses improve efficiency, reduce costs, and build more resilient supply chains, giving them a competitive edge in global markets.

ii. Suppliers

For suppliers, AI fosters greater transparency and accountability. It allows continuous monitoring of performance metrics and better alignment with buyer expectations. Suppliers that adopt AI tools can strengthen relationships with buyers through consistent, data-driven performance, improving contract negotiations and opening doors to business growth. AI also provides early feedback, enabling suppliers to proactively address areas for improvement.

iii. **Procurement Professionals**:

AI empowers procurement professionals to analyze vast amounts of data, manage supplier portfolios effectively, and make informed decisions that minimize procurement risks. By leveraging AI for supplier selection, contract management, and performance monitoring, procurement teams can ensure better supplier partnerships and avoid supply chain disruptions.

iv. **Technology Providers and Policymakers**:

Technology providers have a growing opportunity to develop AI-enabled SRM tools that handle large datasets and integrate with existing systems. Policymakers play a key role in ensuring regulatory frameworks support the ethical use of AI in supply chain management, addressing data privacy, security, and compliance concerns.

1.5 INVESTIGATION APPROACH

This study employs a comparative model analysis approach to evaluate time series models, using real-world supply chain data from DataCo Global's Smart Supply Chain Dataset. Data analysis tools like Python and libraries such as Statsmodels, Prophet, and SciKit-Learn will be used for model implementation. The research will compare the accuracy of each model based on RMSE values for sales and profit predictions, and their impact on delivery risk management will be assessed using classification techniques like Random Forest to predict late delivery risks.

CHAPTER 2

LITERATURE REVIEW

2.1 AI-Driven Supplier Selection in Global Supply Chains

Supplier selection plays a critical role in determining the overall efficiency and reliability of supply chains. Traditional methods often rely on a limited evaluation of suppliers based on past performance, price, and quality. Ali, Nipu, and Khan (2023) propose a machine learning-based decision support system, utilizing Random Forest algorithms to classify supplier selection criteria. This model enables businesses to analyze multiple variables, such as supplier risk, market dynamics, and sustainability practices, to select the most appropriate suppliers. The study demonstrates how AI-based models can integrate various criteria to improve the supplier selection process and enhance overall supply chain performance.

2.2 Predictive Analytics for Supplier Risk Management

Predictive analytics powered by AI has transformed supplier risk management by moving from reactive to proactive risk identification. Baryannis et al. (2019) emphasize the use of AI models to predict risks such as financial instability, political issues, or environmental concerns in supplier regions. AI allows businesses to track real-time data and historical patterns to forecast potential risks and take preventive actions. Tirkolaee et al. (2021) highlight how machine learning algorithms help in predicting supply chain disruptions by analyzing a vast array of data, including market conditions, supplier performance, and geopolitical risks. This shift from reactive to proactive risk management has reduced unplanned disruptions and improved overall resilience in supply chains.

2.3 Enhancing Supplier Performance Management through AI

Supplier performance management is another area where AI has made significant contributions. According to Belhadi et al. (2024), AI tools automate performance monitoring, making it possible to evaluate suppliers continuously using data from scorecards, production outputs, and customer feedback. These systems predict performance trends, such as late deliveries or product defects, allowing businesses to intervene before issues escalate. Kalekar (2004) also discusses the application of time series forecasting models, such as Holt-Winters exponential smoothing, to improve the monitoring of supplier performance, particularly in terms of forecasting seasonal trends.

2.4 AI and Sustainable Supplier Management

Sustainability has become a key concern in global supply chains, and AI offers innovative solutions to monitor and manage sustainable practices. Dumitrascu et al. (2020) explore how AI can enhance sustainability in supply chains by monitoring environmental and social governance (ESG) metrics. AI systems analyze real-time data from sources such as satellite images and social media to track issues like deforestation, water usage, or labor practices, helping businesses ensure compliance with sustainability goals. Reddy et al. (2021) further demonstrate how AI systems optimize supply chain routes to reduce carbon footprints and manage resources more efficiently

2.5 AI in Sales Forecasting and Demand Prediction

Sales forecasting and demand prediction are critical areas where AI is showing substantial benefits. Kilimci et al. (2019) present a deep learning-based demand forecasting model that surpasses traditional models in accuracy and adaptability. Their model incorporates external factors such as market trends, customer behavior, and seasonality, allowing businesses to optimize their production and inventory planning. Valipour (2015) explores the use of SARIMA models to forecast long-term trends, such as runoff in meteorological applications, highlighting the versatility of time series models for forecasting both demand and environmental factors in supply chains. Kalekar (2004) emphasizes the use of Holt-Winters for handling seasonal demand variations, which are common in industries such as retail and consumer goods.

2.6 Predictive Maintenance in Supply Chains

Reddy et al. (2021) discuss the role of AI in predictive maintenance, where machine learning models analyze sensor data from equipment to predict failures before they occur. This proactive maintenance strategy reduces equipment downtime, optimizes resource use, and enhances production continuity. By leveraging data such as temperature, vibration, and usage patterns, businesses can predict when a piece of equipment is likely to fail and schedule maintenance before the issue becomes critical.

2.7 AI for Inventory Optimization

Maintaining optimal inventory levels is critical for minimizing costs and ensuring product availability. Traditional inventory management methods often rely on historical data and reactive strategies. Reddy et al. (2021) explain that AI-powered inventory optimization can dynamically adjust inventory levels by analyzing real-time demand, lead times, and supplier performance. Techniques such as reinforcement learning allow AI systems to adapt inventory strategies in real-time, ensuring that businesses maintain sufficient stock to meet demand without overstocking, which leads to increased storage costs.

2.8 Time Series Forecasting Models: Holt-Winters and SARIMA

Kalekar (2004) presents the Holt-Winters exponential smoothing model as a robust tool for time series forecasting, particularly in handling seasonal data. This model is well-suited for industries with seasonal demand fluctuations, such as retail or consumer goods. Valipour (2015) similarly discusses SARIMA models in forecasting long-term runoff data, demonstrating their versatility for both supply chain and environmental applications. These models allow businesses to forecast sales and demand with higher precision, enabling better inventory planning and risk management.

2.9 AI-Powered Web Development in Supply Chains

Though not directly related to the primary focus of supply chain management, Idris et al. (2021) discuss the role of web technologies like Flask and Django in developing scalable and efficient platforms for managing supply chain data. AI-powered platforms can leverage these technologies to provide real-time dashboards, monitoring tools, and analytics, improving decision-making for supply chain managers.

2.10 Future Trends in AI for Supply Chain Management

The future of AI in supply chain management will involve more sophisticated and integrated systems that can manage increasingly complex global networks. Reddy et al. (2021) emphasize that AI will play a pivotal role in making supply chains more resilient and agile, with real-time analytics, automated decision-making, and improved transparency across the supply chain. This will be critical in addressing future challenges such as resource shortages, climate-related disruptions, and increasingly complex regulatory environments.

CHAPTER 3

OVERVIEW OF METHODS

Machine Learning (ML) and Time Series Analysis are two distinct yet complementary approaches in predictive modeling. Machine Learning involves the use of algorithms that enable systems to automatically learn and improve from experience without being explicitly programmed. The core of ML lies in its ability to identify patterns and make decisions based on data, where the model is trained using input-output pairs. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. (Mahesh, 2018) The flexibility of ML models allows them to handle high-dimensional data, complex dependencies, and nonlinear relationships that may be difficult to capture with traditional statistical methods. In the context of predictive analytics, ML methods are well-suited for learning from a wide range of features, allowing for accurate predictions in scenarios with intricate and dynamic data structures.

On the other hand, Time Series Analysis is specifically focused on data that is collected over time at consistent intervals. It is primarily used to model the temporal dependencies, such as trends, seasonality, and autocorrelations, that arise in sequential data. Unlike conventional regression models, Time Series Analysis accounts for the time-based ordering of observations, which introduces dependencies across different time points. The key objective is to forecast future values based on historical trends, making it highly valuable in applications like demand forecasting, risk management, and anomaly detection. Time Series Analysis methods typically assume that the underlying process generating the data is stationary, or that appropriate transformations can make it stationary, which is critical for producing reliable forecasts.

When combined, Machine Learning and Time Series Analysis create powerful hybrid models that benefit from the strengths of both techniques. ML enhances time series models by incorporating external covariates and learning complex relationships, while Time Series Analysis ensures that temporal dependencies are preserved, making this combination highly effective in real-world applications such as late delivery risk prediction.

3.1 MODELS IMPLEMENTED AND TESTED

3.1.1 SARIMA

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model builds on the ARIMA model, allowing it to handle both seasonal and non-seasonal patterns in time series data. While ARIMA is focused on non-seasonal trends, SARIMA adds the ability to account for seasonal fluctuations, making it ideal for data with recurring cycles. The model is expressed as SARIMA (p, d, q).

p, d, and q	➢ number of non-seasonal autoregressive (AR) terms, the level of differencing needed to make the data stable, and the number of non-seasonal moving average (MA) terms.

P, D, and Q	➢ seasonal counterparts for autoregression, differencing, and moving average components.
s	➢ length of the seasonal cycle (e.g., monthly or quarterly data).

Table 1: SARIMA Model Parameters Evaluation

SARIMA is built on the Box-Jenkins methodology, which involves identifying the appropriate model, estimating its parameters, and validating it through diagnostic checks. (Valipour, 2015) The seasonal differencing operation (D) enables the model to remove periodic trends, making the time series stationary and, therefore, more predictable. This combined approach allows SARIMA to handle both short-term dependencies (through AR and MA terms) and long-term seasonal fluctuations (via seasonal components), making it exceptionally effective for data with recurring cycles. Its flexibility in addressing both trend and seasonality has made SARIMA a preferred choice in domains such as econometrics, meteorology, and hydrology, where capturing complex seasonal patterns is essential for accurate long-term forecasting.

3.1.2 PROPHET

Prophet is a powerful time series forecasting tool created by Facebook’s Core Data Science team. It’s built to work well with time series data that show clear seasonal trends and patterns, even when there are gaps or unusual spikes in the data. Prophet breaks the data down into three key parts: trend, which tracks long-term changes; seasonality, which captures regular cycles; and holidays or special events, which account for specific occurrences that could influence the forecast.

Unlike traditional statistical models, Prophet uses an additive model that allows users to adjust for various complexities in the data. Its main advantage lies in its ability to scale for large datasets with minimal tuning, making it user-friendly and accessible even for those without deep expertise in time series analysis. (Satrio, Darmawan, Nadia, & Hanafiah, 2021) Prophet is designed for long-term forecasting and excels at capturing trends in data where sudden shifts or trend changes occur, such as in business operations or supply chain forecasting

3.1.3 HOLT-WINTERS EXPONENTIAL SMOOTHING

The Holt-Winters Exponential Smoothing model is a powerful time series forecasting method designed to handle data with both trend and seasonality components. (SolarWinds, 2019) It extends basic exponential smoothing by incorporating three key elements: level, trend, and seasonality, making it highly effective for time series that exhibit regular patterns over time.

There are two variations of the Holt-Winters model:

1. **Additive Model:** This is used when the seasonal variations remain constant in size, regardless of the level of the data. It is suitable for time series where seasonal fluctuations do not change in amplitude as the data grows or shrinks.

2. **Multiplicative Model:** This is applied when seasonal variations increase or decrease proportionally to the level of the data. It works well when the amplitude of the seasonality grows with the trend of the data.

The model uses three smoothing equations to update the estimates of the level, trend, and seasonality at each time step:

- Level (L): The smoothed value of the series after accounting for trend and seasonality.
- Trend (T): The rate of change in the series' level over time.
- Seasonality (S): The repeating pattern observed in the data.

The forecasting equations combine these components to predict future values, with the general form being:

$$\boxed{y_{t= (L_{t-1} + T_{t-1})} \cdot S_{t-L}}$$

L_{t-1}	level at time t−1
T_{t-1}	trend at time t−1
S_{t-L}	seasonal factor from the same period in the last cycle (where LLL is the seasonal length)

Table 2: Holt-Winters Model Components

This model's flexibility in adjusting to both trend and seasonal changes makes it ideal for applications in forecasting sales, inventory levels, and demand, especially in industries with strong seasonal patterns. (Kalekar, 2004).

3.1.4 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machines (SVM) are supervised learning algorithms commonly used for both classification and regression. They work by identifying the best hyperplane that separates data points from different classes with the widest margin possible. This approach helps reduce the chances of misclassification, making SVM particularly effective for handling complex, high-dimensional data.

The core idea behind SVM is to transform the input data into a higher-dimensional feature space where a linear separation is possible, even for data that is not linearly separable in the original space. This is achieved through the use of kernel functions (Jakkula, 2006), such as polynomial and radial basis functions (RBF), which map the data into a new space where the optimal separating hyperplane can be defined.

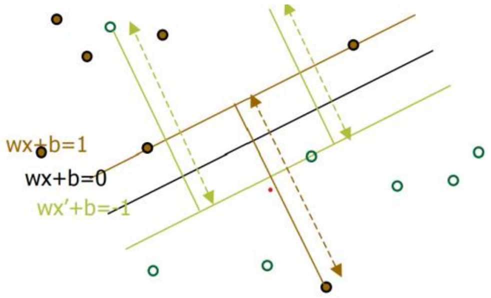


Figure 1: Hyper plane

Source: (Jakkula, 2006)

Mathematically, SVM aims to minimize the following optimization problem:

$$\boxed{min\frac{1}{2}||w||^2 \text{ subject to } y_{i(w^T\ x_i+b)} \geq 1 - \xi_i}$$

Where:

w	weight vector
b	bias term
y_i	labels of the data points

x_i	the input features
ξ_i	slack variables to allow for soft margins, handling outliers and non-linear separability

Table 3: Support Vector Machine (SVM) Model Parameters

SVM’s strength lies in its ability to balance complexity and error through the use of regularization, which controls the trade-off between maximizing the margin and minimizing classification errors (Gatto, 2021). Additionally, the kernel trick allows SVM to efficiently handle non-linear decision boundaries by implicitly computing the transformations without ever needing to explicitly transform the data.

3.1.5 K-NEAREST NEIGHBOUR

K-Nearest Neighbours (KNN) is a versatile, non-parametric algorithm used for both classification and regression tasks. It classifies data points based on the majority vote of their 'k' nearest neighbors in a feature space. The distance between data points is calculated using metrics such as Euclidean, Manhattan, or Minkowski distance (Peterson, 2009). KNN is particularly effective for tasks where the decision boundary is nonlinear and does not require an explicit training phase, making it highly adaptable to various domains like image recognition, medical diagnosis, and pattern recognition. KNN’s performance depends on the choice of 'k' and the distance metric, which determine its sensitivity to local data structures. It has a clear advantage in simplicity and ease of interpretation but can become computationally expensive as the dataset grows.

