Optimizing Supply Chain Efficiency: Predicting Late Deliveries and Analyzing Performance Patterns with Historical Data

Team Lead - Alex Cundall
Analyst - Isaac Doescher
Analyst - Yaling Wang
Analyst - Arjun Agrawal

Table of Contents

Abstract

This research investigates the route optimization for an e-commerce company, Congolian, by identifying key determinants of late shipments and to develop strategies to increase logistic efficiency. The dataset comprises 180,519 shipment records. Advanced predictive analytics and geospatial analysis were performed to uncover patterns and potential causes of delivery delays. Key methodologies used included classification, random forest, logistic regression and clustering techniques to predict late delivery risks. The results indicate that shipping mode, order region, and store location significantly influence delivery times. Interventions of operation in warehouse and shipment methods could significantly improve delivery timeliness. The research provides insights for Congolian to optimize store distributions, shipping strategies, and in overall supply chain.

Project Plan

Company Profile and Background of Opportunity

Primary Company Details:

Founded - July 14, 2004

Founders - Eric Steinberg, Jeffery Brezo

Headquarters - San Francisco, CA

IPO - Private

Categories - E-Commerce Site, Shopping, Commercial Goods

Address:

308 Negra Arroyo Lane,

San Francisco, CA, 87104

United States

Company Communication:

Phone Number: (415) 831-8799

Fax Number: (415) 733-1849

Website: www.congolion.com

Introduction

As e-commerce continues to go global, optimizing supply chain operations is top of mind for companies who want to meet customer expectations. According to the US International Trade Administration, global B2C e-commerce is projected to grow at 14.4% CAGR and businesses need to get their logistics right. Efficient supply chain management means on time deliveries, cost savings and customer happiness. But logistical inefficiencies, transportation bottlenecks and external disruptions are major hurdles to smooth delivery operations.

This project will analyze Congolian's supply chain data to find the key factors causing late deliveries. Using predictive modeling and geospatial analysis we will get actionable insights to inform logistics decisions. With a dataset of over 180,000 shipment records we will find the top causes of late shipments, evaluate regional delivery performance and propose data driven strategies to optimize supply chain operations. The outcome of this study will help Congolian refine its logistics framework, reduce delivery risks and improve overall customer experience.

Business Description

Congolian is a world-wide e-commerce website dedicated to helping customers get amazing products with fast shipping. With a focus on global operations, Congolian prioritizes fast shipping as well as delivering high revenue to shareholders. They accomplish this by hosting a wide variety of their products on their online web store with global shipping options. With competitive pricing and lower than average global shipping, Congolain takes aim at some of their competitors within the market. Congolian specializes in B2C e-commerce similar to that of Amazon. Selling commercial and luxury goods, their aim is to capture as large of a customer

base as possible with money to spare. Their website features product names, pictures, prices, quantities and status. While they lack a reviews section, they recently expanded faster shipping options to their customers.

Financials:

Ticker Symbol - CON

Latest Financial Report - December 2024

Revenue - $379.34 billion

Assets - $449.45 billion

Liabilities - $67 billion

Return on Equity - (+24.38)%

Earnings - (+3.01)/share

No. of Employees - 990,778 thousand

Key Executives:

Eric Steinberg, 64, Co-Founder, Chief Executive Officer

Jeffery Brezo, 48, Co-Founder, Director, Chief Operating Officer

Ali Williams, 46, Chief Financial Officer

Marcus Levi, 39, Jr. VP of Shipping Logistics

Major Competitors:

Amazon Shopify eBay Target Walmart

Business/Analysis Opportunity

Using a data sample of 180,519 Congolian online business shipment records, our data scientists will analyze key issues in delivery efficiency and provide suggestions for actionable insight. The data sample comprises shipment details, store locations, order quantity, and delivery time, which allows for rich analysis for the cause of late delivery. From machine learning models and geospatial analysis, we expect to identify logistical inefficiencies and supply chain optimization opportunities.

This analysis will spot store distribution gaps, shipping mode performance gaps, and regional gaps. Through the use of historical trend analysis, we will construct predictive analytical models that classify delays in shipments and highlight high-risk delivery zones. These models will be used for making strategic program recommendations, including shipping mode tweaks and store locational optimization. The outcome of this project will assist Congolian in optimizing its logistics network, improving customer satisfaction, and achieving maximum profitability through the elimination of shipment delays and operational inefficiencies.

