Supply Chain Data Analysis Report

# Introduction

This report outlines the comprehensive analysis of supply chain data aimed at identifying key factors influencing delivery risks, predicting delivery outcomes, and optimizing supply chain efficiency. The analysis involved various methodologies including data preprocessing, statistical analysis, predictive modeling, clustering, dimensionality reduction using PCA, and time-series analysis.

# Data Preprocessing

The dataset was preprocessed by converting date columns and handling missing values. Zipcodes were filled using the mode, and numerical columns were filled using the median values. An additional 'Shipping Delay' column was created to measure the difference between 'Days for shipping (real)' and 'Days for shipment (scheduled)'.

# Key Findings from Statistical Analysis

Various statistical tests such as the Chi-square test and ANOVA were applied to explore the relationships between variables. The Chi-square test revealed a significant association between shipping modes and late delivery risk, while ANOVA uncovered significant differences in shipping delays across delivery statuses.

Chi-square Test Results:

Chi-square statistic: 37716.04, p-value: 0.0

ANOVA Test Results: p-value: 0.0

**Interpretation:** The Chi-square test is typically used to determine whether there is a significant association between two categorical variables. In this case, the very high Chi-square statistic along with a p-value of 0.0 strongly suggests that there is a statistically significant association between the variables being tested. A p-value of 0.0 indicates that the likelihood of observing such a data distribution due to chance is extremely low, thus the null hypothesis (no association between the variables) can be rejected with confidence.

### ANOVA Test Results

* **P-value: 0.0**

**Interpretation:** The ANOVA test is used to determine whether there are any statistically significant differences between the means of three or more independent (unrelated) groups. Here, a p-value of 0.0 indicates that there is a statistically significant difference between the group means. This means you can confidently reject the null hypothesis that all group means are equal, suggesting that at least one group mean differs from the others.

# Predictive Modeling

The Logistic Regression Classification Report and the results from Ridge and Lasso Regression provide comprehensive insights into model performance in predicting a binary outcome and estimating continuous outcomes respectively. Here’s an interpretation of these results:

### Logistic Regression Classification Report Interpretation

1. **Precision, Recall, and F1-Score:**
   * **Class 0 (Precision: 1.00, Recall: 0.94, F1-Score: 0.97):** This class shows a perfect precision indicating that whenever the model predicts class 0, it is correct. However, the recall is slightly lower, suggesting that while the model is precise, it misses some actual class 0 instances (6% miss rate). The high F1-score indicates an excellent balance between precision and recall.
   * **Class 1 (Precision: 0.96, Recall: 1.00, F1-Score: 0.98):** The precision here is slightly lower than for class 0, but still very high, indicating few false positives. The recall is perfect, meaning the model captures all actual instances of class 1. The F1-score is exceptionally high, reflecting a robust predictive ability for this class.
2. **Overall Model Accuracy:**
   * The overall accuracy of 0.97 suggests that the model accurately predicts the correct class for 97% of the cases in the test set. This is supported by high macro and weighted average scores for precision, recall, and F1-score, demonstrating strong performance across both classes.

### Ridge and Lasso Regression RMSE Interpretation

1. **Ridge Regression (RMSE: 0.310185066096759):**
   * The RMSE (Root Mean Square Error) for Ridge Regression is relatively low, indicating that the predictions are close to the actual data points. This model is more robust against overfitting due to its ability to handle multicollinearity and shrink coefficients, which is reflected in the smaller error.
2. **Lasso Regression (RMSE: 0.4976584802346626):**
   * Lasso Regression shows a higher RMSE compared to Ridge, suggesting that the predictions deviate more from the actual values. Lasso tends to zero out less important features as part of its regularization process, which might result in losing some useful information if not all features are equally significant.

### Insights and Implications

* **Model Choice and Feature Importance:**
  + The Logistic Regression model is extremely effective for classification tasks within this dataset, achieving high accuracy and balance between precision and recall.
  + For continuous prediction, Ridge Regression seems to perform better than Lasso in this context, which may guide the choice of regularization technique based on the specific characteristics of the dataset and the prediction objectives.
* **Operational Use:**
  + The Logistic Regression model could be effectively used for operational decisions where classifying into one of two categories (e.g., pass/fail, yes/no) is required.
  + Ridge Regression might be more suited for predictions involving a range of continuous outcomes where maintaining the influence of all features is beneficial.
* **Further Investigation:**
  + It would be useful to explore why Lasso Regression did not perform as well as Ridge, perhaps by examining the distribution and relevance of the features used. Additionally, tuning the regularization strength (lambda) and reassessing the feature set might improve its performance.