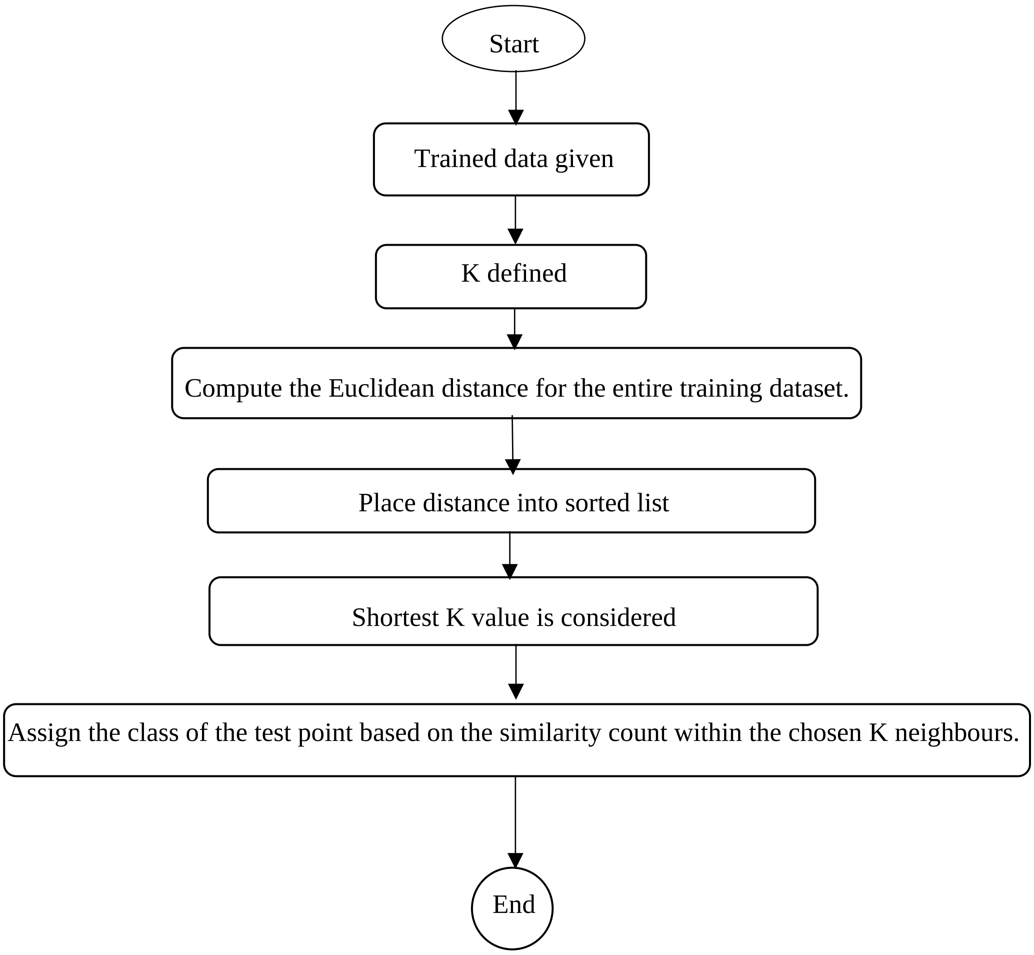

Start
Trained data given
K defined
Compute the Euclidean distance for the entire training dataset.
Place distance into sorted list
Shortest K value is considered
Assign the class of the test point based on the similarity count within the chosen K neighbours.
End

Figure2: KNN Flow Chart

Source: (Kumar & Jain, 2022)

3.1.6. XGBoost

XGBoost is a cutting-edge implementation of gradient boosting designed for maximum efficiency and flexibility. One of its primary innovations is the use of regularization to prevent overfitting, optimizing the model's balance between accuracy and complexity. Additionally, XGBoost supports parallel processing, which significantly speeds up training time compared to traditional boosting algorithms. It introduces a sparsity-aware algorithm, making it highly effective at handling missing values and sparse datasets, common in real-world applications (Chen & Guestrin, 2016). Moreover, weighted quantile sketching ensures that even large datasets with imbalanced distributions are managed efficiently. XGBoost's ability to distribute workloads across clusters allows it to scale seamlessly, making it suitable for massive datasets and computationally intensive tasks. Its predictive accuracy and robustness have made it a dominant algorithm in machine learning competitions and production environments, where it excels in both classification and regression tasks. By combining these innovations, XGBoost has become a widely adopted solution for tasks like fraud detection, risk modeling, and click-through rate prediction.

3.1.7 RANDOM FOREST

Random Forest is an ensemble learning algorithm used for both classification and regression tasks, known for its robustness and versatility (Breiman, 2001). It operates by constructing a multitude of decision trees during training. Each tree is built using a different random subset of data, and at each node, the features are randomly selected to split the data. This introduction of randomness at multiple levels helps to reduce variance without significantly increasing bias, making the model resistant to overfitting, especially with noisy datasets.

The process starts with bootstrap aggregating, or bagging, where each tree is trained on a random subset of the data, giving each tree a unique view of the dataset. At each split within a tree, a random set of features is chosen, which helps reduce correlation between the trees and increases model reliability. In the end, the Random Forest makes its final prediction by combining the output of all the trees, either through majority vote for classification or averaging for regression.

Mathematically, for classification, the Random Forest prediction $\hat{y}$ for an input x is defined as:

$$\boxed{\hat{y} = mode\{h_{k(x)}\}N_{k=1}}$$

Where:

N	number of trees in the forest
$h_{k(x)}$	is the prediction from the k^{th} decision tree for input X
mode	most frequent prediction among the trees (majority vote)

Table 5: Random Forest Model Parameters

For regression tasks, the equation is expressed as:

$$\boxed{\hat{y} = \frac{1}{N} \sum_{k=1}^{N} h_{k(x)}}$$

Where the final prediction $\hat{y}$ is the average of the individual tree predictions.

Random Forests are known for their generalization error decreasing as more trees are added, eventually converging due to the Strong Law of Large Numbers. This ensures that adding more trees doesn’t result in overfitting, making the model scalable and robust for both small and large datasets.

Additionally, Random Forests provide estimates of feature importance, which is calculated by evaluating how much each feature contributes to reducing impurity in the dataset. This allows for better interpretability of the model, as it highlights the most significant predictors in high-dimensional data.

Random Forests are widely used in a variety of fields, including finance, medicine, and e-commerce, due to their flexibility and ease of use in handling complex, nonlinear data patterns.

3.1.8. ROOT-MEAN-SQUARE ERROR (RMSE)

Root Mean Square Error (RMSE) is a popular metric for evaluating how well a predictive model performs by measuring the difference between its predictions and actual outcomes. It’s calculated by taking the square root of the average squared differences between the predicted and actual values, which means that larger errors have a bigger impact on the final score.

The RMSE is calculated as:

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

Where:

n	number of observations
y_i	actual observed values
y_i	the predicted values

Table 5: Regression Model Notation

RMSE is particularly useful when errors are normally distributed, as it amplifies large deviations between predicted and observed values. This characteristic makes RMSE a more sensitive metric than the Mean Absolute Error (MAE), which treats all errors equally (Chai & Draxler, 2014). RMSE provides insight into the overall model accuracy and is commonly applied in fields such as regression analysis, time series forecasting, and machine learning.

3.1.9. CONFUSION MATRIX

A Confusion Matrix is a comprehensive tool used in classification problems to evaluate the performance of a model. It represents the model’s predictions against actual values in a tabular format. The matrix contains four key values:

1. True Positives (TP): Cases correctly predicted as positive.
2. True Negatives (TN): Cases correctly predicted as negative.
3. False Positives (FP): Negative cases incorrectly predicted as positive (Type I error).
4. False Negatives (FN): Positive cases incorrectly predicted as negative (Type II error).

		Predicted	
		POSITIVE	NEGATIVE
Actual	POSITIVE	True Positive	False Negative
	NEGATIVE	False Positive	True Negative

Figure 3: Confusion Matrix Terminology

From the Confusion Matrix, several performance metrics can be derived, including:

- **Accuracy:** Accuracy measures the proportion of correct predictions out of all predictions made.

$$\frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}$$

- **Precision:** Precision quantifies the proportion of positive predictions that were correct.

$$\frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$$

- **Recall:** Recall measures how well the model captures actual positive cases.

$$\frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$$

- **F1 Score:** The F1 Score balances precision and recall, providing a harmonic mean.

$$\frac{2.(Precision.Recall)}{Precision+Recall}$$

The Confusion Matrix and these derived metrics provide deeper insight into a model’s strengths and weaknesses, especially in scenarios involving imbalanced datasets, where accuracy alone may be misleading. For instance, high false positives in medical diagnostics could indicate a problem with the classifier in distinguishing between healthy and diseased patients

3.1.10. FLASK

Flask is a micro-framework for Python, known for its simplicity, flexibility, and minimalistic nature. It is designed to build small to medium-scale web applications quickly and efficiently, focusing on providing the essentials needed for web development without unnecessary complexity. Flask is based on two core components: **Werkzeug**, a WSGI (Web Server Gateway Interface) toolkit, and **Jinja2**, a powerful templating engine (Idris, Mohd, & Shamala, 2021). These components enable developers to manage HTTP requests and dynamically generate HTML content.

One of Flask’s strengths is its ability to handle **routing**, where URL patterns are mapped to Python functions, allowing developers to define how requests to specific URLs should be handled. Additionally, Flask provides a robust environment for handling GET, POST, and other HTTP methods, while maintaining a high degree of customization for request and response handling. By offering built-in support for request data, sessions, and cookies, Flask simplifies web development while maintaining flexibility.

Key Features of Flask:

1. **Minimalism and Flexibility**: Flask does not impose any project layout, making it ideal for developers who want full control over the application structure.

2. **Modularity through Extensions**: Flask's core functionality can be extended by integrating external libraries for database support (e.g., **SQLAlchemy**), authentication, form validation, and more. This modularity makes Flask scalable, from simple applications to more complex ones.

3. **Templating with Jinja2**: Flask's integration with **Jinja2** allows developers to create dynamic HTML templates, incorporating variables and control structures like loops and conditionals. This enables efficient web page rendering with content based on user input or database data.

4. **WSGI Compliant**: Flask leverages **Werkzeug** to provide a clean interface between Python applications and web servers, ensuring that Flask apps can run in diverse deployment environments such as **Gunicorn** or **uWSGI**.

5. **Built-in Development Server**: Flask comes with a built-in debugger and development server, which streamlines the development process by providing real-time error tracking and debugging.

Deployment and Scaling:

Despite its minimalism, Flask is capable of handling production-level applications. It can be scaled using various deployment solutions such as **Docker** containers, and its ability to integrate with other tools (e.g., **Nginx** for reverse proxying) makes it a strong candidate for microservices architecture. Flask is also compatible with NoSQL databases like MongoDB, adding to its versatility for modern web application stacks.

3.1.11 CONFIDENCE LEVEL

A **confidence interval** provides a range of values within which we expect the true value of a parameter or prediction to fall, given a specific level of confidence. It is a crucial tool in statistical analysis, quantifying the uncertainty associated with predictions or estimates. A **95% confidence interval**, for example, implies that if the same process is repeated many times, 95% of the computed intervals would contain the true value.

General Formula:

$$CI_{upper= \widehat{y}} + z.std\_error$$

$$CI_{lower\ = \widehat{y}} - z.std\_error$$

Where:

$\widehat{y}$	represents the predicted value
z	z-score for the desired confidence level (1.96 for 95%)
std_error	standard error of the estimate

Table 6 : Confidence Interval Formula Components

Significance:

- **Quantifying Uncertainty**: Confidence intervals help provide a range around the estimate, accounting for the inherent variability in data.

- **Decision Support**: They aid in making informed decisions by showing the reliability of predictions, useful in forecasting and risk analysis.

CHAPTER 4

ANALYSIS

The DataCo Smart Supply Chain dataset, sourced from Kaggle, (DataCo SMART SUPPLY CHAIN, n.d.) is a comprehensive collection of data designed for advanced analysis of supply chain dynamics. This dataset provides detailed information on various aspects of supply chain operations, including product orders, sales, customer information, delivery performance, and regional sales insights. It contains a wide range of variables that are essential for understanding the relationships between supply chain efficiency, sales performance, and customer behavior across different markets.

Attribute	**Description**	**Data Type**
Type	The type of payment method used for the order (e.g., DEBIT, TRANSFER)	String
Days for shipping (real)	The actual number of days it took to ship the order	Integer/Float
Days for shipment (scheduled)	The scheduled number of days for shipping the order	Integer/Float
Benefit per order	Profit or loss per order after costs are deducted	Float
Sales per customer	Total sales amount per customer	Float
Delivery Status	Current status of the order's delivery (e.g., Advance shipping, Late)	String
Late_delivery_risk	Binary indicator showing whether the order is at risk of late delivery	Integer (0/1)
Category Id	Numeric identifier for the product category	Integer
Category Name	Name of the product category	String
Customer City	The city where the customer is located	String
Customer Country	The country where the customer is located	String
Customer Email	Email address of the customer (masked)	String
Customer Fname	First name of the customer	String
Customer Id	Unique identifier for the customer	Integer
Customer Lname	Last name of the customer	String
Customer Password	Password associated with the customer's account (masked)	String
Customer Segment	Segment the customer belongs to (e.g., Consumer)	String
Customer State	The state or province of the customer	String
Customer Street	Street address of the customer	String
Customer Zipcode	Postal code of the customer	String/Integer
Department Id	Numeric identifier for the department handling the product	Integer

Department Name	Name of the department handling the product (e.g., Fitness)	String
Latitude	Latitude of the customer's location	Float
Longitude	Longitude of the customer's location	Float
Market	Market region where the order was placed	String
Order City	The city where the order was placed	String
Order Country	The country where the order was placed	String
Order Customer Id	Unique identifier of the customer who placed the order	Integer
Order date (DateOrders)	Date and time the order was placed	DateTime
Order Id	Unique identifier for the order	Integer
Order Item Cardprod Id	Unique identifier for the product in the order	Integer
Order Item Discount	Discount applied to the product in the order	Float
Order Item Discount Rate	Discount rate applied to the product in the order	Float
Order Item Id	Unique identifier for the order item	Integer
Order Item Product Price	Price of the product in the order	Float
Order Item Profit Ratio	Profit ratio from the product in the order	Float
Order Item Quantity	Quantity of the product ordered	Integer
Sales	Total sales generated by the order	Float
Order Item Total	Total price of the product after discounts	Float
Order Profit Per Order	Profit margin earned on each order	Float
Order Region	Geographical region where the order was placed	String
Order State	State where the order was placed	String
Order Status	Status of the order (e.g., COMPLETE, PENDING)	String

Order Zipcode	Postal code where the order was placed	String/Integer
Product Card Id	Unique identifier for the product	Integer
Product Category Id	Numeric identifier for the product category	Integer
Product Description	Description of the product	String
Product Image	URL link to the product image	String
Product Name	Name of the product (e.g., Smart watch)	String
Product Price	Price of the product	Float
Product Status	Status of the product availability (e.g., 0 for out of stock)	Integer
Shipping date (DateOrders)	Date and time the order was shipped	DateTime
Shipping Mode	Mode of shipment (e.g., Standard Class, Express Class)	String

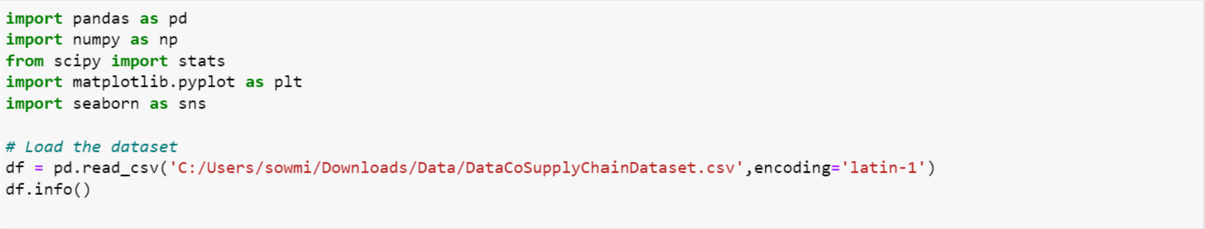
Table 7 : Data Attributes

4.1 DATA PREPARATION

Data preprocessing is a crucial step in preparing this dataset for analysis. It involves cleaning, transforming, and organizing the data to ensure that it is ready for exploration, modeling, and drawing meaningful insights. The steps taken for preprocessing are as follows:

4.1.1 Loading and Initial Inspection:

After loading the dataset using pandas, the first step was to conduct an initial inspection using the .info() method. This provided a clear overview of the data structure, including the data types, column names, and the presence of any missing or inconsistent values. Understanding the dataset's structure is critical for identifying potential issues early in the preprocessing pipeline.



```python
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('C:/Users/sowmi/Downloads/Data/DataCoSupplyChainDataset.csv',encoding='latin-1')
df.info()
```

4.1.2 Dropping Irrelevant Columns:

Several columns, such as 'Customer Email', 'Customer Fname', 'Customer Lname', 'Customer Password', and 'Product Description', were deemed unnecessary for the analysis and were dropped. These fields, although important for certain contexts (e.g., customer personalization or security), do not contribute to the supply chain's operational analysis and were therefore removed to reduce data dimensionality. This step helps optimize the dataset for further processing and prevents any irrelevant variables from influencing the analysis outcomes**.**

4.1.3 Handling Missing Data:

The dataset's missing values were addressed through two potential approaches:

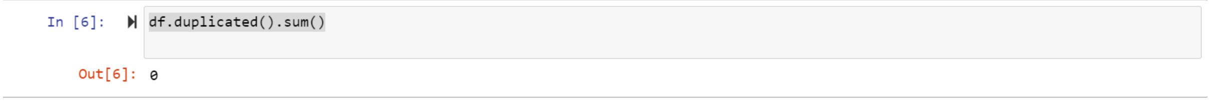
- **Option 1:** Rows with missing values could be dropped entirely, which is appropriate when the number of missing entries is relatively small and not crucial for the analysis.

