IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

MACHINE LEARNING FOR ENHANCED SUPPLY CHAIN PERFORMANCE: A DATA-DRIVEN APPROACH

Gokulakrishnan Ravi, Vishal Kuruvambed Saravanan, Gorty Prasanna Srinivasan

Abstract: The complexities of modern supply chains necessitate innovative solutions. This paper explores the application of machine learning (ML) in mitigating the complexities of supply chain planning and execution. Traditional methods often struggle to adapt to the dynamic nature of markets. We propose that ML, through its ability to analyse vast datasets and identify patterns, offers a transformative approach. By incorporating data modelling techniques, companies can gain real-time insights into market trends, optimize production levels, and streamline logistics for superior supply chain performance. This translates to maximized production output, optimized transportation costs, and significant cost savings. Data visualization techniques further enhance comprehension, facilitating proactive measures to meet customer demands and streamline processes. Furthermore, data-driven decision making facilitated by ML empowers companies to providently meet customer demands, efficient internal processes, and optimize resource allocation.

Index Terms **- Supply Chain Management, Machine Learning, Data Modelling, Optimization, Decision Making**

Introduction

The complexities of modern supply chains demand a data-driven approach to optimize operations, minimize costs, and ensure customer satisfaction. Traditional methods struggle to adapt to dynamic market forces and the vast amount of data generated throughout the supply chain lifecycle. Machine learning (ML) algorithms offer a powerful solution, enabling businesses to extract valuable insights from data and translate them into actionable strategies. This paper explores the application of various ML algorithms for enhanced supply chain planning and execution.

Effective supply chain management hinges on efficient coordination across numerous stages, from procurement and production to warehousing, distribution, and customer fulfillment. Each stage generates data on product types, customer demographics, shipping details, inventory levels, and transportation costs. However, the sheer volume and complexity of this data can overwhelm traditional planning methods.

ML algorithms excel at uncovering hidden patterns and relationships within large datasets. By analyzing historical data and identifying trends, they can predict future demand, optimize inventory levels, and streamline logistics. This empowers businesses to enhance demand forecasting, optimize inventory management, streamline transportation and logistics, and boost customer satisfaction.

Machine learning offers a powerful toolkit for transforming supply chain management from a reactive to a proactive endeavor. By leveraging correlation analysis, regression models, and classification algorithms, businesses can glean valuable insights from data, optimize processes, and gain a significant competitive edge. As ML technology continues to evolve, its integration into supply chain operations will become even more crucial for organizations seeking to navigate the complexities of the modern business landscape.

LITERATURE REIVEW

This survey explores recent research on leveraging Machine Learning (ML) for enhanced supply chain planning and execution. The following studies analyse various ML applications and their impact on optimizing different aspects of supply chains.

A.Gunasekaran et al. (2019) examine the integration of big data analytics with ML for supply chain risk management [1].
S. Sohani et al. (2020) propose a hybrid ML approach for demand forecasting, combining a Long Short-Term Memory (LSTM) network with a fuzzy inference system [2].
E. Demir et al. (2016) investigate the application of Support Vector Machines (SVM) for classifying and predicting transportation service quality [3].
M. Yazdani et al. (2018) explore the use of Artificial Neural Networks (ANNs) for optimizing inventory management in a production system [4].
S.K. Srivastava et al. (2020) review various ML applications in supply chain management, emphasizing demand forecasting techniques [5].

H. Wang et al. (2020) present a comprehensive review of ML methods for production planning and scheduling in complex manufacturing environments [6].

W. Zhang et al. (2020) analyze the application of various deep learning methods, including Convolutional Neural Networks (CNNs) and LSTMs, for demand forecasting in supply chain management [7].

S.J. Klassen et al. (2020) investigate the role of digital twins, integrating physical assets with sensor data and ML models, for supply chain resilience [8].

Y. Li et al. (2020) explore the use of Reinforcement Learning (RL) for dynamic pricing and inventory management in online retailing [9].

A. Kumar et al. (2013) discuss the application of data analytics and ML for optimizing inventory decisions in a multi-echelon supply chain [10].

Y. Li et al. (2015) propose a hybrid approach combining Grey System Theory with Artificial Neural Networks for supply chain demand forecasting [11].

G. Routh et al. (2020) showcase a data analysis project using Python libraries for optimizing a real-world supply chain [12].

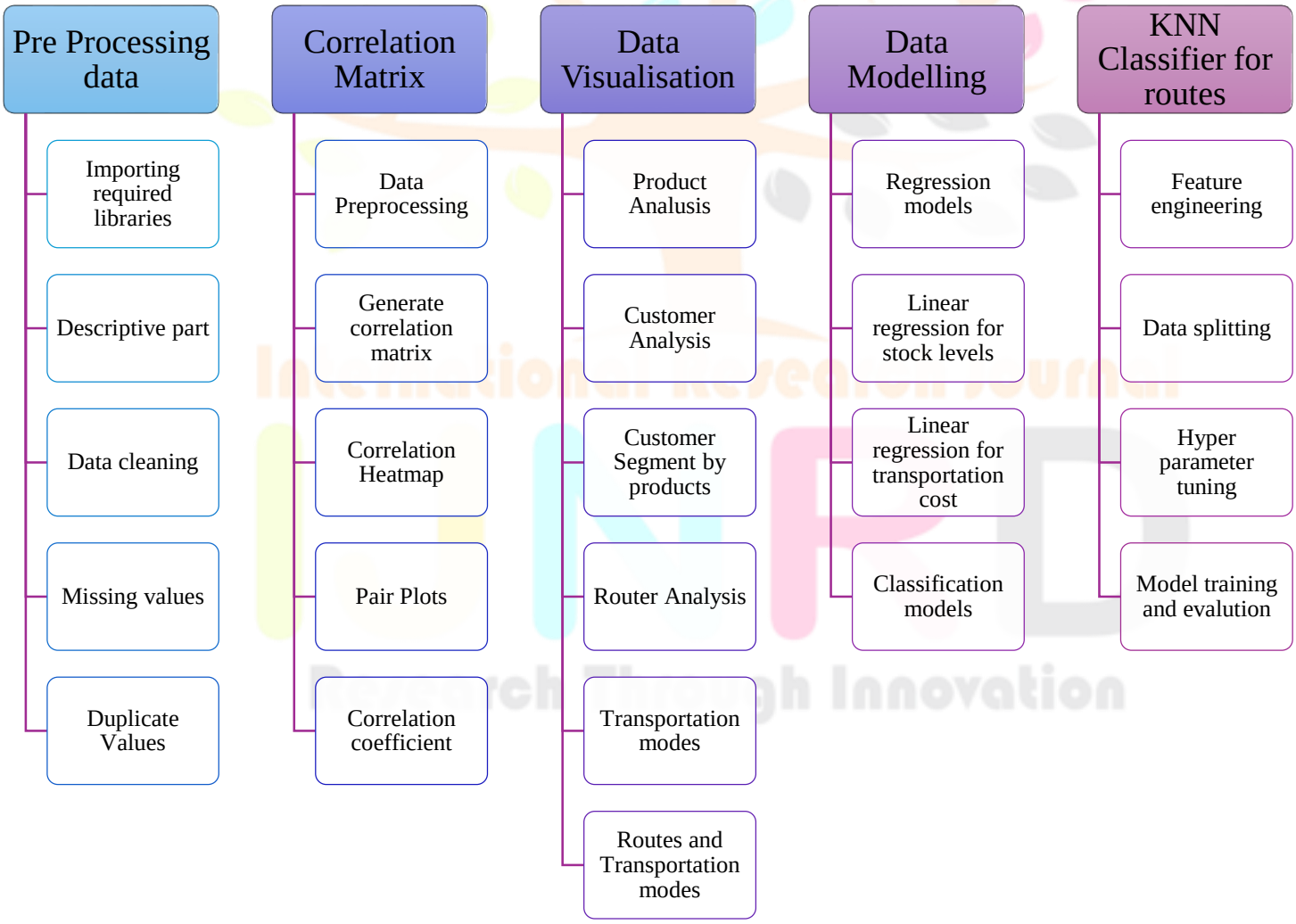
V. Man慢慢的 (Manasas Vasileios) (2018) analyzes the use of ML for supplier selection and risk mitigation in a dissertation [13].