These interpretations provide a foundation for making informed decisions about model selection and application in real-world scenarios based on their performance characteristics.

# Feature Importance (RandomForestClassifier)

The Random Forest feature importance table you provided outlines the relative importance of various features in predicting a target variable (likely related to delivery performance or customer satisfaction). Here’s an interpretation of these results:

### Interpretation of Feature Importance

1. **Shipping Delay (0.757718):**
   * This feature has the highest importance score, significantly higher than any other feature. It suggests that the delay in shipping is the most critical predictor in the model. This could indicate that the extent of shipping delays heavily influences the outcome variable, such as customer satisfaction or the likelihood of late deliveries. The model's effectiveness in making predictions is primarily contingent upon how shipping delays are managed and recorded.
2. **Days for Shipping (Real) (0.163104):**
   * The actual days taken for shipping is the second most important feature. This indicates a substantial impact on the model’s predictions, though less so compared to shipping delays. This feature's importance signifies that the total duration of the shipping process is a crucial factor, possibly affecting delivery timeliness or other operational aspects in the supply chain.
3. **Benefit per Order (0.050050):**
   * The financial benefit associated with each order holds moderate importance. While significantly lower than shipping-related features, its influence suggests that the profitability or cost-effectiveness of orders plays a role in the model’s predictions. This might relate to how financial outcomes impact customer loyalty or order fulfillment satisfaction.
4. **Sales per Customer (0.029128):**
   * This feature has the lowest importance score among the ones listed, indicating a minimal impact on the predictive outcome of the model. This suggests that while sales per customer are relevant for business metrics, they do not heavily influence the more operational or satisfaction-related outcomes modeled by this Random Forest.

### Insights and Implications

* **Operational Focus:**
  + Given the high importance of shipping delay, efforts to minimize delays should be prioritized to improve the predictive outcomes—be it reducing late deliveries, enhancing customer satisfaction, or optimizing logistic operations.
* **Process Optimization:**
  + Addressing factors that contribute to the actual shipping days could also yield significant improvements in operational efficiency and potentially enhance customer experiences, as indicated by its relative importance.
* **Financial Metrics:**
  + While financial metrics like benefit per order have less predictive power in this context, they still contribute to the model and should not be entirely overlooked, particularly in strategies aimed at enhancing profitability or cost management.
* **Sales Strategy:**
  + The lower importance of sales per customer implies that increasing sales alone might not directly influence the operational outcomes considered. Instead, focusing on improving the quality of service and efficiency could yield better results.

This interpretation should guide strategic decisions and resource allocation in addressing the factors most impactful to your model’s predictions, focusing on enhancing shipping efficiency and managing delays effectively.

**Visualizations**

The following charts and visualizations were generated during the analysis:

**Correlation Heatmap of Key Variables**

* + A diagram of a diagram

    Description automatically generated with medium confidence

The correlation heatmap you provided gives insights into the relationships between various supply chain metrics. Here's an interpretation of this heatmap:

### Correlation Heatmap Interpretation

1. **Sales per Customer and Benefit per Order (0.13):**
   * This correlation is relatively low, suggesting that the benefit per order (potentially profit margin or similar financial metric) does not strongly correlate with the sales amount per customer. This indicates that higher sales per customer do not necessarily result in proportionately higher benefits per order.
2. **Days for Shipping (Real) and Shipping Delay (0.61):**
   * There is a moderate positive correlation between the actual days for shipping and shipping delays. This suggests that as the number of real shipping days increases, the likelihood and magnitude of shipping delays also increase. This relationship is significant and implies that longer shipping times could be a predictor of potential delays in the supply chain.
3. **Sales per Customer and Shipping Delay (-0.004), and Benefit per Order and Shipping Delay (-0.0054):**
   * Both correlations are very close to zero, indicating that there is no meaningful relationship between the sales per customer or benefit per order and the shipping delays. This implies that financial metrics such as sales or benefits are independent of the logistical efficiency regarding delays.
4. **Other Low or Insignificant Correlations:**
   * The correlations between Days for Shipping (Real) and both Sales per Customer (0.0018) and Benefit per Order (-0.0051) are negligible. This indicates that the length of the shipping time does not impact the sales volume or profitability per order in any significant way.