- **Option 2:** Missing values could be imputed using the mean (or median) values for numerical columns. This approach is often preferred when preserving the dataset's integrity is important, especially if the missing data is significant.

Proper handling of missing values ensures that the dataset remains complete and avoids biases that could arise from incomplete data.

4.1.4 Removing Duplicates:

Duplicate rows in the dataset can distort the analysis by over-representing certain data points. Therefore, the dataset was examined for duplicates, which revealed that there were no duplicate entries.



```
In [6]:  df.duplicated().sum()

Out[6]: 0
```

4.2. ETHICAL IMPLICATION

In conducting this dissertation, several ethical principles were strictly followed to ensure the research was carried out with transparency, integrity, and accountability.

a) **Informed consent** was not directly required, as the data used was publicly available. However, transparency about the dataset and its purpose was maintained throughout the research to ensure ethical compliance and respect for the original data source.

b) **Data privacy** was a priority, with personally identifiable information (PII) either removed or anonymized. Even though the data was public, adherence to regulations like GDPR ensured the protection of sensitive information.

c**) The principle of non-maleficence** was upheld to prevent harm. The research aimed to improve supply chain efficiency by predicting late delivery risks, resulting in socially beneficial outcomes without negatively affecting individuals or organizations.

d) **Confidentiality and data security** were maintained by securely storing the data in encrypted environments with limited access. This ensured the data's use was restricted solely for this dissertation.

e) **Fairness and bias mitigation** were addressed during data processing and model evaluation. The dataset was checked for imbalances to ensure that no customer segment or region was unfairly disadvantaged.

f) **Honesty and integrity** were prioritized in reporting results. The dissertation transparently documented model performance, limitations (e.g., data imbalances), and outcomes, avoiding any manipulation of findings. It’s worth noting that the 2018 data was incomplete, which explains the sharp decline in the figures for that year.

h) The study emphasized the ethical use of **AI for social good**. By helping supply chain managers predict late delivery risks and make informed decisions, the model contributes to operational improvements, aligning with AI's role in creating sustainable, efficient supply chains.

4.3 EXPLORATORY DATA ANALYSIS FOR INSIGHTS

Having pre-processed the data to ensure its quality and consistency, the next step in the analysis is to gain deeper insights into the dataset through Exploratory Data Analysis (EDA). EDA is a crucial phase in the data analysis workflow that allows for the exploration of underlying patterns, trends, and relationships within the data. By leveraging statistical methods and visualizations, EDA helps uncover key insights, detect anomalies, and identify important features that could impact sales forecasting and delivery risk predictions. This stage sets the foundation for building robust forecasting models by enabling a deeper understanding of the variables and their interactions within the supply chain dataset.

4.3.1 Comprehensive Sales Breakdown by Product Category

This graph gives an in-depth analysis of sales performance across various product categories, offering key insights into which categories are driving revenue and contributing the most to overall sales.

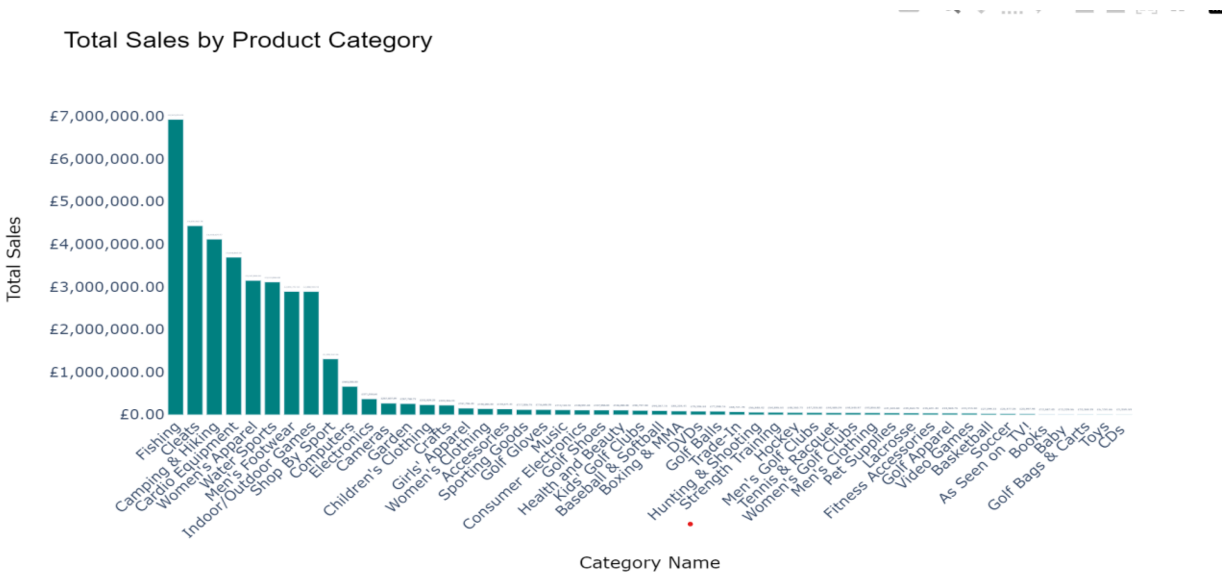


Figure 4.1 : Total Sales by Category

i. **Top-performing Categories**:

Fishing leads in sales, likely driven by its broad consumer appeal and increasing demand for outdoor activities. Categories such as Camping & Hiking and Outdoor Equipment also rank highly, reflecting strong interest in outdoor recreation, especially during periods of heightened leisure activity.

ii. **Middle-performing Categories**:

Children's Clothing and Indoor/Outdoor Games show moderate sales, indicating consistent but niche demand. These categories cater to specific needs, such as seasonal shopping or family-oriented products, contributing steadily but without the broader appeal of outdoor goods.

iii. **Low-performing Categories**:

Golf Bags & Carts, Lacrosse, and Books exhibit lower sales, likely due to their niche markets. Their limited consumer base suggests that these categories face challenges from more specialized demand or competition from digital alternatives, as seen with books.

iv. **Sales Distribution**:

Sales are skewed heavily towards a few categories, with the majority contributing marginally. This highlights an opportunity to focus on expanding top-performing products while reassessing the strategy for lower-performing ones.

4.3.2 Customer Spending Insights Across Product Categories

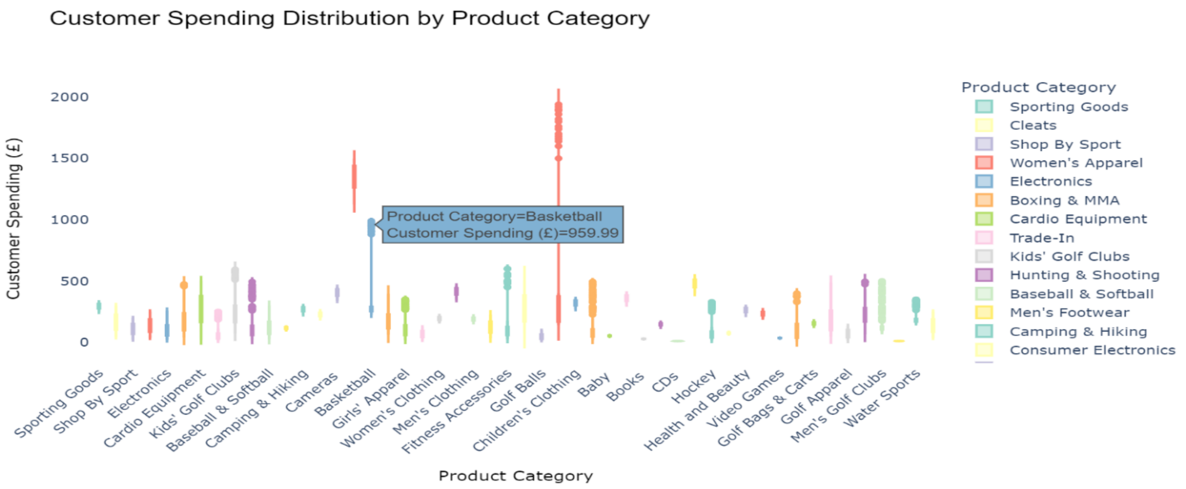


Figure 4.2: Customer Spending by Product Category

Categories like Basketball, Golf Bags & Carts, and Video Games show a wide range of spending, with some outliers reaching over £1,500, suggesting the presence of premium products. In contrast, categories such as Books and CDs see consistently lower spending, with most purchases below £500, likely due to their lower price points. This analysis is valuable for guiding pricing strategies, marketing efforts, and inventory management, particularly for high-value categories with greater spending potential.4.3.3 Profitability Across Categories

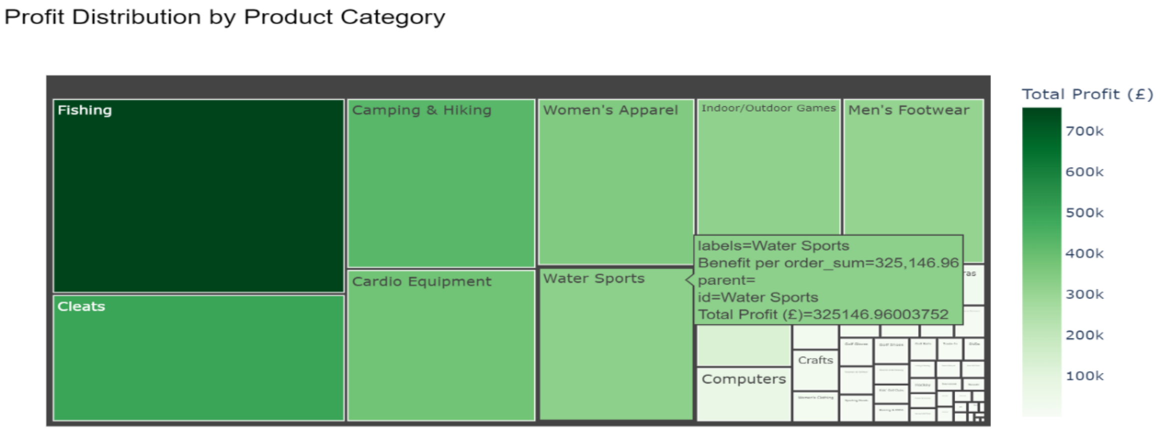


Figure 4.3: Profitability Distribution

categories like Fishing, Camping & Hiking, and Cleats are the top profit drivers, each contributing significant revenue, with Fishing alone exceeding £700k. Smaller categories like Computers and Crafts show much lower profit contributions, indicating less demand or more competitive pricing. This visualization helps identify where the business should focus its efforts, suggesting further investment in high-profit categories to maximize returns while exploring opportunities for growth in underperforming segments.

4.3.4 Shipping Performance and Late Delivery Risk Assessment

Understanding shipping efficiency is essential to identifying potential bottlenecks and optimizing supply chain operations. To gain insights, we analyze the number of orders by shipping mode, comparing the actual shipping time to the scheduled shipping time, and evaluate the late delivery risk across the top 10 product categories. This analysis sheds light on how well the shipping process adheres to schedules and highlights categories where late delivery risks are most prominent, providing a comprehensive view of current performance and areas for improvement.

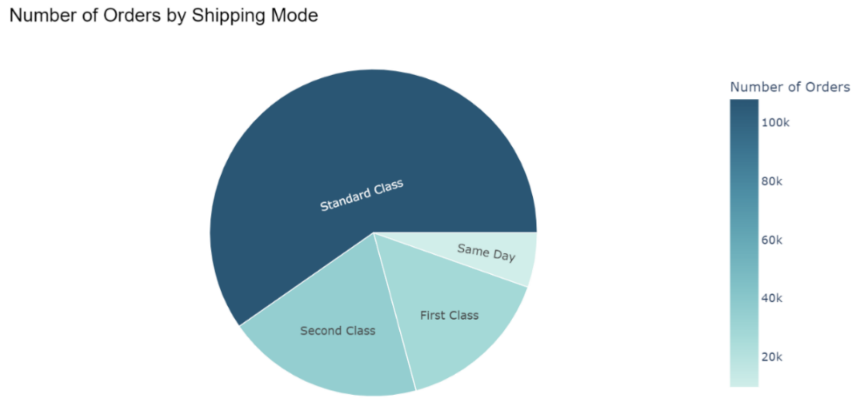


Figure 4.4 : Orders by Shipping Mode

This chart illustrates the distribution of orders based on the shipping mode selected. It is clear that the majority of orders are processed through Standard Class, indicating that it is the most commonly used shipping option, likely due to its cost-effectiveness. Smaller portions of the orders are handled through Second Class, First Class, and Same Day shipping modes, which might be reserved for more urgent deliveries or premium services. This distribution highlights the importance of optimizing Standard Class shipping to minimize late delivery risks, as it handles the bulk of the orders.