With the exponential growth of E-commerce businesses, there is an increasing pressure and competition to meet the customers' satisfaction. An optimized supply chain aids to meet the customers' demands, increases operation efficiency, and reduces overhead cost. The e-commerce environment is evolving rapidly: a decade ago, the standard shipping times were 3-7 business days for the norm. Today, industry standard shifts to same day delivery in response to consumer demand, supply chain optimization, and fierce competition. Supply chain optimization will provide an edge in the competition in many ways.

RQ1: What are the most significant predictors of late delivery, and how can we optimize these factors to reduce delays?

It is essential to identify the significant predictors of late deliveries and to form effective strategies to reduce delays. By understanding which variables are the most influential to delivery times, stakeholders can target interventions to mitigate the delays. Ways of mitigation could be making changes in shipping routes, inventory adjustments, or better coordination between internal and external partners and more. Research and analysis of historical delivery data will pinpoint the root cause of late deliveries and refine the management of the supply chain.

RQ2: Which regions experience the highest late delivery rates, and what operational changes could improve performance in these areas?

One of the biggest variables that plays a significant role in affecting delivery performance is location. Certain regions are susceptible to higher rates of late deliveries due to logistic challenges. This research question focuses on identifying the high-risk areas, and resolving the specific regional and infrastructural challenges causing the delivery delays. Operational methods to decrease late delivery could be but not limited to the following methods: building additional warehouses, increasing route optimization, offering collaboration with a third party for difficult routes and more. Understanding the reasons behind this question improves the reliability and efficiency of the delivery process.

Hypotheses

H1: Deliveries from standard ground shipping experience more delays.

Standard ground shipping can experience more delays due to a lower priority assigned and simply a high volume of deliveries.

H2: Warehouses further away from large cities or areas have longer delays.

Less densely populated areas experience longer delays due to larger distance from the warehouse. This hypothesis seems to be logical.

Data

The data used for this analysis has 180,519 records and 53 variables, each representing a single order item rather than an entire purchase. This level of detail will allow us to accurately review shipment performance and trends and pinpoint specific inefficiencies in Congolian's supply chain. Every row of the data set contains a single item in an order, therefore orders with more than one item are spread over more than a single row. This is to mean that shipping profile of every product can be examined separately, thus having a better picture of the delays in delivery.

The data set contains varying aspects of the supply chain, i.e., shipment logistics, geographic customer information, and finance. Among the most crucial variables is "Late_delivery_risk," a binary indicator of whether or not an order was delivered late. It is the main target variable for our predictive modeling exercise. A second most crucial set of variables

is "Days for shipping (real)" and "Days for shipment (scheduled)," with actual and scheduled shipping, respectively, and therefore providing measures on Congolian's shipping deviations.

Geographic variables are also easy to use for monitoring delivery trends region by region. Variables such as "Order City," "Order State," "Order Region," "Customer City," and "Customer Country" provide location-specific insights into shipment destinations. Additionally, "Latitude" and "Longitude" enable precise geospatial mapping, allowing us to visualize areas with frequent shipping delays. This data is essential for identifying regional bottlenecks and proposing solutions, such as optimizing warehouse placements or improving transportation routes.

The data also includes rich product- and order-level information that can influence shipping performance. "Order Item Quantity" and "Order Item Product Price" enable us to explore whether more orders or high-value items are likely to experience shipping delay. "Shipping Mode" (i.e., Standard, Expedited, First-Class) is a very critical variable that will enable us to test the effectiveness of different delivery modes and ascertain whether certain shipment modes are more likely to be delayed.

Financial measures such as "Benefit per order," "Order Profit Per Order," and "Sales per customer" provide us with an idea of the cost-effectiveness of Congolian's shipping. Combining these with delivery performance measures will enable us to determine whether lower-cost shipping means more delays or whether premium services really do make a difference in delivery reliability.

Other categorical order variables such as "Delivery Status" (e.g., Late Delivery, On-Time, Advance Shipping) and "Order Status" (e.g., Complete, Canceled, Processing) provide

us with order fulfillment patterns. Additionally, "order date (DateOrders)" and "shipping date (DateOrders)" allow us to examine delivery time by different seasons and business cycles. For predictive modeling purposes, categorical fields such as "Product Category Id," "Department Name," and "Market" will be encoded to see if certain types of products or markets have a tendency towards delayed deliveries.