### Insights and Implications

* **Operational Focus:** Given the moderate correlation between actual shipping days and shipping delays, efforts to reduce the actual shipping time could significantly impact reducing delays, thereby improving overall supply chain efficiency.
* **Independent Financial Metrics:** Since sales and benefits do not correlate significantly with shipping delays, strategies aimed at increasing sales or benefits per order can be designed without much consideration for their impact on shipping delays.
* **Strategic Planning:** The lack of correlation between the number of shipping days and financial metrics suggests that strategies to enhance profitability or sales efficiency can be optimized independently of efforts to streamline shipping times.

This heatmap provides useful insights for supply chain optimization, emphasizing the need to manage shipping times separately from sales and benefit strategies to minimize delays and improve customer satisfaction.

**KMeans Clustering Visualization**

A chart with colored dots

Description automatically generated

### KMeans Clustering Interpretation

1. **Cluster Distribution:**
   * The plot shows distinct clusters of customers differentiated by their sales and benefit metrics.
   * The yellow and purple clusters near the bottom of the plot represent customer groups with high sales but varying levels of benefits—some experience high benefits (yellow dots towards the left, higher on the y-axis) while others endure lower or even negative benefits (purple dots spread horizontally).
2. **High Sales, Variable Benefits:**
   * The cluster extending horizontally (yellow and purple) highlights a group of customers generating high sales. However, their benefits range from moderately high to significantly negative. This suggests variability in profitability or cost efficiency within this high-sales group.
3. **Low Sales, Moderate to Low Benefits:**
   * The large teal cluster positioned in the upper right part of the plot represents customers with lower sales and consistently low to moderate benefits. This cluster is more homogenous, indicating that customers within this segment behave similarly in terms of purchasing patterns and the financial impact of their orders.
4. **Outliers:**
   * The clusters in the upper left and far right (small teal and dark blue dots) indicate outlier groups with extreme values in either sales or benefits. These outliers may represent special cases or anomalies in the data, such as customers with exceptionally high benefits per order or those with unusually high sales.

### Insights and Implications

* **Targeted Marketing and Sales Strategies:**
  + The clusters with high variability in benefits, despite high sales, might benefit from customized marketing strategies to enhance profitability. This could include better negotiation of terms, targeted upselling, or more efficient order processing and logistics to cut costs.
* **Customer Retention Efforts:**
  + For the large homogenous cluster with moderate benefits, strategies could focus on improving customer satisfaction and loyalty programs to increase their purchasing frequency and potentially move them into higher benefit brackets.
* **Analysis of Outliers:**
  + The outlier groups warrant further investigation to understand the reasons behind their extreme purchasing behaviors and financial outcomes. Insights from these investigations could uncover new opportunities or risks.
* **Resource Allocation:**
  + Understanding these clusters helps in strategically allocating resources, where more focus can be directed towards the most profitable or potentially profitable customer segments to maximize overall returns.

This clustering effectively segments customers based on financial impact and purchasing behavior, providing a foundation for targeted business strategies to enhance profitability and customer engagement.

**PCA Projection Visualization**

A graph with different colored lines

Description automatically generated

### PCA Projection with KMeans Clustering Interpretation

1. **Principal Components Distribution:**
   * Principal Component 1 (x-axis) and Principal Component 2 (y-axis) capture the most significant variances within the dataset. The plot shows a clear separation along the x-axis, indicating that Principal Component 1 is a strong differentiator among the clusters.
2. **Cluster Identification:**
   * Each cluster is colored differently based on the cluster ID, ranging from 0 to 1.75 as indicated by the color bar. This visualization helps to understand how many unique groups exist within the dataset and how distinctly they are separated.
   * The clusters are lined up vertically along different values of Principal Component 1, suggesting that this component captures a critical variance related to cluster differentiation.
3. **Inter-cluster Spacing and Overlap:**
   * There is minimal overlap between clusters, especially those centered around specific values of Principal Component 1. Each cluster occupies a unique position on the x-axis, which suggests that the clustering algorithm effectively grouped the data points based on underlying patterns.
4. **Outliers and Group Density:**
   * A few clusters, particularly those at the extremes of Principal Component 1, have fewer data points (e.g., the cluster at around 3 on the x-axis), which might represent outlier behaviors or less common data profiles in the dataset.
   * Clusters in the center are denser and have more overlap on the y-axis, indicating that while Principal Component 1 differentiates them well, Principal Component 2 might capture less critical, but still relevant, variations within those groups.