Actual vs Scheduled Shipping Time

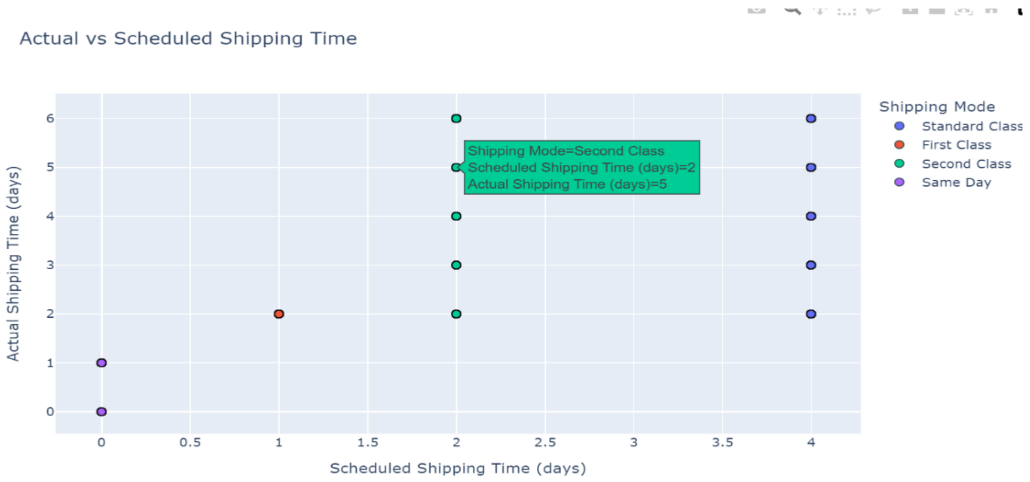


Figure 4.5: Shipping Time Delayed

This scatter plot compares the scheduled shipping time against the actual shipping time for various shipping modes. Each point represents the difference between what was expected and what actually occurred. Notably, the plot reveals that Second Class shipping often experiences significant delays, where actual shipping time exceeds the scheduled time. Identifying these discrepancies can help pinpoint inefficiencies in the shipping process and improve service levels, especially for shipping modes that are prone to longer delays than promised.

Late Delivery Risk

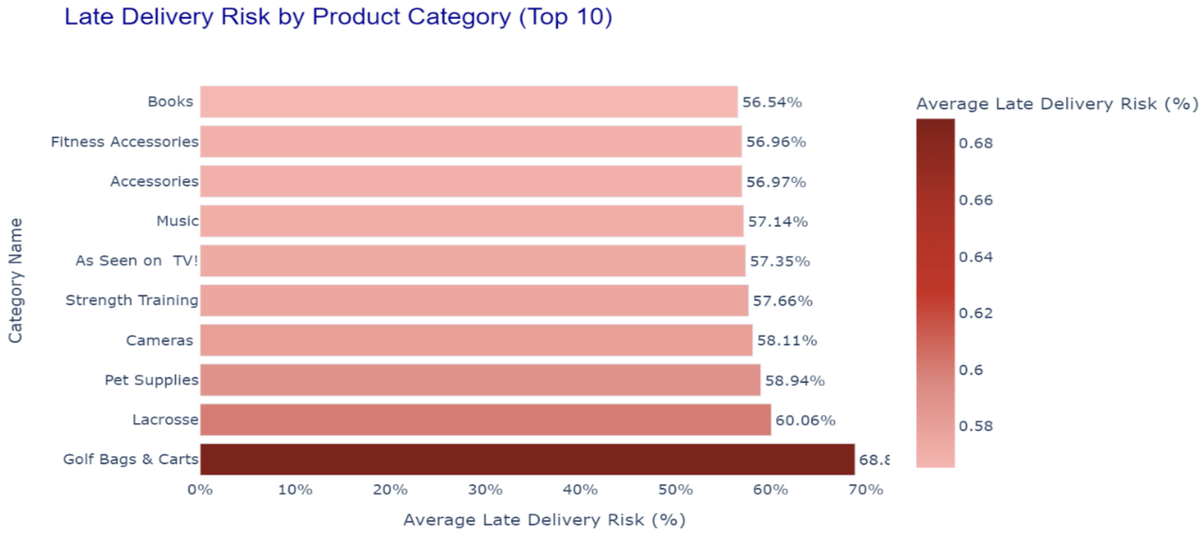


Figure 4.6 : Late Delivery Risk Products

This bar chart showcases the late delivery risk for the top 10 product categories, with Golf Bags & Carts exhibiting the highest risk, followed by Lacrosse and Pet Supplies. These categories consistently show a high percentage of late deliveries, exceeding 60% for some products. On the other hand, categories like Books and Fitness Accessories show comparatively lower, but still concerning, levels of late delivery risk. These insights are crucial for identifying which product categories require focused interventions to reduce delays.

The insights gained from analyzing shipping performance and late delivery risks are critical to the broader goals of this research. By identifying which shipping modes and product categories are more prone to delays, businesses can implement targeted strategies to mitigate these risks.

Additionally, these findings that we have done in the EDA will support our late delivery risk prediction model, ensuring that high-risk categories are prioritized for intervention. Overall, this analysis not only highlights current inefficiencies but also serves as a foundation for building more accurate forecasts and predictive models, ultimately helping to optimize the entire supply chain process.

CHAPTER 5

SALES AND PROFIT FORECASTING

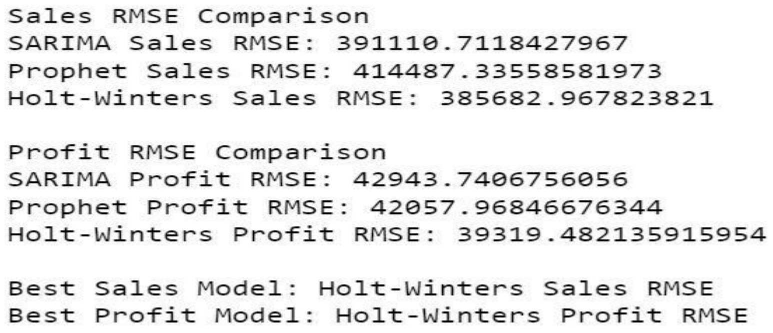
In this, the primary goal was to forecast sales and profit over the next two years using time series data from the DataCo Smart Supply Chain dataset. To achieve this, three different models—SARIMA, Prophet, and Holt-Winters—were employed to generate forecasts. Model selection was guided by a quantitative evaluation using Root Mean Square Error (RMSE), a standard metric for measuring the accuracy of predictive models.

5.1.1 TRAINING AND TESTING:

The dataset was divided into two parts: 80% for training the models and 20% for testing. The training data helps the models learn patterns from past data, while the test data is used to see how well the models perform on new, unseen information. This approach ensures a more accurate and reliable evaluation of the model's performance.

5.1.2 RMSE:

The RMSE was computed for each model’s sales and profit predictions. RMSE measures the square root of the average squared differences between the actual and predicted values, penalizing larger errors more heavily. A lower RMSE value indicates a more accurate model. The table below summarizes the RMSE values for each model



```
Sales RMSE Comparison
SARIMA Sales RMSE: 391110.7118427967
Prophet Sales RMSE: 414487.33558581973
Holt-Winters Sales RMSE: 385682.967823821

Profit RMSE Comparison
SARIMA Profit RMSE: 42943.7406756056
Prophet Profit RMSE: 42057.96846676344
Holt-Winters Profit RMSE: 39319.482135915954

Best Sales Model: Holt-Winters Sales RMSE
Best Profit Model: Holt-Winters Profit RMSE
```

Figure 5.1 : Sales & Profit RMSE Comparison

5.1.2.1 Profit RMSE:

Similarly, Holt-Winters exhibited a Profit RMSE of 39,319.48, lower than SARIMA’s 42,943.74 and Prophet’s 42,057.97. The reduced error in profit forecasting indicates that Holt-Winters was better equipped to capture the underlying trends and seasonality associated with profit variations, which is essential for making informed decisions regarding financial planning and budgeting.

5.1.2.1 Sales RMSE:

Holt-Winters achieved the lowest RMSE value of 385,683, compared to SARIMA’s 391,110.7 and Prophet’s 414,487.3. This difference, though seemingly marginal, is crucial in forecasting large-scale operations like supply chain management, where even slight improvements in prediction accuracy can translate to significant financial benefits.

5.2 MODEL COMPARISON AND INSIGHTS

5.2.1 **SARIMA:** The SARIMA model (Seasonal Autoregressive Integrated Moving Average) was effective at capturing both trend and seasonal patterns in the data. While its RMSE values were relatively low, it did not outperform Holt-Winters in terms of predictive accuracy for either sales or profit.

5.2.2 **PROPHET:** Prophet, developed by Facebook, is well-suited for data with irregular seasonal patterns. Although it performed reasonably well, the Prophet model had the highest RMSE values for both sales and profit, indicating that it was less precise in capturing the decline in sales and profit during the latter part of the testing period.

5.2.3 **HOLT-WINTERS:** Holt-Winters Exponential Smoothing proved to be the most accurate model, with the lowest RMSE for both sales and profit. The model effectively captured both seasonal variations and trend changes, particularly during periods of sharp decline. Given its superior performance, Holt-Winters was selected as the final model for forecasting future sales and profit.

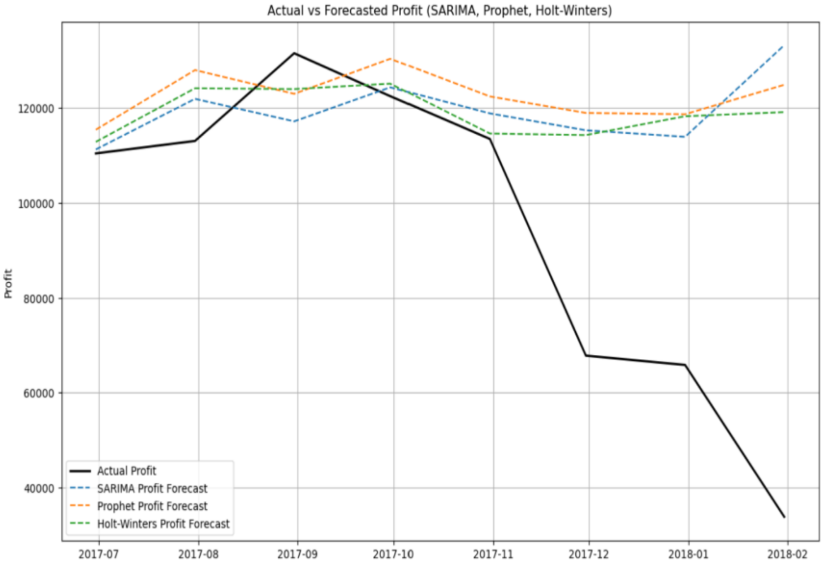


Figure 5.2: Actual vs Forecasted Profit Model

The Actual vs Forecasted Profit graph demonstrates that Holt-Winters tracked the actual profit more closely than SARIMA and Prophet, particularly during the steep decline toward the end of the testing period.

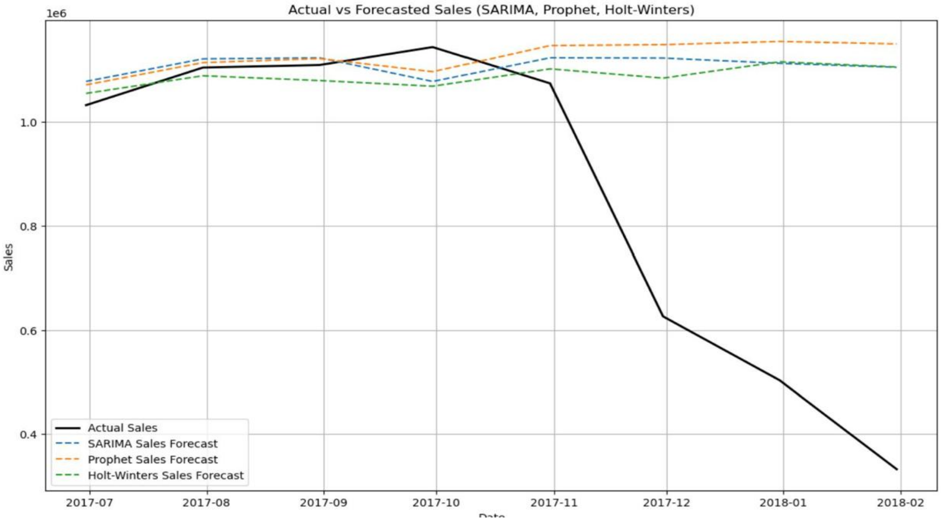


Figure 5.3: Actual vs Forecasted Sales Model

The **Actual vs Forecasted Sales** graph similarly shows that Holt-Winters outperformed the other models in capturing the trend and seasonality in sales data, leading to more reliable forecasts.

5.3 JUSTIFICATION OF MODEL SELECTION

The decision to select Holt-Winters as the final model was primarily driven by its superior performance in terms of predictive accuracy, as measured by the Root Mean Square Error (RMSE). RMSE provides a quantitative assessment of how well the model's forecasts match the actual values, with lower RMSE values indicating better performance.

Winters provides a high level of interpretability due to its reliance on exponential smoothing, which is intuitive and easy to explain. The model’s results are highly interpretable for stakeholders without a deep technical background. This is particularly important in a business context, where forecasting models need to not only provide accurate predictions but also be understandable and actionable.

5.4 INSIGHTS

The Holt-Winters Exponential Smoothing method was employed to generate a 2-year forecast for sales and profit. This method is particularly effective for time series data with both trend and seasonality, making it well-suited for the supply chain dataset used in this analysis.

5.4.1. Before that lets look into the Historical Sales Trend (2015-2018)

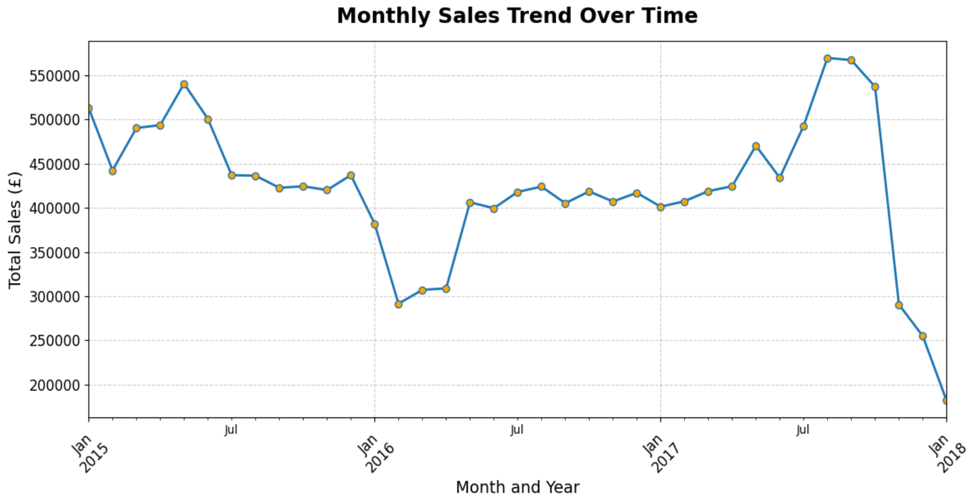


Figure 5.4: Monthly Sales Trend Over Years

The graph depicting the **Monthly Sales Trend Over Time** from January 2015 to early 2018 shows clear fluctuations, reflecting the cyclical nature of sales in the supply chain context. Here are some key observations:

5.4.1.1 Seasonal Peaks and Troughs:

i. From early 2015 to mid-2016, sales experienced noticeable peaks, with the highest reaching over £500,000 in mid-2015. This spike likely corresponds to seasonal demand or specific product launches, indicating a period of strong sales performance.

ii. A sharp decline is observed in early 2016, where sales drop significantly. The dip below £400,000 could indicate an off-season or external market factors impacting demand during this period.

5.4.1.2 Recovery and Growth (2016-2017):

i. Following the dip in 2016, sales recover gradually from mid-2016 to early 2017, with figures stabilizing around £400,000 to £450,000. This improvement may reflect an enhancement in market conditions or revitalized demand.

ii. In mid-2017, another strong surge in sales is recorded, reaching £550,000, likely due to seasonal promotions or increased consumer spending. This further growth suggests successful sales strategies or market expansion during this time.

5.4.1.2 2018 Data Caveat:

The significant decline seen at the start of 2018 should be interpreted with caution. As the data for 2018 is not fully available in the dataset, the sharp drop likely does not represent actual performance but rather missing data for the year. This should be considered in the overall analysis.