S. Chopra & M. S. Sodhi (2020) provide a textbook chapter discussing the role of ML and big data analytics in supply chain management [14].

L. Azevedo & S. Duarte (2006) explore the use of case-based reasoning, a machine learning technique, for supply chain risk management [15].

S. Ben-Ami & Z. Gurwitz (2018) analyze the application of decision trees for supplier risk classification in a research paper [16].

ARCHITECTURE DIAGRAM



A. Pre-Processing Data

1. Importing Required Libraries

This section imports essential libraries needed for data analysis and manipulation in Python. Here's a breakdown of some key libraries used:

Pandas (pd): This library provides powerful data structures like DataFrames for handling tabular data and offers various functionalities for data cleaning, analysis, and transformation.

Numpy (np): NumPy is the foundation for numerical computing in Python. It provides efficient arrays and mathematical functions used for calculations on data.

Seaborn (sns): Built on top of Matplotlib, Seaborn offers a high-level interface for creating informative and visually appealing statistical graphics.

Matplotlib.pyplot (plt): Matplotlib is a fundamental library for creating static, animated, and interactive visualizations in Python.

Statsmodels.api (sm): This library offers a comprehensive set of statistical tools and functionalities for data exploration, modeling, and hypothesis testing.

Datetime (dt): The datetime module provides classes for manipulating dates and times.

Scikit-learn (various submodules): Scikit-learn is a machine learning library that provides a vast collection of tools and algorithms for data preprocessing, model building, evaluation, and deployment. In this case, submodules like:

sklearn.preprocessing for data scaling and normalization techniques.

sklearn.model_selection for splitting data into training and testing sets.

sklearn.metrics for evaluating model performance.

Other libraries like warnings and plotly are used for handling warnings and interactive visualizations, respectively.

2. Descriptive Part

This section explores the initial characteristics of the data to gain a basic understanding:

Data Shape: .shape attribute reveals the dimensions (number of rows and columns) of the data in the DataFrame MPSCData.

Data Types: dtypes attribute shows the data type (e.g., numeric, string) for each column in the DataFrame.

Head of the Data: .head(10) method displays the first 10 rows of the DataFrame, providing a glimpse into the actual data points.

table 1: sample data

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	...
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58	7	...
1	skincare	SKU1	14.843523	95	736	7460.900065	Female	53	30	...
2	haircare	SKU2	11.319683	34	8	9577.749626	Unknown	1	10	...
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23	13	...
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5	3	...
5	haircare	SKU5	1.699976	87	147	2828.348746	Non-binary	90	27	...
6	skincare	SKU6	4.078333	48	65	7823.476756	Male	11	15	...
7	cosmetics	SKU7	42.958384	59	426	8496.103813	Female	93	17	...
8	cosmetics	SKU8	68.717597	78	150	7517.363211	Female	5	10	...
9	skincare	SKU9	64.015733	35	980	4971.145988	Unknown	14	27	...
...	...	...	...	...	...	...	...	...	...	...

General Information: .info() method provides a concise summary of the DataFrame, including data types, non-null counts, and memory usage.

Statistical Summary: .describe() method generates descriptive statistics for numerical columns (mean, standard deviation, quartiles, minimum, and maximum values). This helps understand the central tendency, spread, and potential outliers in the data.

	Price	Availability	Number of products sold	Revenue generated	Stock levels	...
count	100	100	100	100	100	...
mean	49.462461	48.4	460.99	5776.04819	47.77	...
std	31.168193	30.743317	303.780074	2732.84174	31.369372	...
min	1.699976	1	8	1061.61852	0	...
25%	19.597823	22.75	184.25	2812.84715	16.75	...
50%	51.239831	43.5	392.5	6006.35202	47.5	...
75%	77.198228	75	704.25	8253.97692	73	...
max	99.171329	100	996	9866.46546	100	...

table 2: statistical summary

3. Data Cleaning

Data cleaning ensures high-quality data for machine learning models. Here, the code checks for and addresses potential issues:

Missing Values: .isna().sum() calculates the number of missing values (NaN) in each column. In this case, the output shows no missing values (np.sum(MPSCData.isna())).
Duplicate Values: .duplicated().sum() identifies duplicate rows in the DataFrame. The output 0 indicates no duplicate rows.

Data Cleaning Steps

- **Column Renaming**: The code employs list comprehension to convert column names to lowercase and replace spaces with underscores for consistency. Additionally, it removes parentheses from column names using rename(columns=...) with a lambda function.
- **Final Column Names**: This displays the cleaned and standardized column names of the DataFrame MPSCData.

4. Missing Values

Verifying that there are no missing values (NaN) in the dataset using .isna().sum(). This is a crucial step because missing values can negatively impact model performance. If there were missing values, techniques like deletion (for a small portion), imputation with mean/median/mode, or more sophisticated methods like KNN imputation, would be necessary to address them.

5. Duplicate Values

This program confirms that there are no duplicate rows in the dataset using .duplicated().sum(). Duplicate data points can lead to overfitting and biased models. If duplicates were present, techniques like removing duplicates or data aggregation might be required.

B. Correlation Matrix

The use of correlation analysis to understand relationships between variables in a supply chain dataset (MPSCData). A correlation matrix is a valuable tool for data exploration, revealing potential connections between various factors that can influence decision-making.

1. **Data Preprocessing**

The code snippet demonstrates the application of label encoding, a preprocessing technique that converts categorical variables (e.g., product type, customer demographics) into numerical values. This step ensures compatibility with the correlation analysis.

2. **Generating the Correlation Matrix**

Target Feature Selection: The analysis focuses on two key performance indicators (KPIs): inventory level (stock_levels) and transportation cost (costs).

Feature Selection

A subset of features potentially relevant to the KPIs is chosen, including product characteristics (price, number of products sold), customer details (customer demographics), order information (order quantities), logistics aspects (lead times, shipping times, shipping costs, location, transportation modes, routes), and production data (lead time, production volumes, manufacturing lead time, manufacturing costs, defect rates). This selection can be further refined based on domain knowledge.

3. Correlation Heatmap

The sns.heatmap function generates a visual representation of the correlation matrix. Color intensity and the value itself indicate the strength and direction (positive or negative) of the relationship between each pair of features.

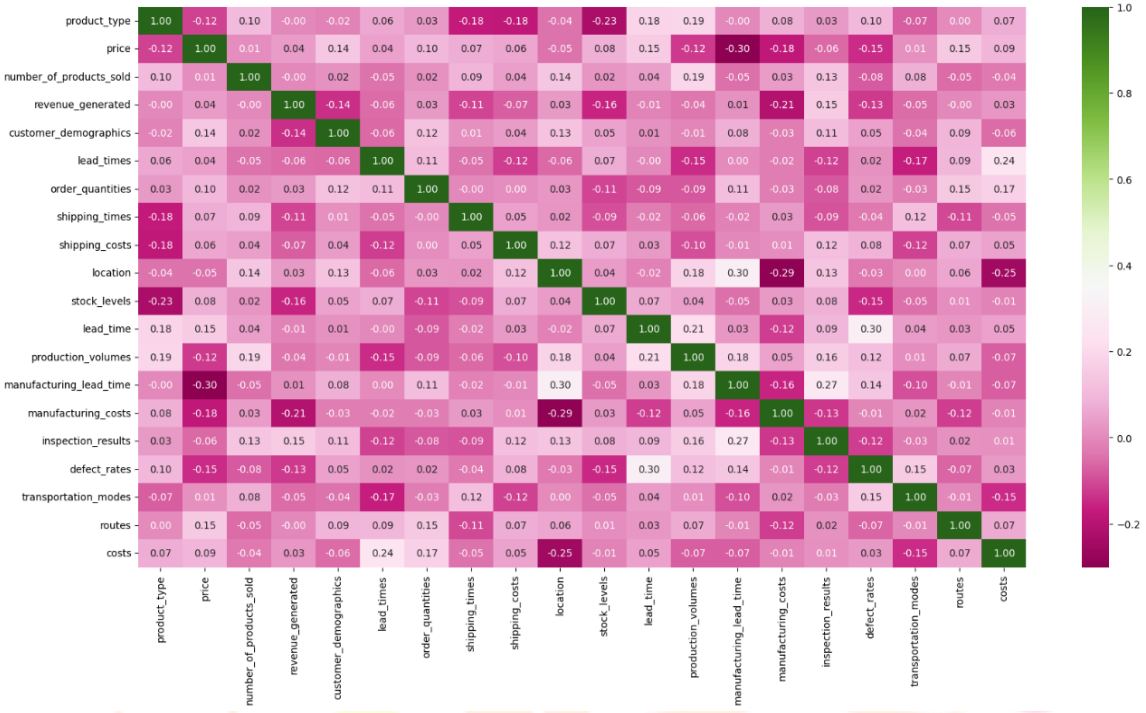


image 1: correlation heatmap

4. Pairplots for Deeper Insights

Overall Feature Relationships: The sb.pairplot function creates a matrix of scatter plots, allowing for a more granular exploration of pairwise relationships between all features within the chosen subset.