Measurements

A key component of any supply chain optimization project is a clear understanding of what exactly is being measured. Given our objectives, efficiency, reliability, and being cost effective are the main focus points. Efficiency explains how well the operations in SC are performing, which involves metrics such as order fulfilment time and inventory turnover rates. Reliability shows how consistently the transportation system meets the expected time for delivery, which is essential to understanding how to avoid interruptions. Lastly, cost effectiveness measures the financial aspects of SC decisions using variables such as transportation cost and inventory storage expense, so that the business can optimize spending.

The dataset used in this project provides information regarding supply chain operations, which will allow us to gather meaningful insights. With transactional data from different stages of the delivery process, we can determine how many different factors, such as location and product type, impact the efficiency of the operations. By integrating these measurements with predictive analysis techniques, we aspire to create strategies that improve SC performance while saving on cost.

Methodology

For research question 1, we aim to identify inefficiencies in the supply chain in order to improve operations. To achieve this, we will utilize our data to review key supply chain metrics like inventory turnover rates, transportation costs and order fulfillment time. By applying machine learning models such as regression and clustering techniques, we will identify patterns in supply chain inefficiencies and be able to make more informed decisions. Also, using historical trends within our data, we can identify common bottlenecks and recommend improved strategies.

For research question 2, we will investigate regions that experience the highest late delivery rates and identify changes we can make in operations. Our analysis will focus on geographical data, and the efficiency of certain store locations. This project will use clustering techniques and heatmaps to visualize trends and identify areas with consistently poor shipping time. Through these patterns in performance by location, we will create specific optimizations, such as adjusting distribution center locations or changing shipping times, to improve reliability in the supply chain.

We will conduct exploratory data analysis to identify the most important factors contributing to delays. In this process, we will visualize trends, look into correlations, and detect outliers that may be impactful to our research. Using these insights, we will create recommendations to optimize delivery operations and reduce inefficiencies in poorly performing regions.

Computational Methods and Outputs

To discover the predictors of late delivery, we will use classification and regression methods so that we can determine the most conclusive factors. Accuracy, recall, and F1-score are important metrics for us to discover precisely how accurately our model is performing, while AUC/ROC curves will assist in determining how our model is identifying on time vs. late deliveries. In order to make sure that our models generalize appropriately, we will employ cross validation methods to mitigate overfitting. By doing this, we will be in a position to optimize our models based on testing performance across different subsets of the entire dataset.

For regional comparison, we will utilize clustering algorithms such as K-means to cluster the regions based on their late delivery rates. We can observe the regions with the delivery inefficiencies from the cluster plots. Decision tree models will also help in identifying factors that are responsible for this in some regions. If needed, we will modify our strategy for the detection of high risk areas so that we can be able to derive concrete insights.

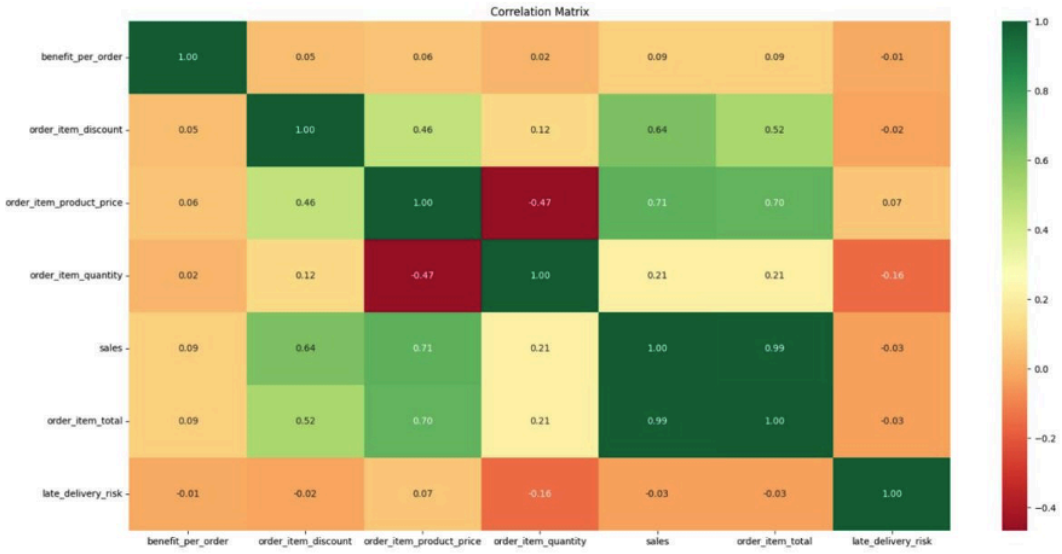
Our answer to research question 1 will be a ranked list of the most predictive causes of late delivery, and a model that we can use to estimate the probability or chance of a late delivery. For research question 2, we will generate a regional risk analysis, which will show the most affected regions by late deliveries and give us insight into possible operation improvements. For being able to make useful suggestions, we will monitor confidence levels through our predictions and provide figures which can aid the stakeholders in identifying and utilizing our findings in an optimal manner.