5.5 PROFIT FORECAST:

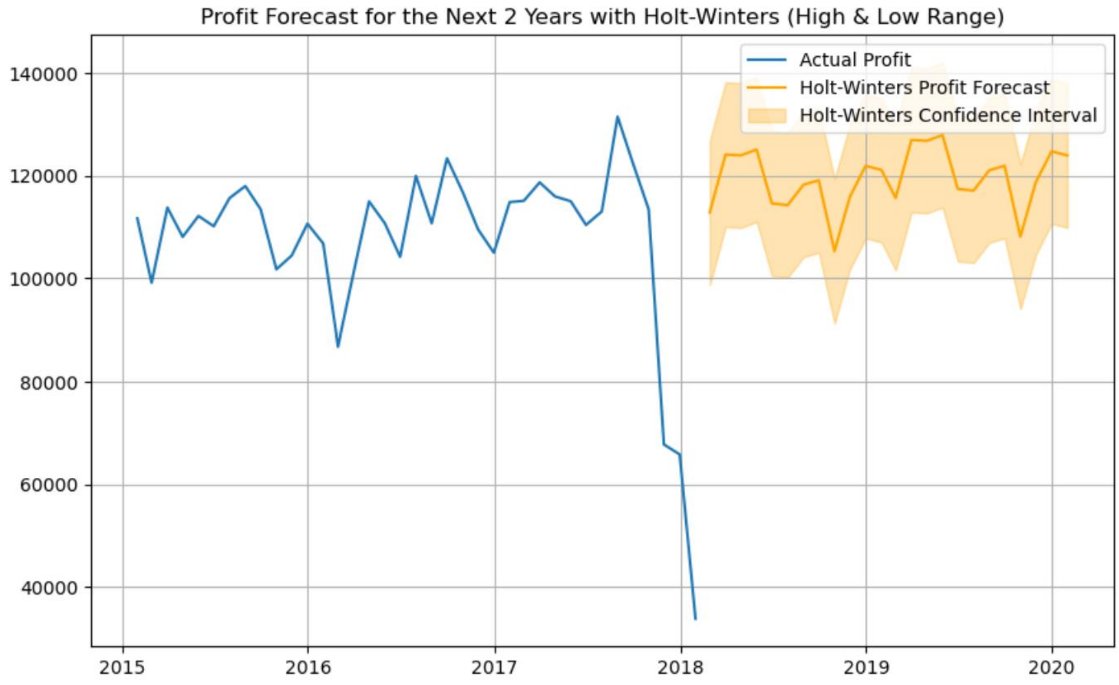


Figure 5.5: Profit Forecast

5.5.1 Future Outlook Based on the Holt-Winters Profit Forecast

The profit forecast for the next two years shows a steady and predictable pattern. This is largely driven by the model's ability to capture seasonal variations and trend components, both of which play a significant role in business profitability.

5.5.2 Sustained Growth with Fluctuations

The model projects a continuation of the historical trend, with periodic rises and falls in profit over the forecast period. However, the fluctuations are expected to be more moderate, with profits hovering between £120,000 and £140,000. These patterns reflect seasonal or cyclical factors, such as market demand fluctuations, promotional activities, or external economic conditions that could affect sales and overall profitability.

5.5.3 Confidence Intervals and Risk Management:

The confidence intervals provide a range of certainty for future profits. While the forecast anticipates consistent profit growth, the upper and lower bounds of the confidence interval suggest that unexpected external factors could impact the exact profit levels. These factors could include:

i. **Economic downturns or booms**: Significant global or regional economic changes could push profits outside the expected range.

ii. **Market dynamics**: Shifts in consumer demand, changes in supplier costs, or new market entrants could affect profit margins.

iii. **Operational changes**: Internal changes in supply chain efficiency or pricing strategies may also impact future profitability.

Given the narrow confidence range, it is likely that the business will experience stable profits, with minimal risk of major declines, assuming that external conditions remain relatively stable.

5.5.4 Strategic Forecast and Future Outlook

i. The forecasted trends allow businesses to plan strategically by preparing for seasonal demand shifts. Understanding when profits will peak helps align marketing, inventory, and sales strategies.
ii. Stable profit growth suggests that long-term investments in supply chain improvements or new product lines can yield predictable returns with minimal risk.
iii. Over the next two years, steady profit growth is expected with minor seasonal dips. This foresight enables companies to optimize performance during peak periods and mitigate slower periods, ensuring data-driven decision-making and long-term success.

5.6 SALES FORECAST

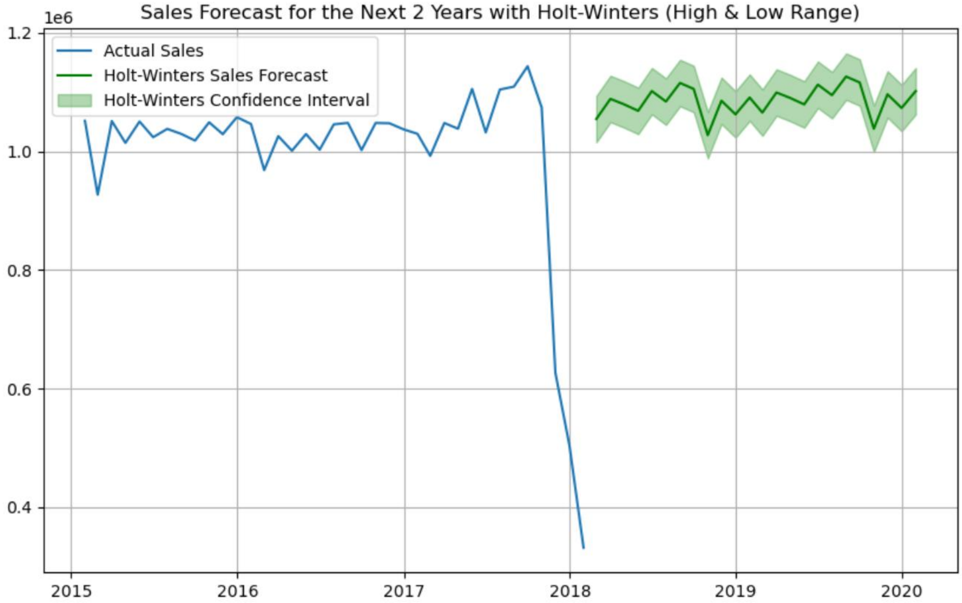


Figure 5.6: Sales Forecast

5.6.1 Sales Forecast Using Holt-Winters Method

The graph illustrates the sales forecast for the next two years, from 2018 to 2020, using the Holt-Winters

5.6.2 Sustained Sales with Fluctuations

The Holt-Winters model projects a continuation of the historical trend, with periodic rises and falls in sales over the forecast period. However, these fluctuations are expected to be moderate, with sales expected to stabilize between £1 million and £1.2 million.

5.6.3 Confidence Intervals and Risk Management

The confidence intervals offer a range of certainty for future sales, providing an upper and lower bound to account for potential variability.

i. **Economic Volatility:** A recession could result in decreased consumer purchasing power, reducing overall sales. Conversely, during an economic boom, consumer spending might increase, driving sales above the forecasted range.
ii. **Market Competition:** Shifts in customer behavior, such as preferences for eco-friendly or premium products, may disrupt established sales patterns and alter demand levels.
iii. **Operational Efficiency:** Internal changes within the company could also influence sales outcomes.Efficient supply chains enable faster deliveries and better stock management, leading to higher customer satisfaction and potential sales growth

5.6.3 Strategic Forecast and Future Outlook

i. The forecasted trends enable businesses to plan strategically by preparing for seasonal demand fluctuations. Understanding the peaks and troughs in sales allows companies to align their marketing efforts, inventory management, and sales strategies to capitalize on high-demand periods.
ii. The consistent sales projections suggest that long-term investments, such as expanding into new markets or developing additional product lines, can be pursued with minimal risk, as stable sales growth is expected.
iii. Over the next two years, the business can expect steady sales growth with some seasonal variations. This foresight allows the company to optimize its performance during peak periods and prepare for slower periods, ensuring data-driven decision-making and sustained success in the market.

CHAPTER 6

PREDICTION OF DELIVERY RISK

A user-friendly interface, HTML page was developed to predict late delivery risk in real-time, allowing supply chain managers and decision-makers to input key order details and receive immediate predictions. This tool, offers a proactive approach to managing potential delays by leveraging data-driven insights. By identifying high-risk deliveries, businesses can take primptive actions to mitigate disruptions and enhance operational efficiency.

6.1 IMPLEMENTATION:

We aimed to predict Late Delivery Risk based on various features from the supply chain dataset. We employed multiple classification models, including Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost.

6.2 DATA PRE-PROCESSING

Before building the model, we pre-processed the data by standardizing numerical columns and one-hot encoding categorical columns using a Column transformer. This ensures that all variables are in a suitable format for machine learning algorithms to process. Feature such as 'Customer Id', 'Order Id', 'Order Zipcode', and 'Shipping Delay', among others, were removed from the dataset to focus on variables that contribute directly to predicting late delivery risk. The remaining variables included customer segment, market, product details, shipping mode, and more.

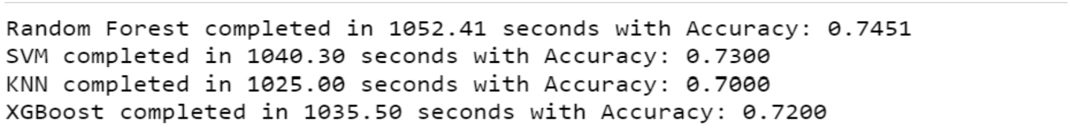
The dataset is divided into 80% for training and 20% for testing using the train_test_split function. The model is trained on the training data and then tested on the unseen test data, ensuring that the evaluation reflects how well the model generalizes to new information.

6.3 MODEL EVALUATION

i. A **Pipeline** is established for each model to streamline the workflow. This pipeline automates the process by applying the necessary data preprocessing steps, such as **standardization** and **encoding**, before fitting the specific machine learning model to the training data.

ii. During the **Training and Prediction** phase, each model is trained on the provided dataset, using the training data. After training, the models make predictions on the test data, and these predictions are compared with actual values to determine their accuracy.

iii. Finally, the **Performance Reporting** phase includes capturing both the time taken for each model to train and the respective accuracy score. This allows for a comprehensive comparison of each model's performance, balancing accuracy and computational efficiency to guide the selection of the most suitable model for the task.



```
Random Forest completed in 1052.41 seconds with Accuracy: 0.7451
SVM completed in 1040.30 seconds with Accuracy: 0.7300
KNN completed in 1025.00 seconds with Accuracy: 0.7000
XGBoost completed in 1035.50 seconds with Accuracy: 0.7200
```

Figure 6.1: Accuracy

From the above figure we can see that Random Forest achieved the highest accuracy at 74.51%, making it the top choice despite taking slightly longer to train. SVM and XGBoost followed with 73.00% and 72.00% accuracy, respectively, while KNN was the fastest but least accurate at 70.00%. Overall, Random Forest offers the best balance between accuracy and training time, with SVM and XGBoost being viable alternatives.

After training the **Random Forest Classifier**, the performance was further assessed using a **Confusion Matrix** and **Classification Report**. The confusion matrix provided a detailed view of the model's predictions as shown below,

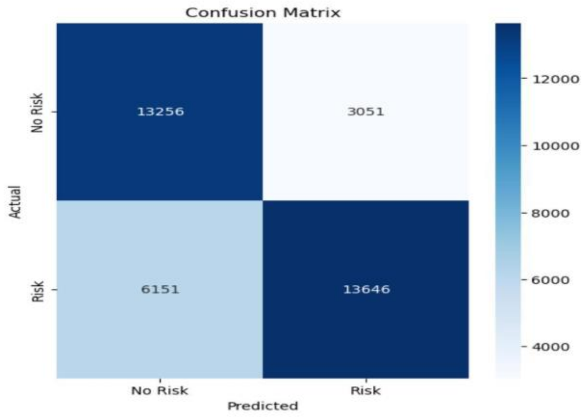
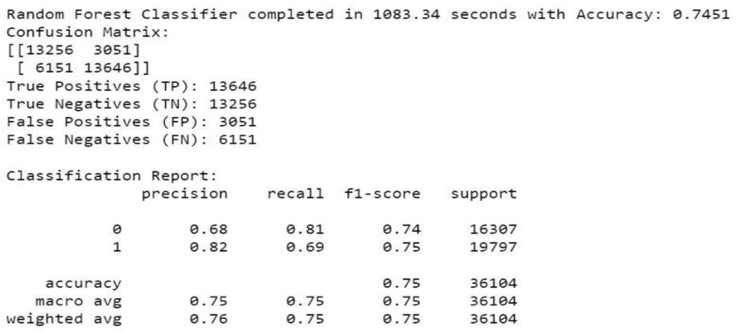


Figure 6.2: Confusion Matrix

The classification report offered additional metrics, highlighting that the model had a precision of 82% for predicting late delivery risk, indicating that 82% of the orders predicted as risky were correctly identified. The recall for predicting risk was 69%, meaning that the model was able to capture 69% of all actual risky orders. The overall F1-score for the model was 75%, balancing both precision and recall.

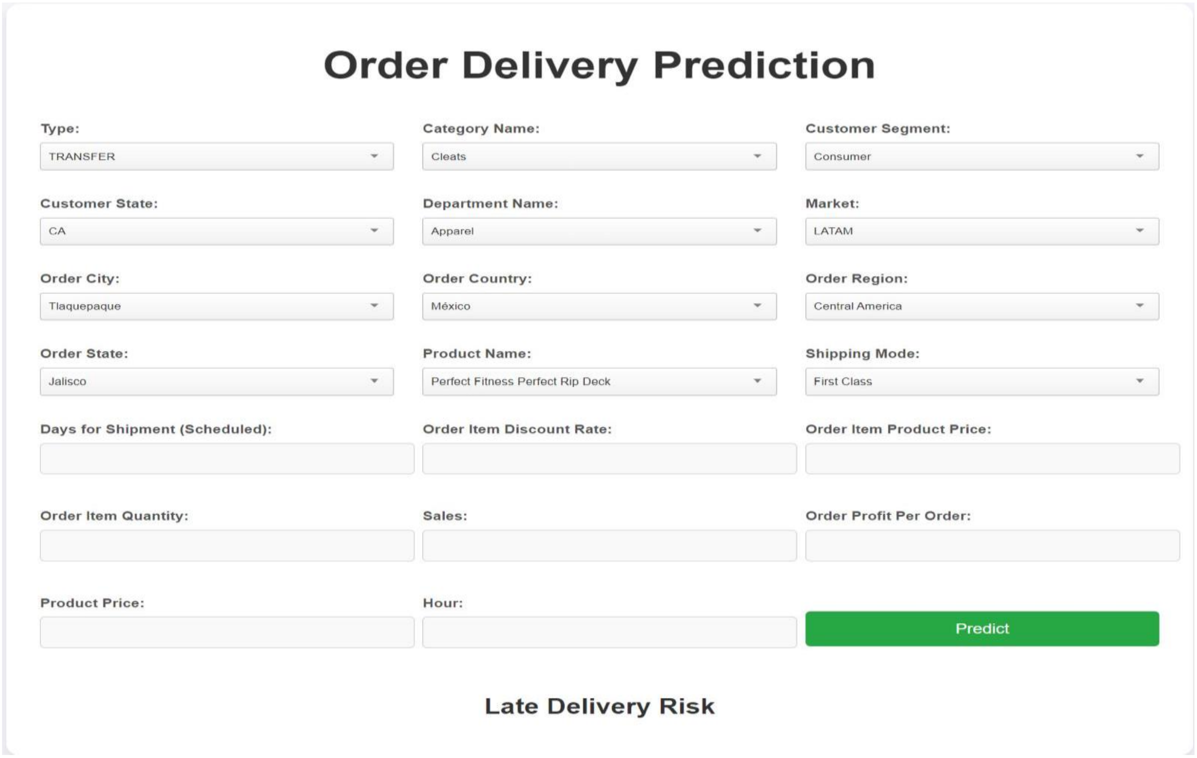


```
Random Forest Classifier completed in 1083.34 seconds with Accuracy: 0.7451
Confusion Matrix:
[[13256  3051]
 [ 6151 13646]]
True Positives (TP): 13646
True Negatives (TN): 13256
False Positives (FP): 3051
False Negatives (FN): 6151

Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.81      0.74     16307
           1       0.82      0.69      0.75     19797

    accuracy                           0.75     36104
   macro avg       0.75      0.75      0.75     36104
weighted avg       0.76      0.75      0.75     36104
```

Figure 6.3: Random Forest Classification Report

The HTML page, developed using Flask, serves as an intuitive interface for predicting late delivery risk in real-time. By inputting key details such as product category, shipping mode, customer segment, and location, we can quickly receive predictions based on the Random Forest Classifier, the most accurate model identified during evaluation.