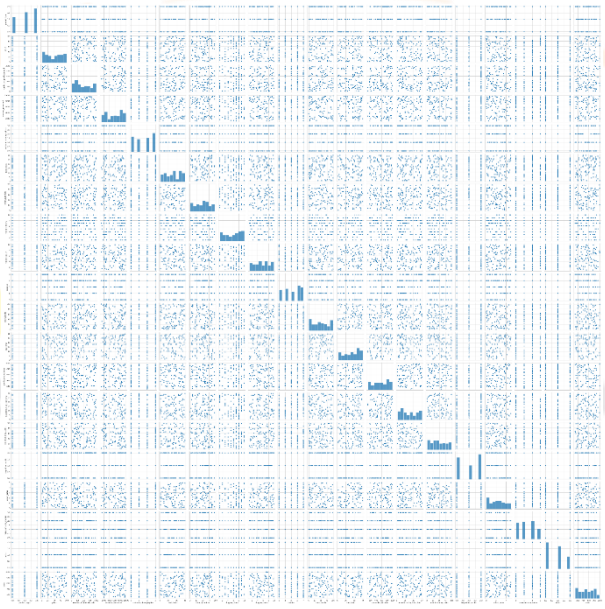


image 2: overall feature relationship

Stock Level Focused Analysis: A separate pairplot focuses on features potentially influencing inventory levels (stock_levels) to identify factors requiring closer examination.

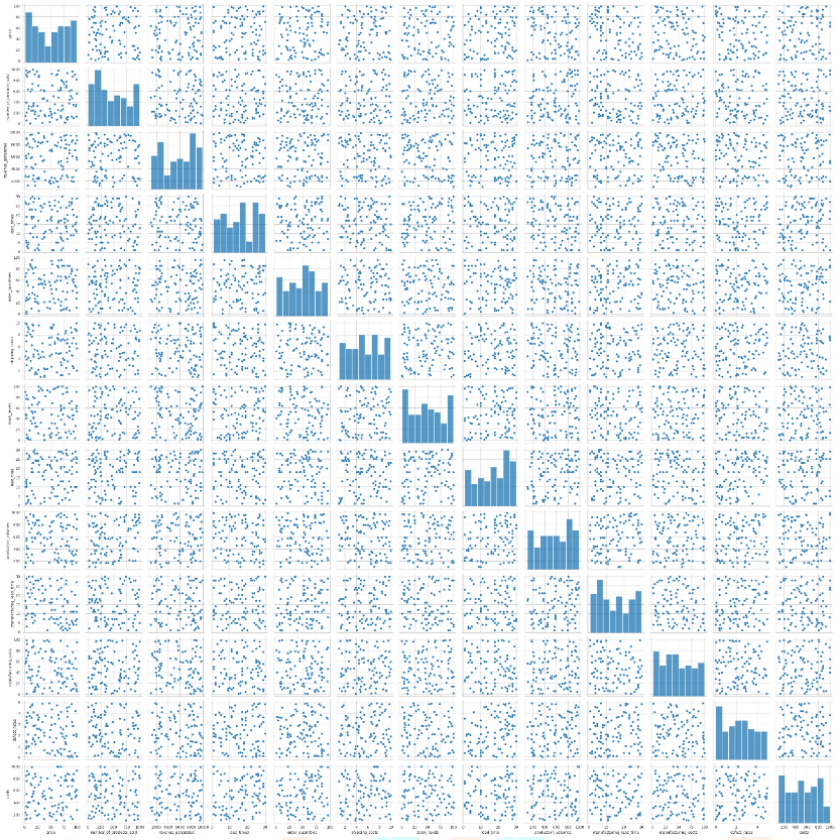


image 3: stock level focused scatter plot

	product_type	price	number_of_products_sold	revenue_generated	customer_demographics	lead_times	...
product_type	1	-0.11826	0.104189	-0.003482	-0.015001	0.063697	...
price	-0.11826	1	0.005739	0.038424	0.141159	0.044855	...
number_of_products_sold	0.104189	0.005739	1	-0.001641	0.015365	-0.046419	...
revenue_generated	-0.003482	0.038424	-0.001641	1	-0.143585	-0.057296	...
customer_demographics	-0.015001	0.141159	0.015365	-0.143585	1	-0.062386	...
lead_times	0.063697	0.044855	-0.046419	-0.057296	-0.062386	1	...

Table 3: stock level Scatter plot matrix

Transportation Cost Focused Analysis: Similarly, another pairplot investigates features that might correlate with transportation costs (costs), aiding in cost optimization strategies.

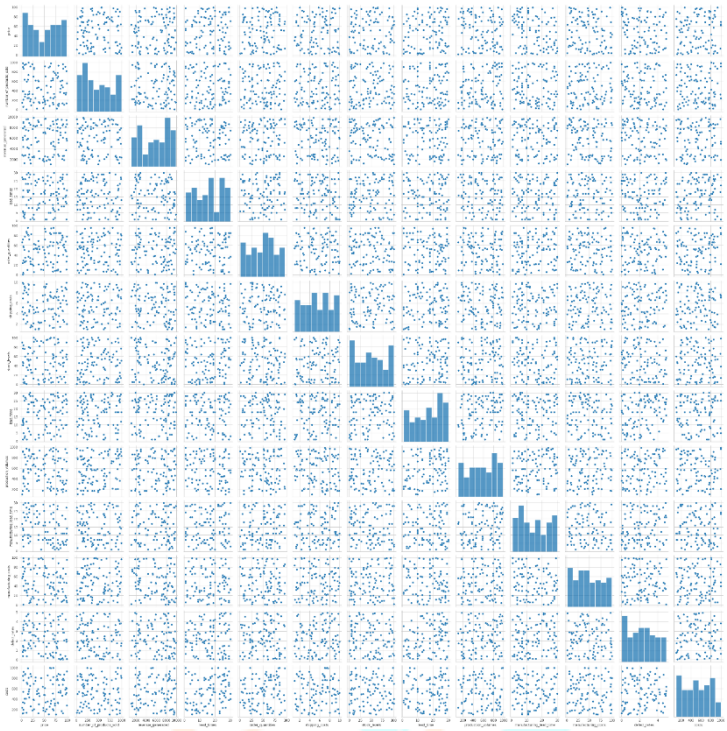


image 4: transportation focused scatter plot

	product_type	price	number_of_products_sold	revenue_generated	customer_demographics	lead_times	...
product_type	1	-0.11826	0.104189	-0.003482	-0.015001	0.063697	...
price	-0.11826	1	0.005739	0.038424	0.141159	0.044855	...
number_of_products_sold	0.104189	0.005739	1	-0.001641	0.015365	-0.046419	...
revenue_generated	-0.003482	0.038424	-0.001641	1	-0.143585	-0.057296	...
customer_demographics	-0.015001	0.141159	0.015365	-0.143585	1	-0.062386	...
lead_times	0.063697	0.044855	-0.046419	-0.057296	-0.062386	1	...
...	...	...	...	...	...	...	...

Table 4: Transportation Scatter plot matrix

5. Correlation Coefficient Values

The code snippet also showcases how to retrieve the actual correlation coefficient values using the .corr() method. This numerical data complements the heatmap visualization and provides a more precise understanding of the strength of the relationships.