### Insights and Implications

* **Targeting Strategies:**
  + The separation along Principal Component 1 suggests it could be associated with a significant business metric or combination of metrics, such as customer value or purchasing frequency. This understanding can guide targeted marketing or customer service strategies focused on specific clusters.
* **Resource Allocation:**
  + Knowing which clusters represent more common versus outlier behaviors allows for more efficient allocation of resources. For example, resource-intensive strategies might be justified for large, central clusters but not for sparse, outlier groups.
* **Further Analysis:**
  + It may be beneficial to perform additional analysis on the features contributing most to Principal Component 1 and 2, as this can provide deeper insights into the factors driving these distinctions among clusters.
* **Customized Interventions:**
  + With clear cluster identification, interventions can be customized for each group based on their unique characteristics, potentially increasing the effectiveness of such efforts in improving customer retention, satisfaction, or profitability.

This PCA projection with KMeans clustering provides a robust framework for understanding complex, multi-dimensional data in a simplified two-dimensional space, aiding in strategic decision-making and operational optimizations.

**ARIMA Forecast of Sales per Customer**

A graph with blue lines

Description automatically generated

A graph with numbers and a bar

Description automatically generated

### Sales per Customer Over Time

This time series graph illustrates how sales per customer have evolved from early 2015 until early 2018. Key observations include:

1. **Stable Sales with Periodic Spikes:**
   * The sales per customer are relatively stable over time, maintaining a baseline that suggests a consistent purchasing behavior among the majority of customers.
   * There are significant spikes in sales per customer occurring periodically, most notably towards the end of the timeline. These spikes could indicate seasonal promotions, holiday sales, or special events that temporarily boost purchasing volumes.
2. **Late 2017 and Early 2018 Trends:**
   * The largest spikes are observed at the very end of 2017 and the beginning of 2018. This suggests an event or external factor had a considerable impact on sales per customer during this period, warranting further investigation to understand the drivers behind this surge.

### ARIMA Forecast of Sales per Customer

The ARIMA model forecast graph plots the observed and predicted sales per customer:

1. **Forecast Time Frame:**
   * The forecast appears to extend well beyond the observed data's timeline, predicting future sales per customer. The x-axis, labeled from 2000 to 2450, likely represents forecasted time points (perhaps mistakenly labeled or scaled).
2. **Observed vs. Forecasted Data:**
   * The observed data shows a sharp peak initially, which might be representative of historical peaks or an aggregation of early data points.
   * The forecasted data points (red dot) suggest a dramatic decline in sales per customer from previous levels, indicating that the model predicts a normalization or significant drop-off from the observed peak.

### Insights and Implications

* **Understanding Spikes in Sales:**
  + Analyzing the causes behind the observed spikes can provide insights into customer behavior and potentially guide marketing and sales strategies to capitalize on these periods.
* **Model Evaluation and Calibration:**
  + The ARIMA model’s prediction showing a decline could be reflective of expected market conditions, or it may indicate the need for model recalibration if it does not align with business expectations or other market analyses.
* **Strategic Planning:**
  + If the forecast is accurate, preparing for lower future sales per customer could involve diversifying sales strategies, enhancing customer engagement practices, or revising inventory and supply chain logistics to adapt to lower demand.
* **Data and Model Verification:**
  + It would be advisable to verify the scaling and labeling of the forecast graph to ensure correct interpretation and application of the model outputs.

These interpretations should help in understanding the underlying trends in sales per customer and assessing the predictive accuracy and implications of the ARIMA model forecast in your strategic planning.

# Recommendations

Based on the analysis, the following recommendations are proposed to improve supply chain efficiency:  
- Focus on optimizing shipping processes for 'Second Class' deliveries, which exhibited the highest delays.  
- Monitor 'Shipping Delay' closely as it is the most important factor contributing to late deliveries.  
- Use predictive models to proactively identify high-risk deliveries and take corrective actions before delays occur.

# Conclusion

This analysis identified critical factors influencing delivery delays, developed predictive models for forecasting delivery risks, and provided actionable insights to enhance overall supply chain efficiency. Leveraging these findings will help reduce late deliveries, improve customer satisfaction, and optimize operational processes.

**Code**

!pip install seaborn

!pip install scikit-learn

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression, Ridge, Lasso

from sklearn.metrics import classification\_report, mean\_squared\_error

from sklearn.preprocessing import StandardScaler

from scipy.stats import chi2\_contingency, f\_oneway

from sklearn.metrics import confusion\_matrix

# Load the data

data = pd.read\_csv('/Users/sayalideshmukh/Desktop/ALY6015/final\_project/DataCoSupplyChainDataset.csv', encoding='ISO-8859-1')

# Convert dates and handle missing values

data['shipping date (DateOrders)'] = pd.to\_datetime(data['shipping date (DateOrders)'], errors='coerce')

data['Order Zipcode'].fillna(data['Order Zipcode'].mode()[0], inplace=True)

data.fillna(data.median(numeric\_only=True), inplace=True)