This tool provides immediate, actionable insights, allowing supply chain managers to proactively mitigate delivery risks by adjusting shipping strategies or prioritizing resources. The simple interface ensures that critical decisions can be made efficiently, improving overall supply chain performance and customer satisfaction.



Order Delivery Prediction

Type: TRANSFER

Category Name: Cleats

Customer Segment: Consumer

Customer State: CA

Department Name: Apparel

Market: LATAM

Order City: Tlaquepaque

Order Country: México

Order Region: Central America

Order State: Jalisco

Product Name: Perfect Fitness Perfect Rip Deck

Shipping Mode: First Class

Days for Shipment (Scheduled):

Order Item Discount Rate:

Order Item Product Price:

Order Item Quantity:

Sales:

Order Profit Per Order:

Product Price:

Hour:

Late Delivery Risk

Figure 6.4: HTML page using Flask

CHAPTER 7

LIMITATIONS

a) FEATURE SET LIMITATION

The dataset lacks key variables like real-time shipping conditions (e.g., weather, traffic), supplier performance metrics, and processing times, which could enhance the model's predictive accuracy. The absence of these features may restrict the model's ability to capture more complex, real-world delivery dynamics.

b) DATA QUALITY

Missing values and inconsistent entries were handled during preprocessing, but the removal of certain records may have led to a loss of valuable information. This could have impacted the model's ability to learn from the full complexity of the data, particularly for features like shipping delays.

c) MODEL GENERALIZATION

The model was trained on historical data specific to certain regions, markets, and customer segments. As a result, its ability to generalize to new markets or adapt to unforeseen supply chain disruptions may be limited. The current model may underperform in environments that differ significantly from the training data.

d) MODEL INTERPRETABILITY

While the Random Forest Classifier demonstrated high accuracy, it is inherently a black-box model (Rigatti, 2017). This reduces the transparency of decision-making, making it harder for end-users (e.g., supply chain managers) to understand why certain predictions are made. This could hinder adoption and trust in real-world applications.

e) DEPLOYMENT SCALABILITY

The current deployment using a Flask-based web interface may face performance bottlenecks in large-scale applications. As the data volume and the number of users increase, scalability challenges could arise, impacting the system's ability to deliver real-time predictions in high-demand environments.

f) STATIC DATA SOURCE

The dataset used was static, meaning it reflects historical data and does not capture real-time changes in the supply chain. As a result, the model cannot adapt to dynamic market conditions or real-time disruptions, limiting its applicability in fast-changing environments. Incorporating real-time data streaming could improve responsiveness.

CHAPTER 8

CONCLUSION

This dissertation has demonstrated the significant potential of machine learning, particularly the **Random Forest model**, in predicting late delivery risks in supply chain management. By building a robust model that leverages various features such as customer segments, product details, and shipping modes, this research has shown how data-driven approaches can enhance operational efficiency and reduce uncertainties in logistics.

Applications:

i. **Supply Chain Managers**: The ability to predict late deliveries enables managers to proactively mitigate delays by making informed decisions regarding shipping methods, order prioritization, and resource allocation. This directly leads to improvements in customer satisfaction, reduced operational costs, and enhanced overall performance.

ii. **Logistics and Operations Teams**: With the prediction model deployed in a **Flask-based web interface**, logistics teams now have a practical tool that allows them to assess delivery risks in real time. This tool empowers them to adjust strategies on the fly, ensuring more accurate delivery estimates and fewer disruptions in the supply chain process.

iii. **Business Decision Makers**: For organizations, especially those dealing with large volumes of orders, the ability to forecast delivery risks provides an opportunity to optimize resources more effectively. This tool allows businesses to allocate resources, such as vehicles or warehousing, more efficiently, and build more reliable delivery networks.

iv. **Customers**: One of the biggest beneficiaries of this system is the end customer. By reducing the number of late deliveries and ensuring better fulfillment of orders, customers experience higher satisfaction, fewer delays, and an overall better service experience.

Future work:

i. **Demand Forecasting**: Predicting future customer demand more accurately, enabling businesses to optimize production schedules and inventory levels.

ii. **Inventory Management**: Implementing models that forecast stock needs to reduce both overstock and stockouts, improving cost efficiency and minimizing waste.

iii. **Supplier Relationship Management**: Leveraging machine learning to evaluate supplier performance, predict potential delays, and optimize procurement strategies.

This dissertation, therefore, not only addresses the current challenges of late delivery in supply chains but also lays the groundwork for further innovation in supply chain management. By employing advanced machine learning techniques and providing an accessible tool for operational use, this project serves as a significant step towards smarter, more efficient, and customer-centric supply chains.

As businesses continue to adopt data-driven methods, the framework and methodologies developed in this research will be instrumental in helping organizations stay competitive in an increasingly complex logistics environment. The ultimate goal is to ensure that supply chains are not just reactive but predictive and proactive, thus contributing to a more resilient and responsive supply chain ecosystem.

CHAPTER 9

PERSONAL REFLECTION

My experience on developing the delivery risk prediction tool, made me realize how much this project has contributed to both my technical growth and practical understanding of machine learning in a real-world context. Throughout the process, I gained a deeper appreciation for balancing model performance with user accessibility a challenge that pushed me out of my technical comfort zone.

Initially, my focus was purely on achieving the best model accuracy. I spent considerable time selecting and comparing models like Random Forest, SVM, KNN, and XGBoost, using metrics such as precision, recall, and F1-score. However, I quickly realized that even the most accurate model has limited value if it can’t be integrated into a tool that users can easily interpret and apply. This was a significant turning point for me; it made me appreciate how important usability is in data science projects.

Developing the HTML page using Flask was one of the most rewarding aspects. I had to think from the perspective of end-users—people who are not necessarily familiar with machine learning but need quick, actionable insights. Creating a simple, intuitive interface that allowed users to input data and get immediate predictions gave me a new skill set in web development and user-centric design, something I had not focused on previously. This process also made me reflect on the importance of bridging the gap between technical implementation and real-world decision-making.

Another key realization was the impact of feedback loops. Throughout this project, I had to constantly evaluate the performance of the models and re-adjust my approach based on the results. This iterative process taught me the value of patience and adaptability, as I learned that machine learning isn't about achieving perfection in one go but involves continuous refinement.

In conclusion, this project gave me a comprehensive view of how data science can directly contribute to operational efficiency in industries like supply chain management. More importantly, it underscored the importance of thinking not just as a data scientist, but as a problem-solver who must make complex solutions accessible and useful to a broad audience. This reflection has deepened my commitment to building tools that are both technically sound and easy to implement in real-world applications.

CHAPTER 10

REFERENCES

1. *DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS*. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/shashwatwork/dataco-smart-supply-chain-for-big-data-analysis

2. Abaku, E. A., Edunjobi, T. E., & Odimarha, A. C. (2024). Theoretical Approaches to AI in Supply Chain optimization: Pathways to Efficiency and Resilience. *International Journal of Science and Technology Research Archive*, *6*(1), 092-107. https://doi.org/10.53771/ijstra.2024.6.1.0033
3. Tirkolaee, E. B., Sadeghi, S., Mooseloo, F. M., Vandchali, H. R., & Aeini, S. (2021). Application of Machine Learning in Supply Chain Management: A Comprehensive Overview of the Main Areas. *Mathematical Problems in Engineering*, *2021*(1), 1–14. https://doi.org/10.1155/2021/1476043
4. Kilimci, Z. H., Akyuz, A. O., Uysal, M., Akyokus, S., Uysal, M. O., Atak Bulbul, B., & Ekmis, M. A. (2019). An Improved Demand Forecasting Model Using Deep Learning Approach and Proposed Decision Integration Strategy for Supply Chain. *Complexity*, *2019*, 1–15. https://doi.org/10.1155/2019/9067367
5. Valipour, M. (2015). Long-term runoff study using SARIMA and ARIMA models in the United States. *Meteorological Applications*, *22*(3), 592–598. https://doi.org/10.1002/met.1491
6. Ali, Md. R., Nipu, S. Md. A., & Khan, S. A. (2023). A decision support system for classifying supplier selection criteria using machine learning and random forest approach. *Decision Analytics Journal*, *7*, 100238. https://doi.org/10.1016/j.dajour.2023.100238
7. Kalekar, P. S. (2004). *Time series Forecasting using Holt-Winters Exponential Smoothing*. https://c.mql5.com/forextsd/forum/69/exponentialsmoothing.pdf
8. Idris, N., Mohd, F., & Palaniappan Shamala. (2021). A Generic Review of Web Technology: DJango and Flask. *International Journal of Advanced Science Computing and Engineering*, *2*(1), 34–40. https://doi.org/10.62527/ijasce.2.1.29
9. Moffat, I., & Akpan, E. (n.d.). *Time Series Forecasting: A Tool for Out -Sample Model Selection and Evaluation*. https://doi.org/10.5251/ajsir.2014.5.6.185.194
10. Shine, G., & Basak, S. (n.d.). *Sales Prediction with Time Series Modeling*. Retrieved September 18, 2024, from https://cs229.stanford.edu/proj2015/219_report.pdf
11. *View of Optimizing Supply Chain Management through Artificial Intelligence: Techniques for Predictive Maintenance, Demand Forecasting, and Inventory Optimization | Journal of AI-Assisted Scientific Discovery*. (2024). Scienceacadpress.com. https://scienceacadpress.com/index.php/jaasd/article/view/58/53
12. Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, *184*(3), 1140–1154. https://doi.org/10.1016/j.ejor.2006.12.004
13. Van Calster, T., Baesens, B., & Lemahieu, W. (2017). ProfARIMA: A profit-driven order identification algorithm for ARIMA models in sales forecasting. *Applied Soft Computing*, *60*, 775–785. https://doi.org/10.1016/j.asoc.2017.02.011
14. Wikipedia Contributors. (2019, July 30). *Time series*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Time_series
15. GeeksforGeeks. (2018, November 13). *K-Nearest Neighbours - GeeksforGeeks*. GeeksforGeeks. https://www.geeksforgeeks.org/k-nearest-neighbours/
16. Menden, M. P., Iorio, F., Garnett, M., McDermott, U., Benes, C. H., Ballester, P. J., & Saez-Rodriguez, J. (2013). Machine Learning Prediction of Cancer Cell Sensitivity to Drugs Based on Genomic and Chemical Properties. *PLoS ONE*, *8*(4), e61318. https://doi.org/10.1371/journal.pone.0061318
17. Mahesh, B. (2018). Machine Learning Algorithms -A Review. *International Journal of Science and Research (IJSR) ResearchGate Impact Factor*, *9*(1). https://doi.org/10.21275/ART20203995
18. Satrio, C. B. A., Darmawan, W., Nadia, B. U., & Hanafiah, N. (2021). Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. *Procedia Computer Science*, *179*, 524–532. https://doi.org/10.1016/j.procs.2021.01.036

19. SolarWinds. (2019, December 15). *Holt-Winters Forecasting Simplified*. Orange Matter. https://orangematter.solarwinds.com/2019/12/15/holt-winters-forecasting-simplified/
20. Jakkula, V. (2006). *Tutorial on Support Vector Machine (SVM)*. https://course.khoury.northeastern.edu/cs5100f11/resources/jakkula.pdf

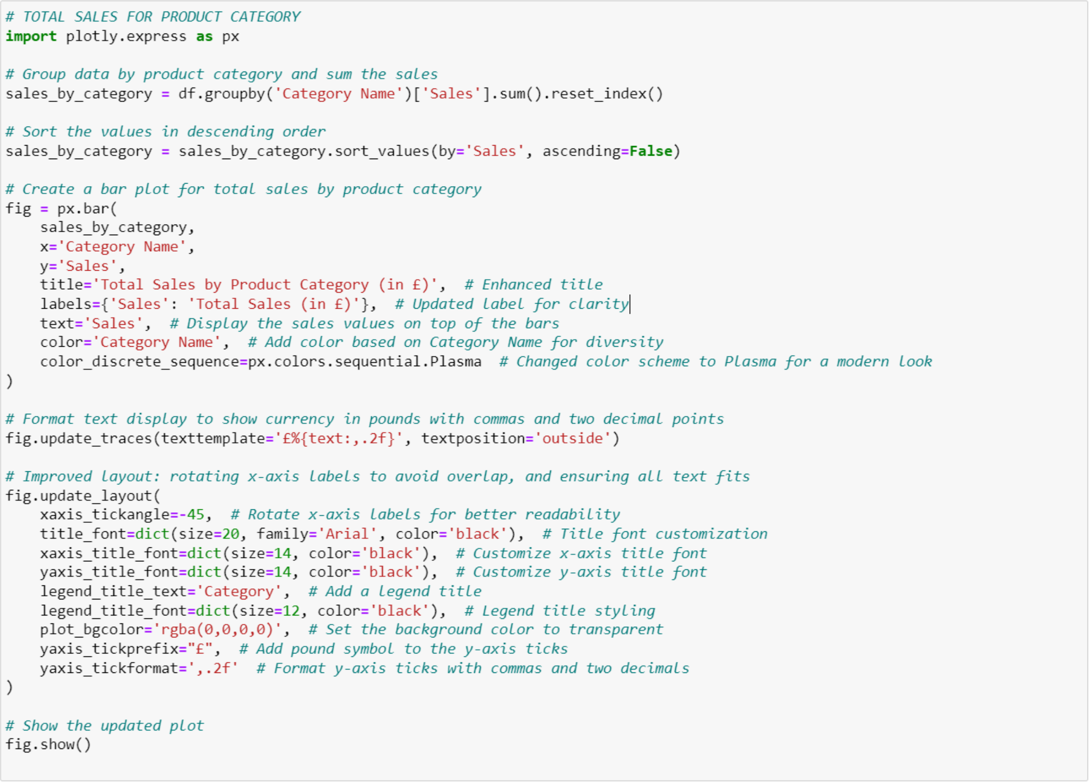
21. Gatto, J. (2021, March 23). *Support Vector Machine Math for people who don't care about optimization*. Medium; Medium. https://joseph-gatto.medium.com/support-vector-machines-svms-for-people-who-dont-care-about-optimization-77873fa49bca
22. Peterson, L. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883. https://doi.org/10.4249/scholarpedia.1883
23. Kumar, P., & Jain, N. K. (2022). Surface roughness prediction in micro-plasma transferred arc metal additive manufacturing process using K-nearest neighbors algorithm. *The International Journal of Advanced Manufacturing Technology*, *119*(5-6), 2985–2997. https://doi.org/10.1007/s00170-021-08639-2
24. Chen, T., & Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 785–794.
25. *What is XGBoost? An Introduction to XGBoost Algorithm in Machine Learning | Simplilearn*. (2022, November 22). Simplilearn.com. https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article#:~:text=XGBoost%20is%20a%20robust%20machine
26. Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324
27. Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? *Geoscientific Model Development Discussions*, *7*(1), 1525–1534. https://doi.org/10.5194/gmdd-7-1525-2014
28. Glen, S. (2022). *RMSE: Root Mean Square Error*. Statistics How To. https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/
29. *RMSE Guide | How to Evaluate & Improve Your ML Models*. (2023, January 31). Greelance. https://greelance.com/ultimate-rmse-guide-how-to-evaluate-improve-machine-learning-models/#:~:text=Model%20selection%3A%20RMSE%20can%20be
30. Rigatti, S. J. (2017). Random Forest. *Journal of Insurance Medicine*, *47*(1), 31–39. https://meridian.allenpress.com/jim/article/47/1/31/131479/Random-Forest
31. Bevans, R. (2023, January 18). *Understanding Confidence Intervals | Easy Examples & Formulas*. Scribbr. https://www.scribbr.co.uk/stats/confidence-interval-meaning/#:~:text=A%20confidence%20interval%20is%20the