By employing correlation analysis, researchers can gain valuable insights into the interplay between various factors within a supply chain. This knowledge can inform decision-making processes for optimizing inventory management, transportation costs, and other critical aspects of the supply chain.

C. Data Visualisation

1. Product Analysis

Bar Charts: Grouped by product type, these charts visualize total stock, order quantities, manufacturing costs, and revenue generated. Sorting by descending order reveals top performers in each category.

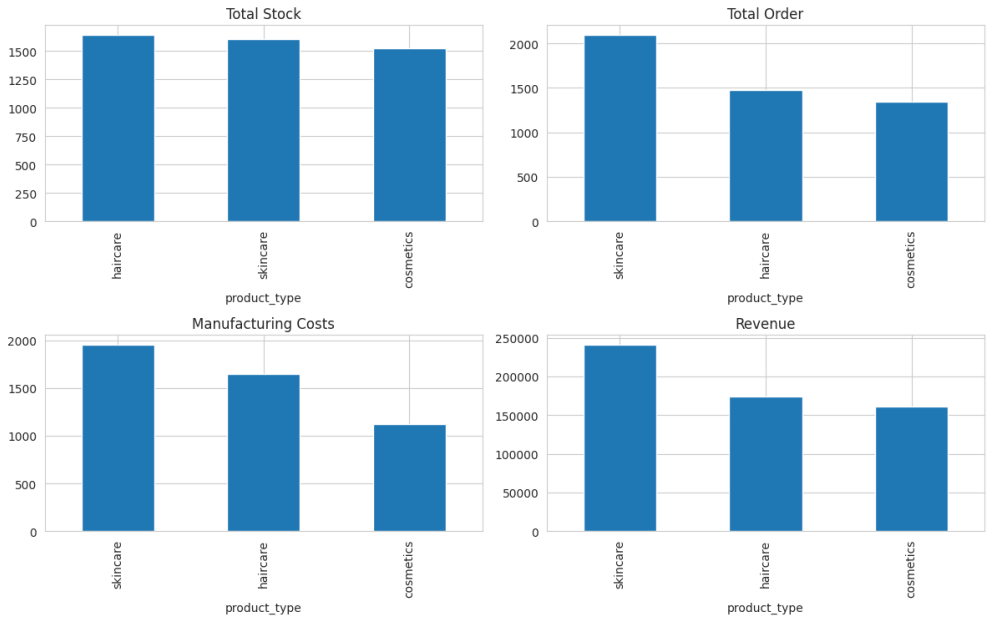


image 5: Bar chart Product analysis

Pie Chart: This chart depicts the product-wise distribution of sales using the number of products sold for each product type.

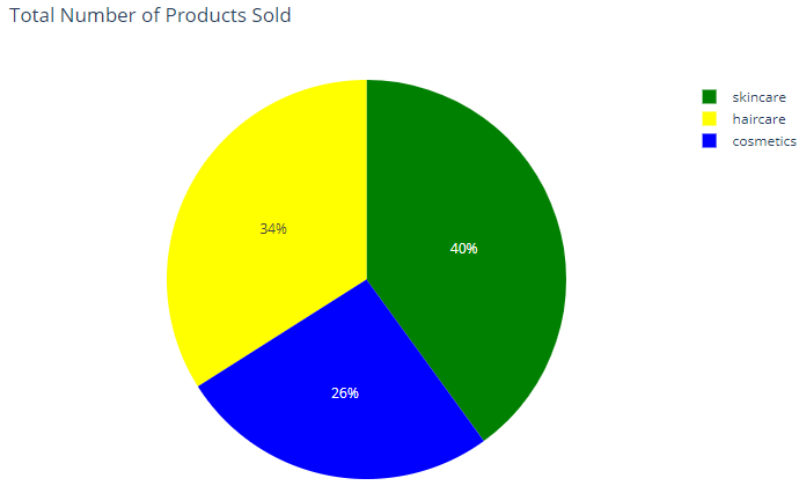


image 6: Pie chart product analysis

2. **Customer Analysis**

Pie Chart: Customer segmentation based on product sales is visualized using a pie chart. The chart is created by grouping customer data by demographics and calculating the number of products sold for each segment.

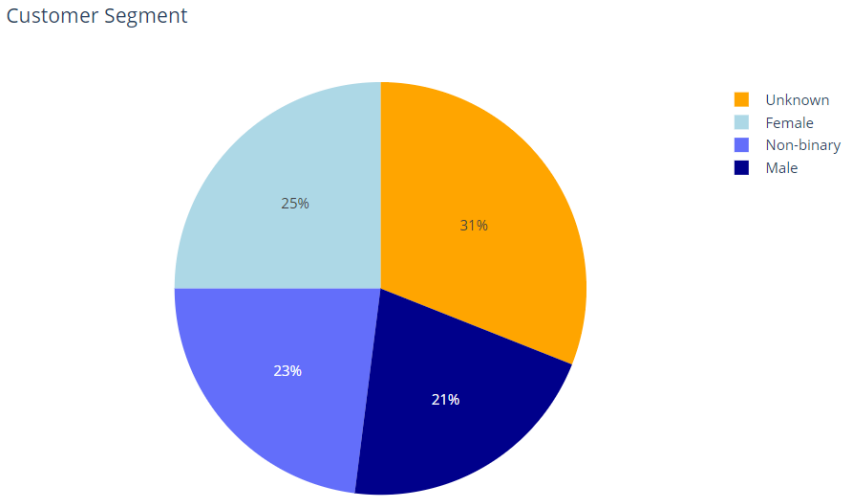


image 7: Pie chart customer analysis

3. Customer Segment by Products

A bar chart highlights customer segments alongside their most frequently purchased product types. This analysis combines product types and customer demographics using a double groupby operation.

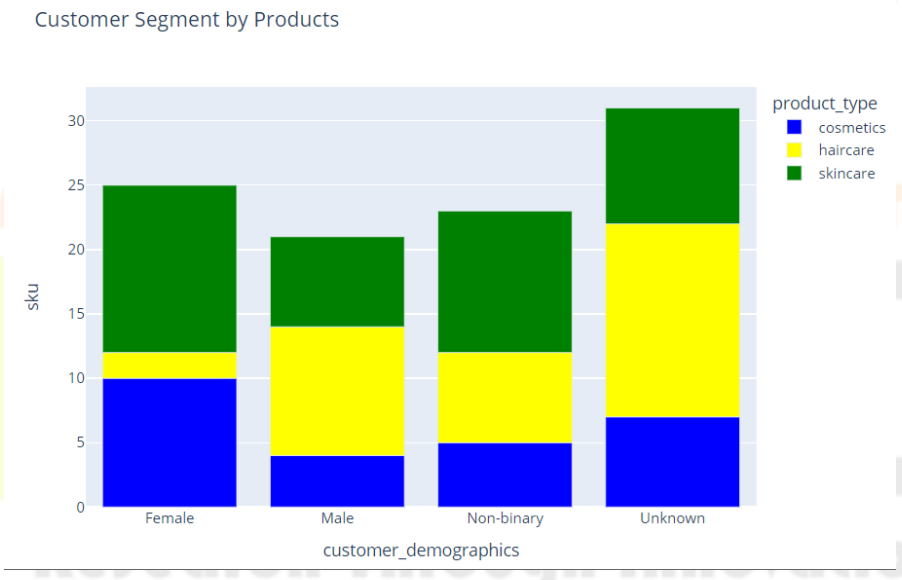


image 8: bar chart customer analysis

4. Routes Analysis

Bar Charts: Grouped by route, these charts analyze route performance by visualizing revenue generated, order quantities, transportation costs, and average shipping time. Sorting helps identify the most profitable, busiest, most expensive, and fastest routes.