# Late delivery risk analysis

late\_delivery\_group = data.groupby('Late\_delivery\_risk').agg({

'Sales per customer': 'mean',

'Benefit per order': 'mean',

'Days for shipping (real)': 'mean',

'Days for shipment (scheduled)': 'mean'

}).reset\_index()

# Category sales analysis

category\_sales = data.groupby('Category Name').agg({

'Sales per customer': 'sum',

'Benefit per order': 'sum',

'Late\_delivery\_risk': 'mean'

}).reset\_index()

# Shipping efficiency analysis

data['Shipping Delay'] = data['Days for shipping (real)'] - data['Days for shipment (scheduled)']

shipping\_efficiency = data.groupby('Shipping Mode').agg({

'Shipping Delay': 'mean',

'Sales per customer': 'mean'

}).reset\_index()

# Plot sales and benefit by category

plt.figure(figsize=(10,6))

category\_sales.plot(kind='bar', x='Category Name', y='Sales per customer', legend=False)

plt.title('Sales by Product Category')

plt.ylabel('Sales')

plt.xticks(rotation=90)

plt.show()

# Preprocess Data

data['shipping date (DateOrders)'] = pd.to\_datetime(data['shipping date (DateOrders)'], errors='coerce')

data['Order Zipcode'].fillna(data['Order Zipcode'].mode()[0], inplace=True)

data.fillna(data.median(numeric\_only=True), inplace=True)

# Feature Engineering - Creating new feature for shipping delay

data['Shipping Delay'] = data['Days for shipping (real)'] - data['Days for shipment (scheduled)']

# Visualize shipping delay vs. late delivery risk

sns.boxplot(x='Late\_delivery\_risk', y='Shipping Delay', data=data)

plt.title('Shipping Delay vs Late Delivery Risk')

plt.show()

# Chi-Square Test - Is there a significant relationship between category and late delivery risk?

category\_contingency = pd.crosstab(data['Category Name'], data['Late\_delivery\_risk'])

chi2, p, dof, ex = chi2\_contingency(category\_contingency)

print(f"Chi-Square Test: chi2 = {chi2}, p-value = {p}")

# ANOVA Test - Does 'Days for shipping (real)' vary significantly between different categories?

anova\_result = f\_oneway(

\*[data.loc[data['Category Name'] == category, 'Days for shipping (real)']

for category in data['Category Name'].unique()]

)

print(f"ANOVA Test: F-statistic = {anova\_result.statistic}, p-value = {anova\_result.pvalue}")

# Prepare data for predictive modeling (Logistic Regression for late delivery risk)

X = data[['Sales per customer', 'Benefit per order', 'Days for shipping (real)', 'Shipping Delay']]

y = data['Late\_delivery\_risk']

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scaling the data

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Logistic Regression Model

log\_reg = LogisticRegression()

log\_reg.fit(X\_train\_scaled, y\_train)

# Predictions and Model Evaluation

y\_pred = log\_reg.predict(X\_test\_scaled)

print("Classification Report (Logistic Regression):")

print(classification\_report(y\_test, y\_pred))

# Confusion Matrix

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues')

plt.title('Confusion Matrix - Logistic Regression')

plt.show()

# Ridge Regression Model (Predicting Shipping Delay)

ridge\_reg = Ridge()

ridge\_reg.fit(X\_train\_scaled, y\_train)

y\_pred\_ridge = ridge\_reg.predict(X\_test\_scaled)

rmse\_ridge = np.sqrt(mean\_squared\_error(y\_test, y\_pred\_ridge))

print(f"Ridge Regression RMSE: {rmse\_ridge}")

# Lasso Regression Model

lasso\_reg = Lasso()

lasso\_reg.fit(X\_train\_scaled, y\_train)

y\_pred\_lasso = lasso\_reg.predict(X\_test\_scaled)

rmse\_lasso = np.sqrt(mean\_squared\_error(y\_test, y\_pred\_lasso))

print(f"Lasso Regression RMSE: {rmse\_lasso}")

# Visualize the distribution of errors for Ridge and Lasso models

plt.figure(figsize=(10,5))

sns.histplot(y\_test - y\_pred\_ridge, color='blue', label='Ridge Regression', kde=True)

sns.histplot(y\_test - y\_pred\_lasso, color='green', label='Lasso Regression', kde=True)

plt.title('Error Distribution - Ridge vs Lasso')

plt.legend()

plt.show()