CHAPTER 11

APPENDIX 1

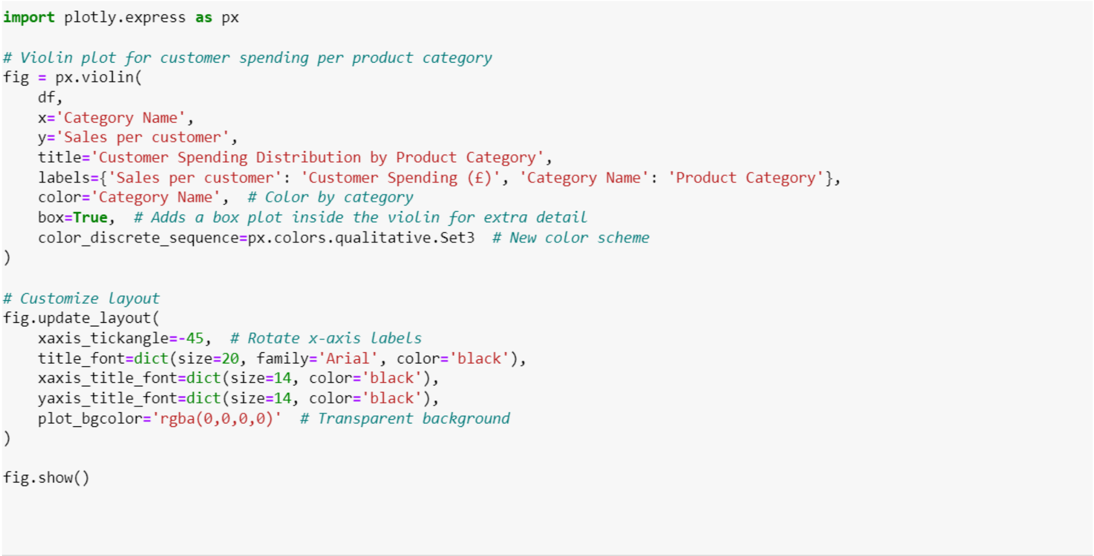
Code Snippets

1. Total Sales for Product Category



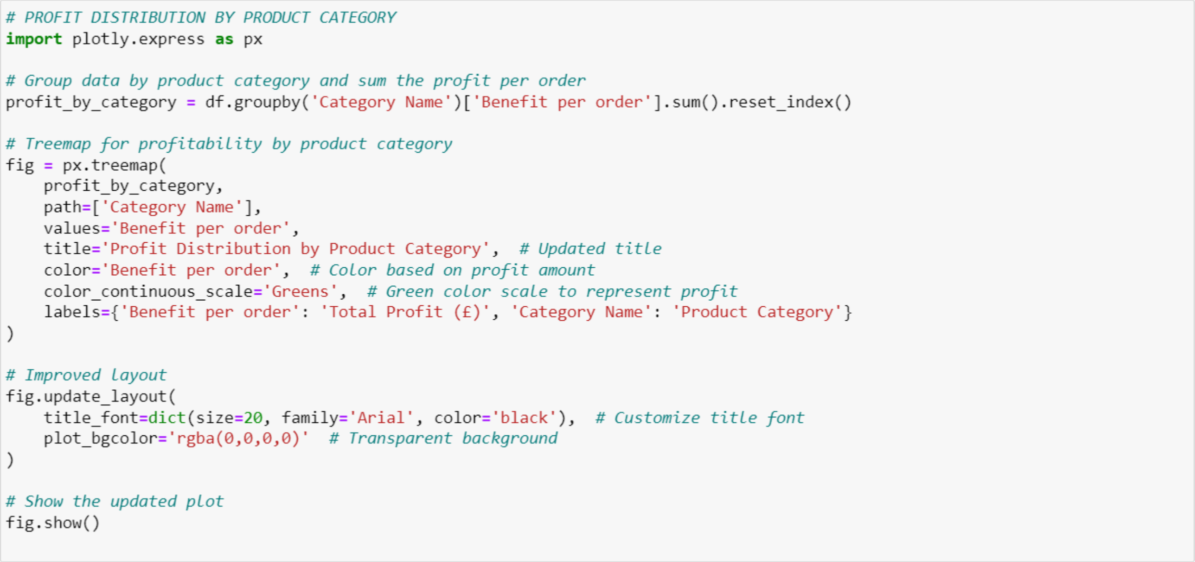
```
# TOTAL SALES FOR PRODUCT CATEGORY
import plotly.express as px

# Group data by product category and sum the sales
sales_by_category = df.groupby('Category Name')['Sales'].sum().reset_index()

# Sort the values in descending order
sales_by_category = sales_by_category.sort_values(by='Sales', ascending=False)

# Create a bar plot for total sales by product category
fig = px.bar(
    sales_by_category,
    x='Category Name',
    y='Sales',
    title='Total Sales by Product Category (in £)',  # Enhanced title
    labels={'Sales': 'Total Sales (in £)'},  # Updated label for clarity
    text='Sales',  # Display the sales values on top of the bars
    color='Category Name',  # Add color based on Category Name for diversity
    color_discrete_sequence=px.colors.sequential.Plasma  # Changed color scheme to Plasma for a modern look
)

# Format text display to show currency in pounds with commas and two decimal points
fig.update_traces(texttemplate='£%{text:,.2f}', textposition='outside')

# Improved layout: rotating x-axis labels to avoid overlap, and ensuring all text fits
fig.update_layout(
    xaxis_tickangle=-45,  # Rotate x-axis labels for better readability
    title_font=dict(size=20, family='Arial', color='black'),  # Title font customization
    xaxis_title_font=dict(size=14, color='black'),  # Customize x-axis title font
    yaxis_title_font=dict(size=14, color='black'),  # Customize y-axis title font
    legend_title_text='Category',  # Add a legend title
    legend_title_font=dict(size=12, color='black'),  # Legend title styling
    plot_bgcolor='rgba(0,0,0,0)',  # Set the background color to transparent
    yaxis_tickprefix="£",  # Add pound symbol to the y-axis ticks
    yaxis_tickformat=',.2f'  # Format y-axis ticks with commas and two decimals
)

# Show the updated plot
fig.show()
```

2. Customer Spending for Product Category



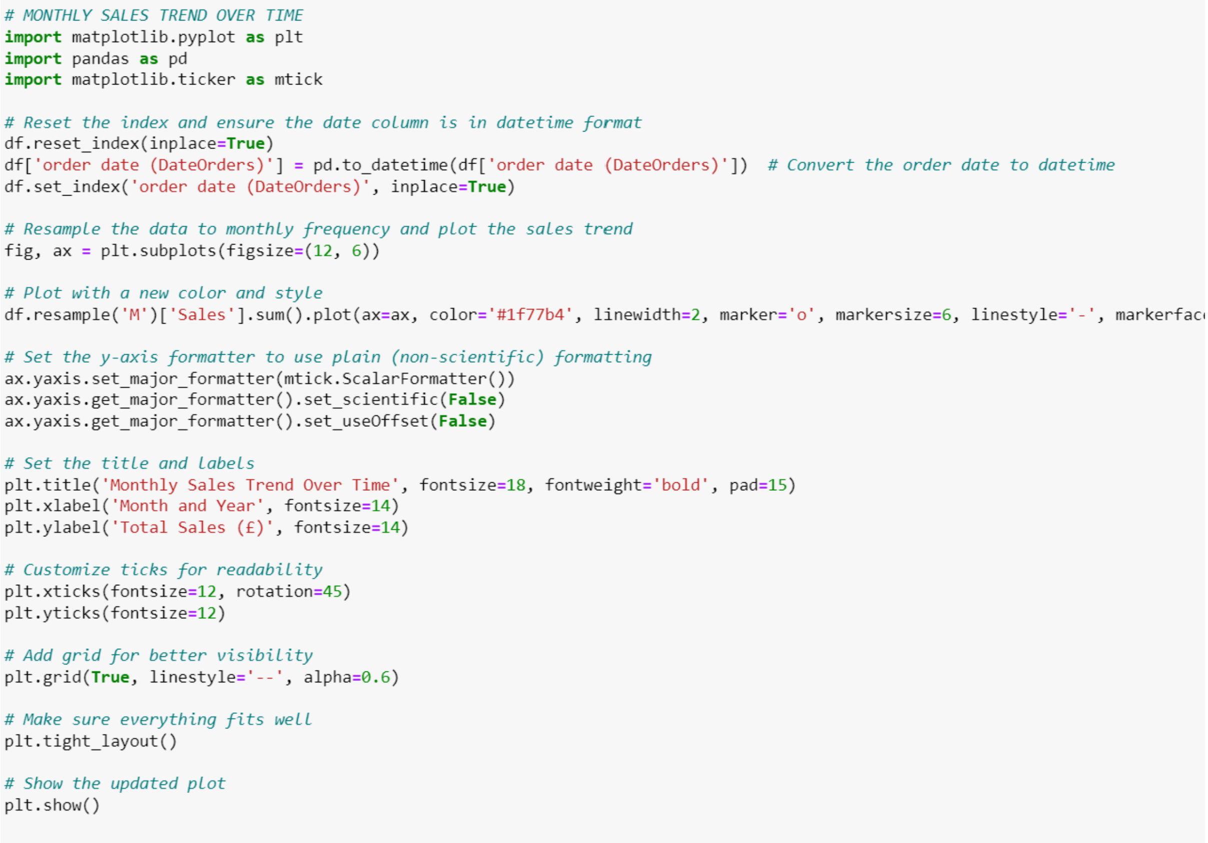
```
import plotly.express as px

# Violin plot for customer spending per product category
fig = px.violin(
    df,
    x='Category Name',
    y='Sales per customer',
    title='Customer Spending Distribution by Product Category',
    labels={'Sales per customer': 'Customer Spending (£)', 'Category Name': 'Product Category'},
    color='Category Name',  # Color by category
    box=True,  # Adds a box plot inside the violin for extra detail
    color_discrete_sequence=px.colors.qualitative.Set3  # New color scheme
)

# Customize layout
fig.update_layout(
    xaxis_tickangle=-45,  # Rotate x-axis labels
    title_font=dict(size=20, family='Arial', color='black'),
    xaxis_title_font=dict(size=14, color='black'),
    yaxis_title_font=dict(size=14, color='black'),
    plot_bgcolor='rgba(0,0,0,0)'  # Transparent background
)

fig.show()
```

3. Profitability Distribution by Product Category



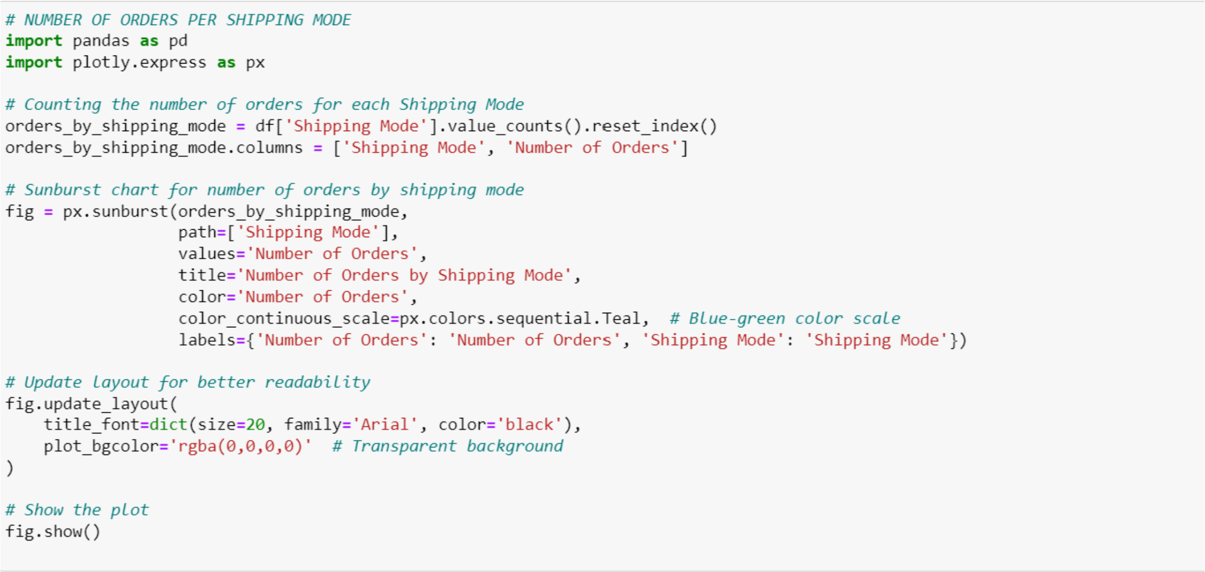
```python
# PROFIT DISTRIBUTION BY PRODUCT CATEGORY
import plotly.express as px

# Group data by product category and sum the profit per order
profit_by_category = df.groupby('Category Name')['Benefit per order'].sum().reset_index()

# Treemap for profitability by product category
fig = px.treemap(
    profit_by_category,
    path=['Category Name'],
    values='Benefit per order',
    title='Profit Distribution by Product Category',  # Updated title
    color='Benefit per order',  # Color based on profit amount
    color_continuous_scale='Greens',  # Green color scale to represent profit
    labels={'Benefit per order': 'Total Profit (£)', 'Category Name': 'Product Category'}
)

# Improved layout
fig.update_layout(
    title_font=dict(size=20, family='Arial', color='black'),  # Customize title font
    plot_bgcolor='rgba(0,0,0,0)'  # Transparent background
)

# Show the updated plot
fig.show()
```