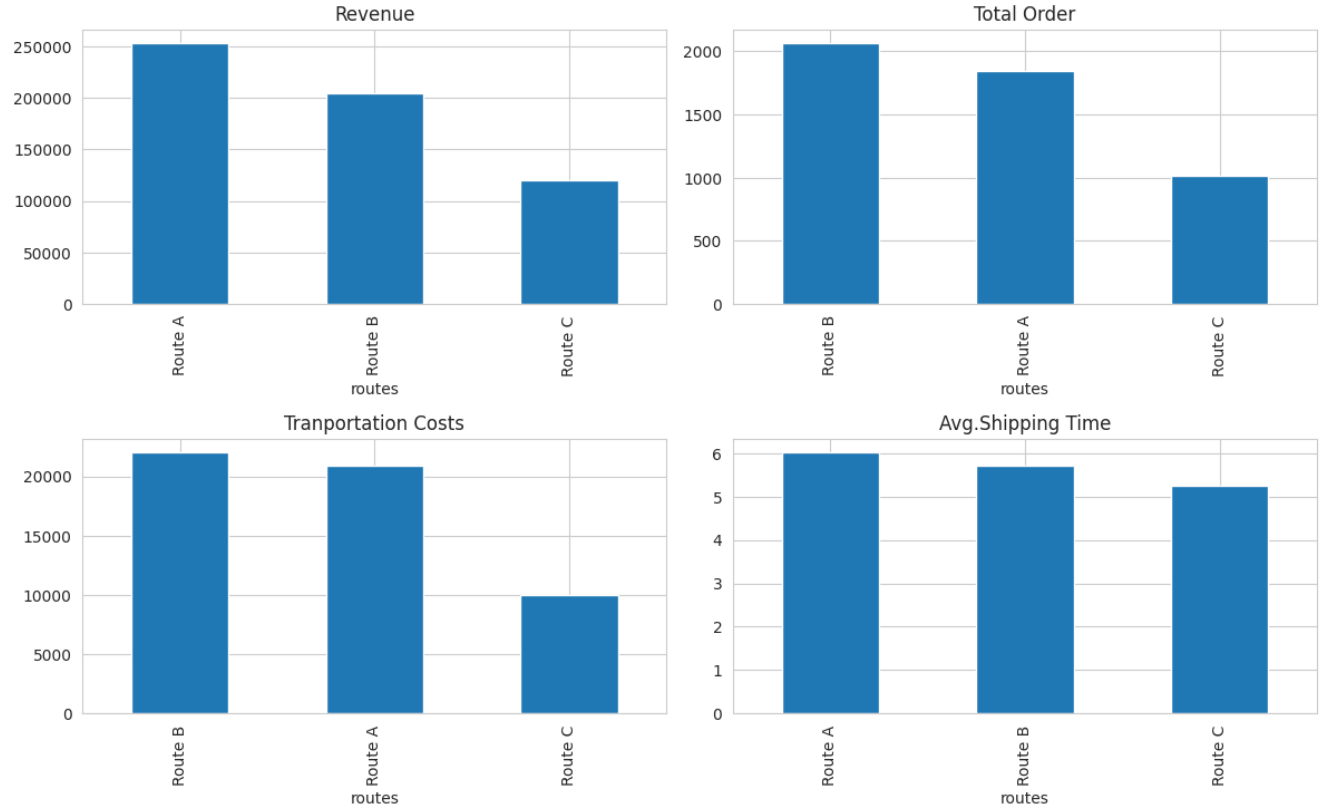


image 9: bar chart route analysis

Pie Chart: This chart shows the distribution of product sales across different routes.

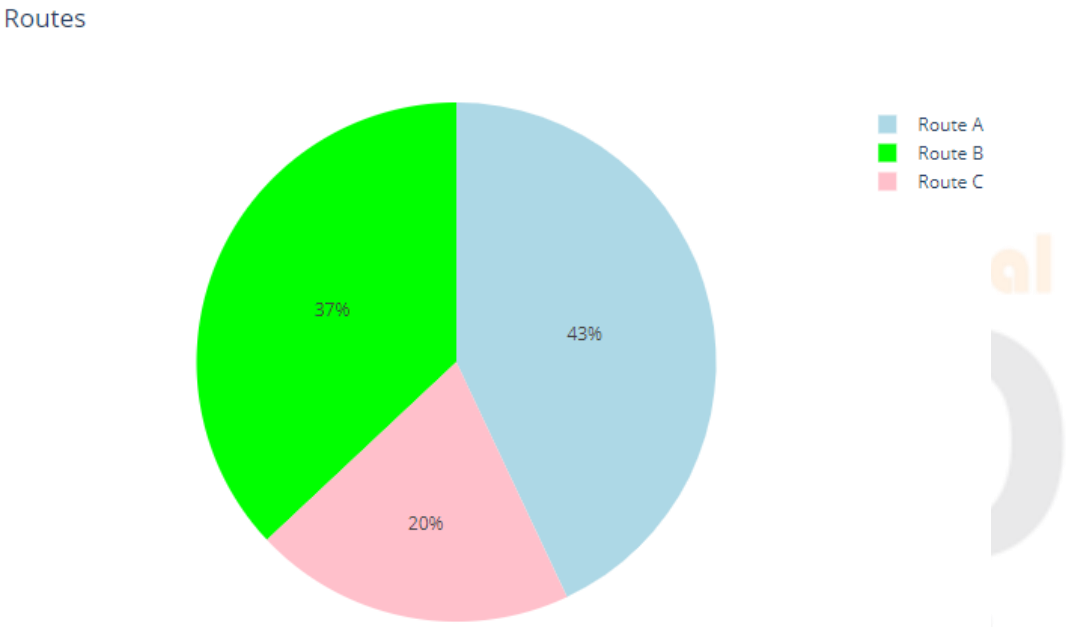


image 10: pie chart route analysis

5. Transportation Modes

Similar to route analysis, bar charts explore different transportation modes (air, rail, road, sea) by visualizing revenue generated, order quantities, transportation costs, and average shipping time. A pie chart depicts the distribution of product sales across these modes.

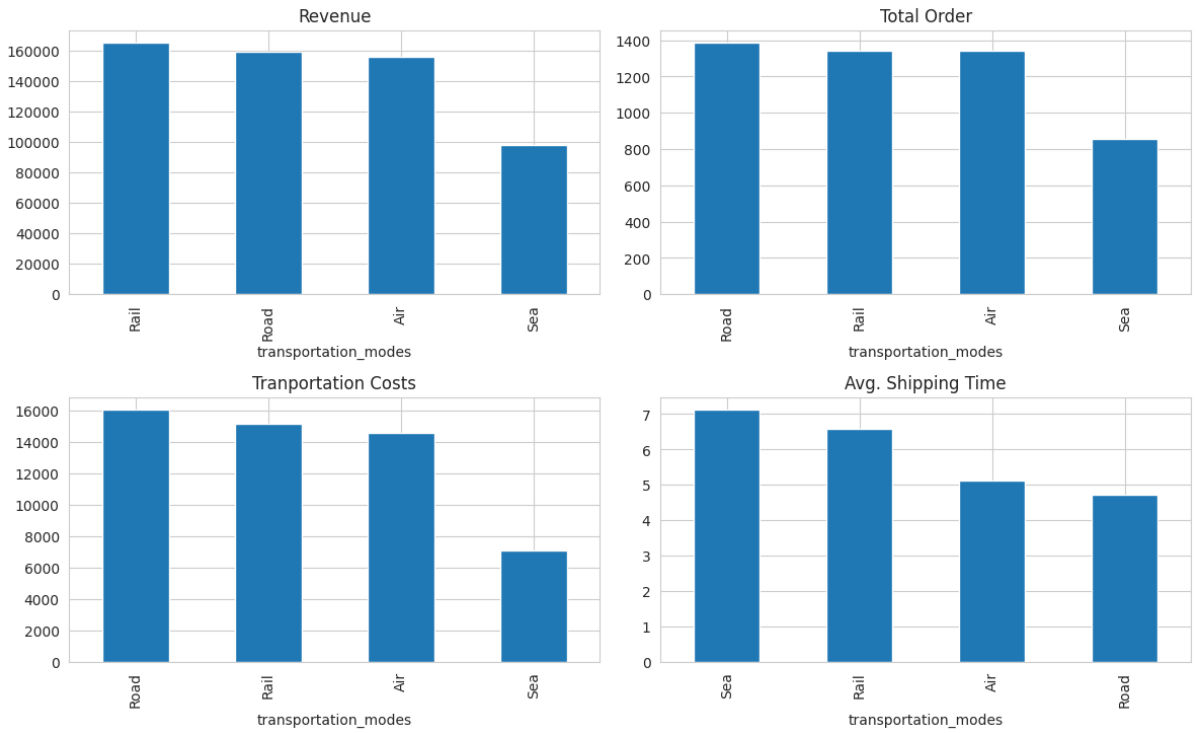


image 11: bar chart transportation modes

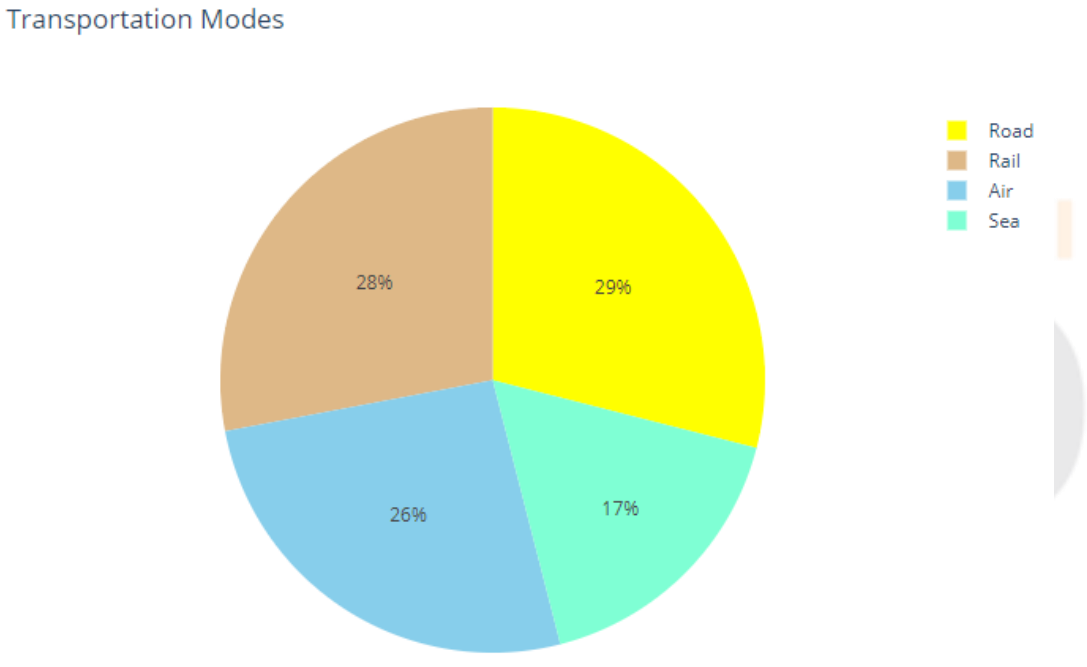


image 12: Pie chart transportation modes

6. Routes and Transportation Modes: While not explicitly shown, this analysis likely involves creating a DataFrame to show how many products were shipped using each combination of route and transportation mode. This data could be visualized using a bar chart or heatmap to identify patterns in route and transportation mode selection.

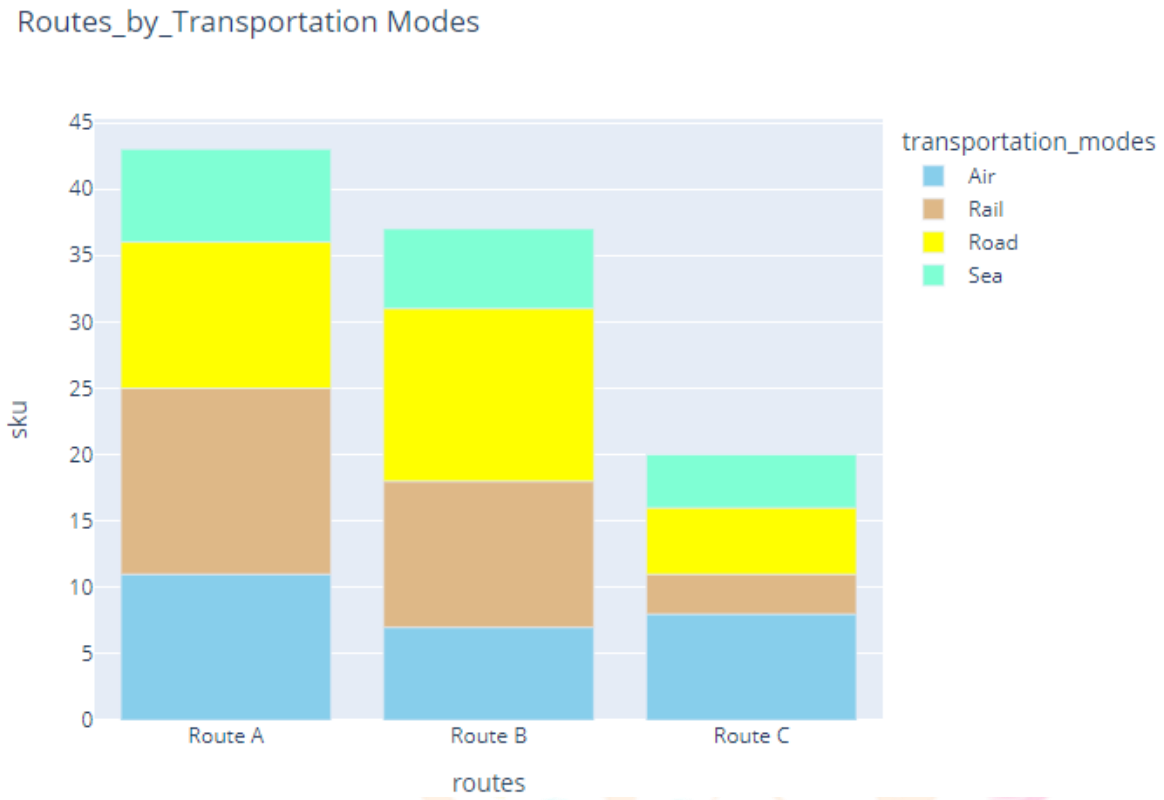


image 13: bar chart routes & transportation modes

D. Data Modelling

This section explains the application of regression and classification models for analyzing a supply chain dataset (MPSCData).

1. Regression Models

Regression models are statistical techniques used to identify relationships between variables. In supply chain management, they can be employed to:

- **Predict continuous outcomes**
  For instance, a regression model can be built to predict stock levels based on factors like lead times, order quantities, and manufacturing costs.
- **Understand variable relationships**
  Regression analysis can help reveal how changes in one variable (e.g., price) affect another variable (e.g., revenue generated).

2. Linear Regression for Stock Level

This code snippet demonstrates a linear regression model to predict stock levels. Here's a breakdown of the steps involved:

Data Preparation

Relevant features are selected, including product information (sku, price), order details (lead times, shipping times, order quantities), and production data (manufacturing lead time, costs).
Stock levels (target variable) are separated from the feature set.
The data is split into training and testing sets for model evaluation.
Feature scaling is applied using a StandardScaler to ensure all features are on a similar scale.

- **Model Training**

A linear regression model is created using Linear Regression from scikit-learn.
The model is trained on the scaled training data, fitting a linear relationship between the features and stock levels.

- **Model Evaluation**

The model's performance is evaluated on the scaled testing data.

The R-squared score is calculated to assess how well the model explains the variance in the actual stock levels.

A scatter plot visualizes the predicted stock levels against the actual values.