4. Monthly Sales Trend



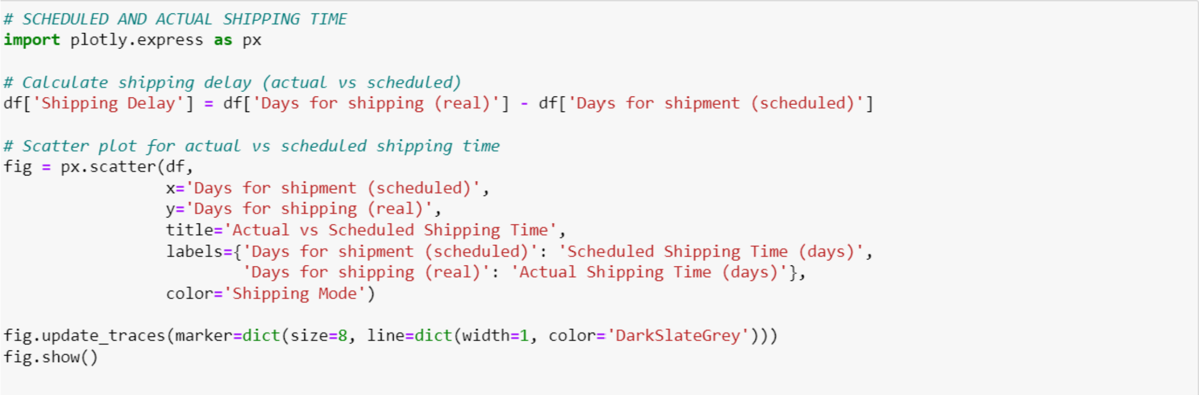
```python
# MONTHLY SALES TREND OVER TIME
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.ticker as mtick

# Reset the index and ensure the date column is in datetime format
df.reset_index(inplace=True)
df['order date (DateOrders)'] = pd.to_datetime(df['order date (DateOrders)'])  # Convert the order date to datetime
df.set_index('order date (DateOrders)', inplace=True)

# Resample the data to monthly frequency and plot the sales trend
fig, ax = plt.subplots(figsize=(12, 6))

# Plot with a new color and style
df.resample('M')['Sales'].sum().plot(ax=ax, color='#1f77b4', linewidth=2, marker='o', markersize=6, linestyle='-', markerfac

# Set the y-axis formatter to use plain (non-scientific) formatting
ax.yaxis.set_major_formatter(mtick.ScalarFormatter())
ax.yaxis.get_major_formatter().set_scientific(False)
ax.yaxis.get_major_formatter().set_useOffset(False)

# Set the title and labels
plt.title('Monthly Sales Trend Over Time', fontsize=18, fontweight='bold', pad=15)
plt.xlabel('Month and Year', fontsize=14)
plt.ylabel('Total Sales (£)', fontsize=14)

# Customize ticks for readability
plt.xticks(fontsize=12, rotation=45)
plt.yticks(fontsize=12)

# Add grid for better visibility
plt.grid(True, linestyle='--', alpha=0.6)

# Make sure everything fits well
plt.tight_layout()

# Show the updated plot
plt.show()
```

5. Orders by Shipping Mode



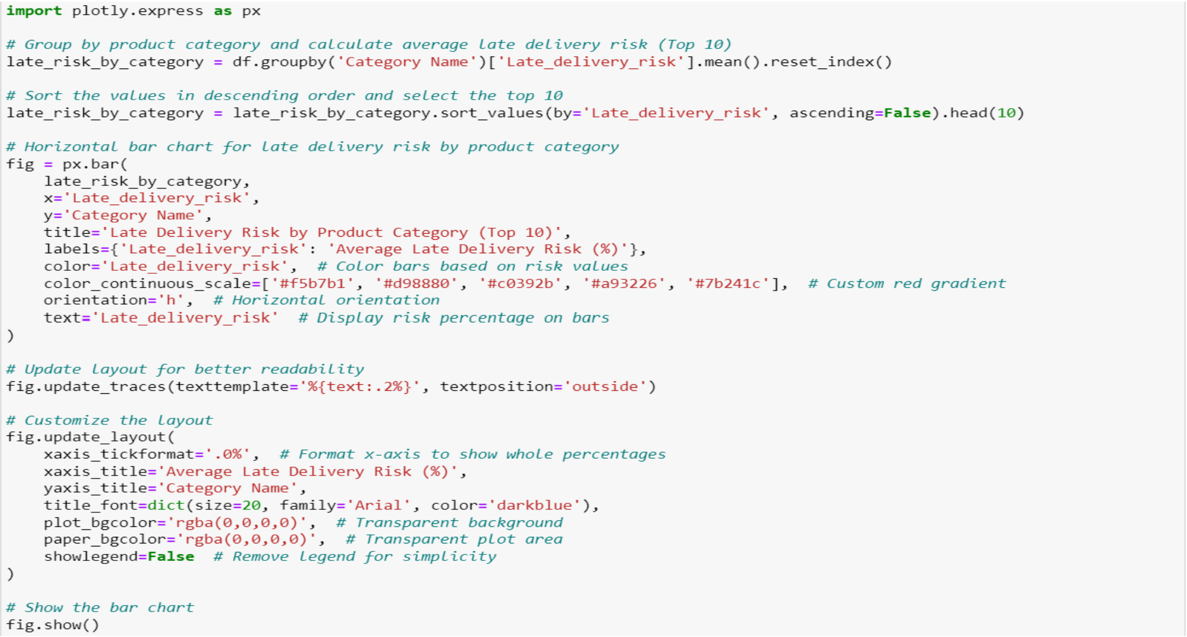
```python
# NUMBER OF ORDERS PER SHIPPING MODE
import pandas as pd
import plotly.express as px

# Counting the number of orders for each Shipping Mode
orders_by_shipping_mode = df['Shipping Mode'].value_counts().reset_index()
orders_by_shipping_mode.columns = ['Shipping Mode', 'Number of Orders']

# Sunburst chart for number of orders by shipping mode
fig = px.sunburst(orders_by_shipping_mode,
                  path=['Shipping Mode'],
                  values='Number of Orders',
                  title='Number of Orders by Shipping Mode',
                  color='Number of Orders',
                  color_continuous_scale=px.colors.sequential.Teal,  # Blue-green color scale
                  labels={'Number of Orders': 'Number of Orders', 'Shipping Mode': 'Shipping Mode'})

# Update layout for better readability
fig.update_layout(
    title_font=dict(size=20, family='Arial', color='black'),
    plot_bgcolor='rgba(0,0,0,0)'  # Transparent background
)

# Show the plot
fig.show()
```

6. Scheduled vs Actual Shipping Mode



```python
# SCHEDULED AND ACTUAL SHIPPING TIME
import plotly.express as px

# Calculate shipping delay (actual vs scheduled)
df['Shipping Delay'] = df['Days for shipping (real)'] - df['Days for shipment (scheduled)']

# Scatter plot for actual vs scheduled shipping time
fig = px.scatter(df,
                 x='Days for shipment (scheduled)',
                 y='Days for shipping (real)',
                 title='Actual vs Scheduled Shipping Time',
                 labels={'Days for shipment (scheduled)': 'Scheduled Shipping Time (days)',
                         'Days for shipping (real)': 'Actual Shipping Time (days)'},
                 color='Shipping Mode')

fig.update_traces(marker=dict(size=8, line=dict(width=1, color='DarkSlateGrey')))
fig.show()
```

7. Late Delivery Risk for Top 10 Category



```
import plotly.express as px

# Group by product category and calculate average late delivery risk (Top 10)
late_risk_by_category = df.groupby('Category Name')['Late_delivery_risk'].mean().reset_index()

# Sort the values in descending order and select the top 10
late_risk_by_category = late_risk_by_category.sort_values(by='Late_delivery_risk', ascending=False).head(10)

# Horizontal bar chart for late delivery risk by product category
fig = px.bar(
    late_risk_by_category,
    x='Late_delivery_risk',
    y='Category Name',
    title='Late Delivery Risk by Product Category (Top 10)',
    labels={'Late_delivery_risk': 'Average Late Delivery Risk (%)'},
    color='Late_delivery_risk',  # Color bars based on risk values
    color_continuous_scale=['#f5b7b1', '#d98880', '#c0392b', '#a93226', '#7b241c'],  # Custom red gradient
    orientation='h',  # Horizontal orientation
    text='Late_delivery_risk'  # Display risk percentage on bars
)

# Update layout for better readability
fig.update_traces(texttemplate='%{text:.2%}', textposition='outside')

# Customize the layout
fig.update_layout(
    xaxis_tickformat='.0%',  # Format x-axis to show whole percentages
    xaxis_title='Average Late Delivery Risk (%)',
    yaxis_title='Category Name',
    title_font=dict(size=20, family='Arial', color='darkblue'),
    plot_bgcolor='rgba(0,0,0,0)',  # Transparent background
    paper_bgcolor='rgba(0,0,0,0)',  # Transparent plot area
    showlegend=False  # Remove legend for simplicity
)

# Show the bar chart
fig.show()
```

APPENDIX 2

RESEARCH PROPOSAL

Artificial Intelligence in Supplier Relationship Management

Introduction

Supplier Relationship Management (SRM) is a crucial aspect of today’s highly interconnected and competitive supply chains. It ensures efficiency, cost-effectiveness, and resilience. However, traditional SRM methods often rely on manual processes and historical data, which are not enough to manage the growing complexity and speed of modern supply chains. With increasing data and the dynamic nature of global markets, there’s a clear need for smarter, faster solutions.

This is where Artificial Intelligence (AI) comes into play. By integrating AI into SRM, companies can enhance decision-making, predict potential risks, and develop more strategic partnerships with suppliers. This proposal seeks to explore how AI can transform SRM to better meet the demands of today's supply chains.

Why This Research Matters

The integration of AI into SRM is not just a trend but a necessity. Here are a few reasons why:

1. Managing Complexity: Modern supply chains generate massive amounts of data. Traditional methods struggle to analyze this effectively, leading to missed opportunities and risks.

2. Predictive Power: AI can predict potential supplier risks and performance issues before they happen, allowing companies to take proactive steps.

3. Optimization: AI can optimize supplier selection, helping businesses make more informed decisions on who to partner with and how to allocate resources.

4. Strategic Insights: AI doesn’t just analyze; it provides deep insights into supplier relationships, allowing companies to make better long-term strategic decisions.

The goal of this research is to fill the gaps left by traditional SRM methods by building an AI-powered SRM framework that enhances supplier relationships and improves overall supply chain performance.

Research Aims and Objectives

The primary aim of this project is to develop an AI-driven SRM framework that helps companies manage their suppliers more effectively. The specific objectives are:

1. **Developing the Framework:** To build a comprehensive AI-based SRM model tailored to modern supply chains.
2. **Evaluating Supplier Performance:** Using AI to assess supplier performance by analyzing historical data and making predictions for future behavior.
3. **Managing Risks:** Creating AI algorithms that can identify and mitigate risks related to suppliers, such as delays or quality issues.
4. **Optimizing Supplier Selection:** Implementing AI models to make better decisions about which suppliers to choose and how to forecast demand.
5. **Validating Effectiveness:** Measuring how well the AI framework performs compared to traditional SRM methods.

Key Research Questions

The project will aim to answer the following questions:

1. How can AI be leveraged to better evaluate supplier performance using both past data and future predictions?
2. Which AI-driven algorithms are most effective in identifying and mitigating supplier-related risks?
3. How can AI models help businesses optimize supplier selection and demand forecasting?
4. How does an AI-based SRM framework improve overall supply chain outcomes compared to traditional methods?

Problem Statement

Today’s supply chains are more complex than ever. With vast amounts of data flowing through global networks, managing relationships with suppliers has become increasingly challenging. Traditional SRM approaches are no longer enough to keep up, often resulting in inefficiencies, higher risks, and missed optimization opportunities. Businesses need real-time insights, predictive capabilities, and a more strategic approach to supplier relationships, which traditional SRM methods can’t provide.

This research addresses these gaps by exploring how AI can revolutionize SRM. It will focus on enhancing predictive risk management, improving supplier selection, and optimizing the entire SRM process, ultimately leading to better supply chain performance.

Review of Relevant Literature

There has been increasing interest in using AI to transform SRM in recent years. Studies show that AI can significantly enhance supplier evaluation, risk management, and selection processes. For instance, research highlights AI's ability to predict supply chain disruptions by analyzing patterns in supplier behavior and performance data (Ivanov & Dolgui, 2020). Similarly, AI-driven optimization models have been found to lower procurement costs and make supplier negotiations more effective (Kumar et al., 2021).

However, while much has been written about the potential of AI in SRM, there is a lack of detailed research on specific methodologies and models for practical implementation. This project aims to build on the existing literature by developing AI models tailored to real-world SRM challenges and rigorously testing their effectiveness.

Research Methodology

The methodology for this project will be conducted in several key phases, combining both quantitative and qualitative approaches to collect and analyze data, develop models, and validate results.

1. Data Collection & Preparation:

   - Data on suppliers, products, locations, turnover, and previous performance will be collected from supply chain databases.

   - The data will be cleaned and prepared, ensuring consistency and accuracy for the AI models.

2. Feature Engineering:

   - Additional features, such as supplier capacity and turnover ratios, will be created to enhance the AI model's predictive power.

3. AI Model Development:

   - Various AI models will be developed and tested for specific SRM tasks:
     - Classification models (like Random Forest and XGBoost) to predict supplier risks and performance.
     - Regression models for forecasting supplier turnover and demand.
     - Clustering models to group suppliers based on similar performance metrics.
     - Recommendation systems to optimize supplier selection based on historical and real-time data.

4. Model Validation:

   - The models will be validated using historical data. Their accuracy, precision, and recall will be evaluated to ensure effectiveness.

5. Implementation & Visualization:

   - A dashboard will be developed to make the AI insights and recommendations easy to understand and act upon.

6. Continuous Improvement:

   - A feedback loop will be established to update the models with new data, ensuring they remain accurate and useful as conditions change.

Ethical Considerations and Risks

AI models can come with risks, particularly around bias and transparency. This research will take the following steps to ensure ethical standards are met:

1. Bias Monitoring: AI models will be checked regularly for biases to ensure they don't unfairly disadvantage any suppliers.

2. Transparency: The AI models will be designed to be transparent, meaning their decisions can be explained and understood by the businesses using them.

3. Ethical Approval: All data collection and research methodologies will undergo ethics approval to ensure they comply with regulations and respect supplier privacy.

Challenges and Limitations

This project faces several potential challenges:

1. Data Quality: The availability and quality of supplier data may vary, which could limit the accuracy of the AI predictions.
2. Model Bias: Despite efforts to mitigate bias, AI models can sometimes reflect historical data patterns that may result in biased predictions.
3. Resistance to Change: Businesses may be hesitant to adopt AI-based SRM systems due to concerns about cost, complexity, or trust in AI-driven decisions.

Project Timeline

Literature Review	1 Weeks	Conduct a comprehensive review of existing research and AI applications in SRM.
Ethics Approval	2 weeks	Submit and obtain ethics approval for data usage and research methodologies.
Data Collection	3 weeks	Gather and preprocess all necessary data for the project
Feature Engineering	4 weeks	Create and refine features to improve model performance
Model Development	6 weeks	Develop and train machine learning models for various SRM functions.
Implementation	8 weeks	Test and validate models, ensuring they meet performance criteria.
Continuous Improvement	Ongoing	)Implement feedback loops for model updates and performance enhancement.
Draft Submission	9 weeks	Submit the first draft of the dissertation for review and feedback.
Final Submission	10 weeks	Submit the final version of the dissertation

Conclusion

The integration of AI into Supplier Relationship Management represents a significant shift in how companies manage their supply chains. This proposal outlines a framework for using AI to enhance supplier evaluation, risk management, and strategic decision-making. By implementing AI, companies can gain deeper insights, make more informed decisions, and ultimately improve their supply chain performance. This research will contribute to both academic understanding and practical applications in supply chain management.

References

- Ivanov, D., & Dolgui, A. (2020). AI in Supply Chain Management. *International Journal of Production Research*.
- Kumar, P., et al. (2021). Optimizing Supplier Selection with AI. *Journal of Operations Management*.