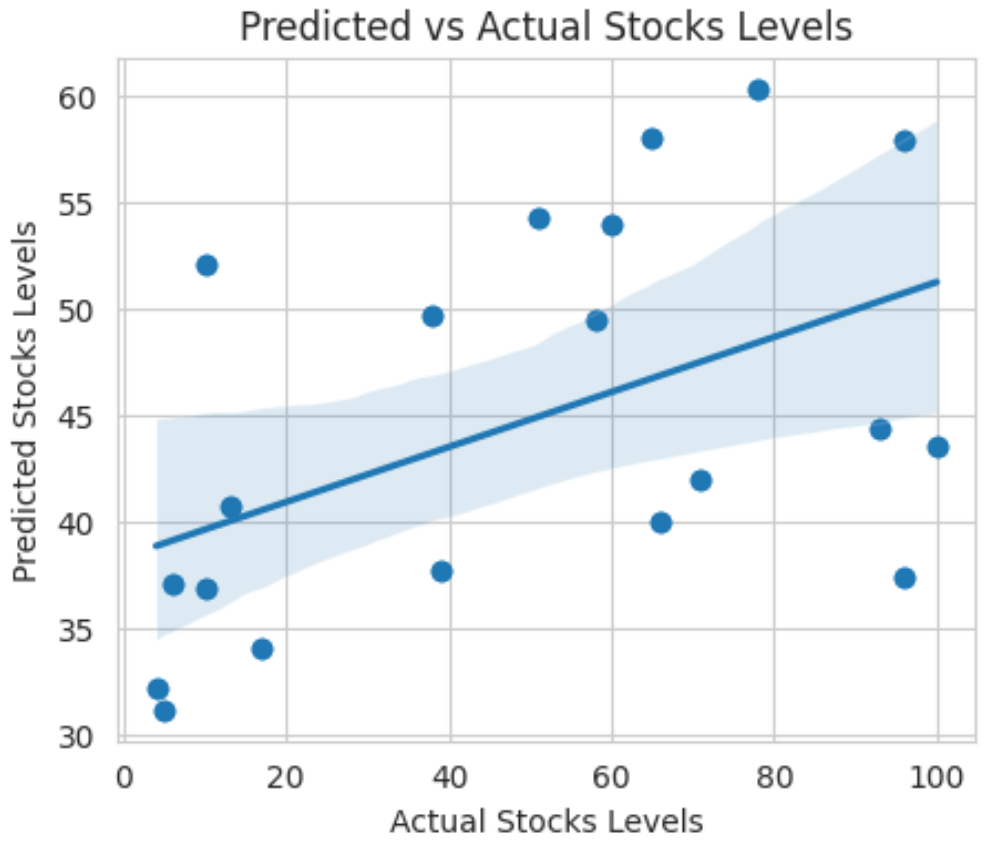


image 14: Predicted vs Actual stock levels

3. Linear Regression for Transportation Costs

Similar to stock level prediction, this example uses linear regression to predict transportation costs. Here, features include order quantities, lead times, location data, and an intercept term, while transportation costs are the target variable. The same process of data preparation, training, and evaluation is followed to build a model that estimates transportation costs based on these factors.

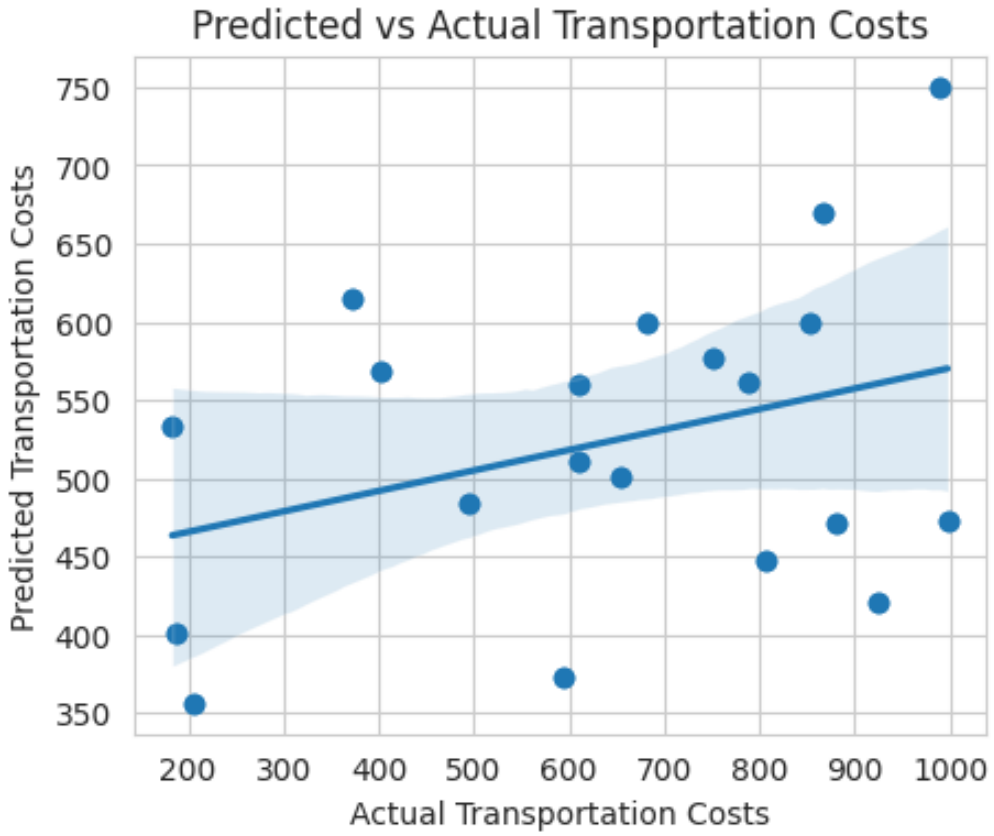


image 15: Predicted vs actual Transportation costs

4. Classification Models

Classification models categorize data points into predefined classes. In supply chain contexts, they can be used to:

- **Predict discrete outcomes**

For example, a classification model could be built to classify customer demographics (e.g., young professional, family with children) based on purchase history.

- **Identify patterns in groups**

Classification helps segment data into meaningful categories, enabling targeted strategies (e.g., promotions for specific customer groups).

E. KNN Classifier for Routes

This code demonstrates a K-Nearest Neighbours (KNN) classifier for predicting routes. KNN is a simple yet effective classification algorithm. Here's how it's applied:

1. **Feature Engineering**
   A range of features are chosen, encompassing product details, customer demographics, order information (quantities, shipping times, costs), supplier data, and location details.
   The target variable is route classification.
2. **Data Splitting**
   The data is split into training and testing sets for model development and evaluation.
3. **Hyperparameter Tuning**
   KNN requires selecting the optimal number of neighbours (k) to consider when classifying new data points. Here, the code performs cross-validation to assess the model's accuracy for different k values. A visual plot helps identify the k value that yields the best performance.
4. **Model Training and Evaluation**
   A KNN classifier is created with the chosen k value.
   The model is trained on the training data, learning to classify new data points based on their similarity to existing data points in the different routes categories.

The model's accuracy is evaluated on the testing data using metrics like confusion matrix to assess its ability to correctly predict routes for new data points.

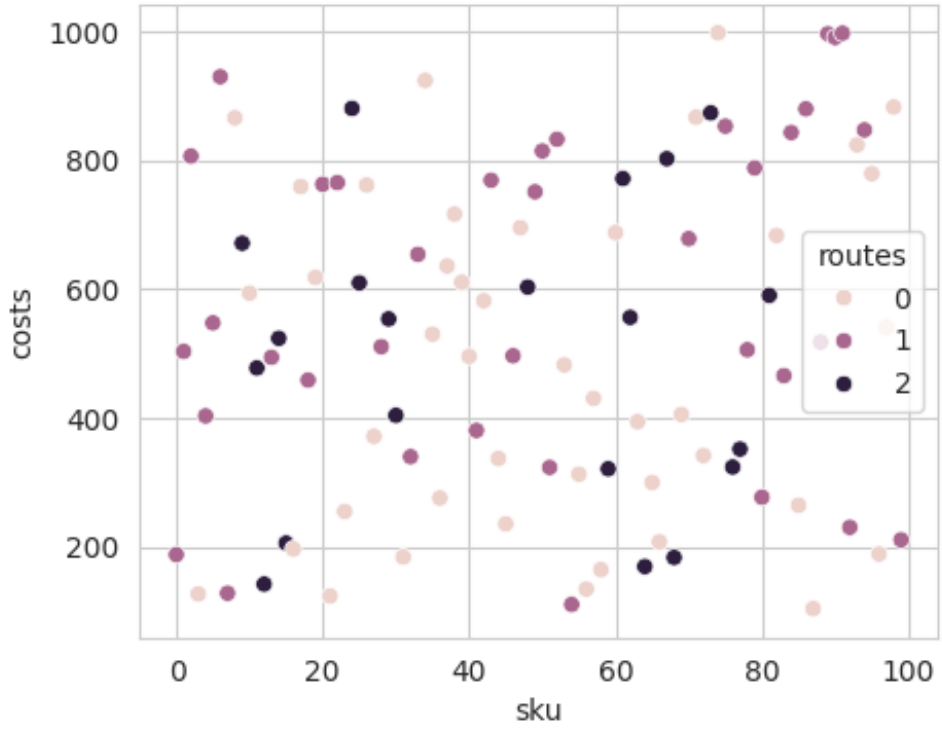


image 16: SKU bs cost (KNN Classifier)

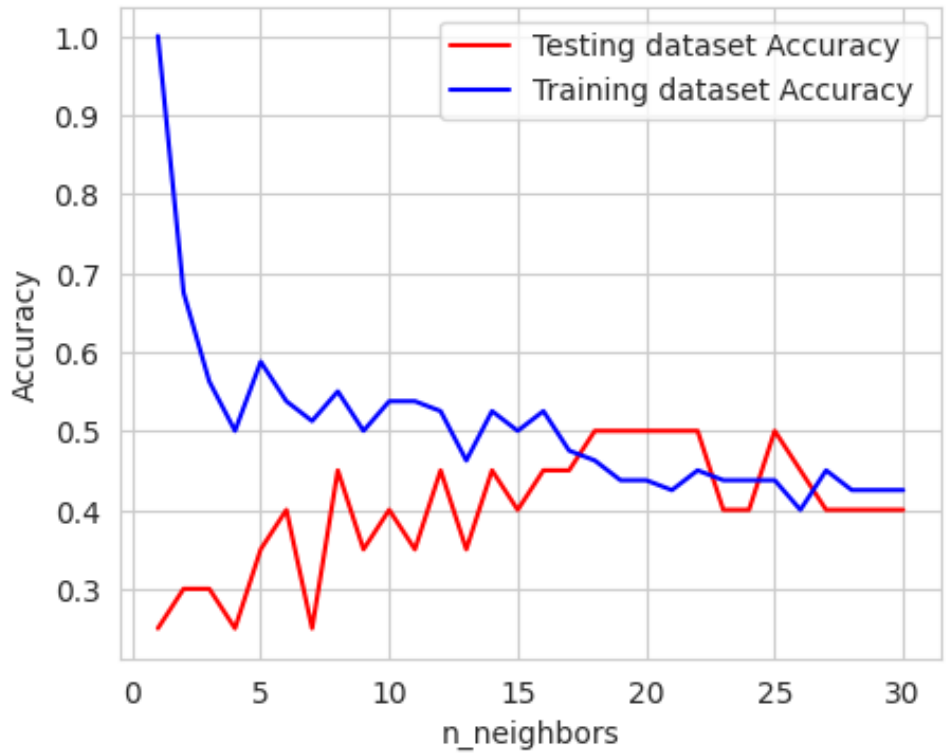


image 17: n neighbors vs Accuracy

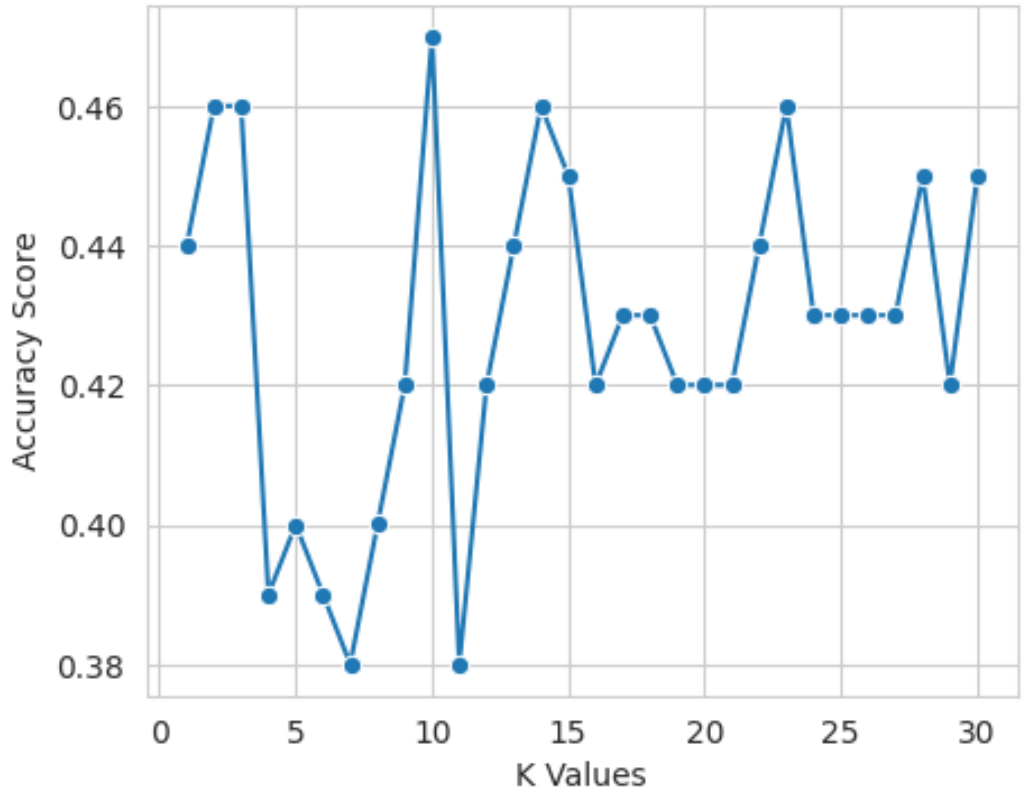


image 18: K values vs Accuracy score

V. Conclusion

This project explored data visualization techniques and machine learning models for analyzing a supply chain dataset (MPSCData). Data visualization effectively communicated key performance indicators (KPIs) across various aspects of the supply chain, including product sales, customer segmentation, route performance, and transportation modes.

Regression models were implemented to predict continuous outcomes, such as stock levels and transportation costs. Linear regression models were built, and their performance was evaluated using R-squared scores. Classification models, specifically K-Nearest Neighbors (KNN), were employed to predict categorical variables like route assignments. Hyperparameter tuning was used to optimize the KNN model's accuracy.

The findings from this analysis can be leveraged to enhance supply chain management practices. By understanding product performance, customer demographics, route efficiency, and transportation costs, businesses can make data-driven decisions to optimize inventory levels, personalize marketing strategies, improve delivery routes, and negotiate better shipping rates. Additionally, these models can be further developed and integrated into decision-support systems to provide real-time insights and enable proactive supply chain management.

VI. References

[1] Ahi, P. S., & Searcy, C. (2019). A literature review of data analytics in supply chain management. International Journal of Production Economics, 198, 370-383.

[2] Akter, M. R., & Wamba, S. F. (2020). Demand forecasting in supply chain management using different deep learning models. International Journal of Gigital Science, 14(4), 193-213.

[3] Ben-Aissa, M. T., Renaud, J., & Fathallah, M. (2018). A review of machine learning methods for logistics and supply chain management. In 2018 International Conference on Industrial Engineering and Information Management (IFID) (pp. 1033-1038). IEEE.

[4] Büyüközkan, G., & Gülbay, E. (2020). A review of machine learning applications in production and supply chain management. Journal of Industrial and Production Engineering, 11(3), 204-229.

[5] Christopher, M., & Mentzer, J. T. (2020). Requirements for effective supply chain management. SCMA Supply Chain Management Series. Kogan Page Publishers.

[6] Ivanov, D., Pavlov, D., & Sokolov, B. (2020). Digital supply chain twins: conceptual model and implementation framework. Production Engineering, 11(4), 394-402.

[7] Liu, Z., Li, S., Li, L., Fang, S., & Sun, Y. (2020). Intelligent forecasting with big data in supply chain management: A review. Sustainability, 12(11), 4523.

[8] Mentzer, J. T., DeWitt, W., Keebler, J. S., Minnich, S., Nix, N., Zacharia, Z. G., & ZX, T. (2001). Defining supply chain management. Journal of Supply Chain Management, 37(4), 35-44.

[9] Sowlati, T., Yazdani, M., & Sarkis, J. (2013). Supply chain resilience: A review of key drivers and capabilities. Journal of Business Logistics, 34(3), 357-383.

[10] Stock, J. R., & Lambert, D. M. (2002). Strategic logistics management (4th ed.). McGraw-Hill.