Output Summaries

RQ 1: What are the most significant predictors of late delivery, and how can we optimize these factors to reduce delays?

The analysis for this research question will focus on identifying and ranking the key predictors of late deliveries. Through machine learning models such as logistic regression and random forests, we will quantify the impact of various shipment attributes, including shipping mode, store location, and order characteristics. By leveraging feature importance analysis, we will determine which factors contribute the most to delivery delays.

The primary output will be a detailed summary table highlighting the top predictive factors of late deliveries. This table will rank the variables based on their influence on shipment delays, providing actionable insights for supply chain optimization. Additionally, we will present visualizations such as bar charts and correlation matrices to illustrate the relationships between different shipment attributes and late delivery risk. These outputs will serve as the foundation for strategic recommendations aimed at minimizing shipment delays. The correlation matrix below, from RIT, gives an example of an output we intend to create.



Research Question 2: Which regions experience the highest late delivery rates, and what operational changes could improve performance in these areas?

To address this research question, we will conduct a geospatial analysis of late delivery occurrences across different regions. By mapping out shipment delays using heatmaps and spatial clustering techniques, we will identify high-risk delivery zones that require operational improvements. Factors such as store density, regional infrastructure, and shipment volume will be analyzed to determine why certain areas experience higher delay rates.

The output for this research question will include a series of geospatial visualizations showcasing delivery hotspots and regional inefficiencies. These visualizations will be accompanied by a regional performance report that provides data-driven recommendations for improving delivery efficiency. Suggestions may include opening new stores in underserved areas, reallocating inventory to reduce transit times, or adjusting delivery schedules to better accommodate regional demand patterns.

Campaign Implementation

Implementation strategy will be focused on providing actionable information to Congolian's logistics personnel in order to reduce delays and optimize shipping performance. Predictive modeling will be integrated into the logistics process to forecast high-risk shipments ahead of delay, allowing intervention before. With machine learning-based delay predictions, logistics personnel are able to pick out shipments most likely to arrive late and act ahead of time such as by rerouting shipments, rescheduling delivery times, or stockpiling at strategic store locations.

Operational improvements will include maximizing store location based on geospatial analysis findings. Through identification of areas with high rates of delays, Congolian can ascertain the feasibility of new stores in under-represented areas in a bid to reduce shipping distances. Additionally, Refining fulfillment strategies such as accelerating processing for high-risk zones will go on to optimize operations. Optimization of shipping mode allocations based on data will ensure that expedited shipping is employed strategically to minimize delays at an affordable cost.

Customer satisfaction will be enhanced by optimizing communication strategies. By integrating predictive analytics with real-time monitoring, customers will be provided with real-time delivery status updates dynamically, reducing uncertainty regarding the arrival of shipments. Moreover, logistics staff will be equipped with delay risk dashboards providing real-time visibility into delivery performance, allowing real-time intervention when issues arise. These predictive modeling and operations-driven enhancements will result in increased cost savings, customer retention, and overall supply chain responsiveness.

Literature Review

According to the United States International Trade Administration global B2C e-commerce revenue is projected to grow at a 14.4% compound annual growth rate (International Trade Administration). As customers spend more capital on consumer goods e-commerce sites have increased their efforts on supply chain optimization. While this includes practical elements, technological ones are primarily the focus. With data garnered from transactions, companies employ data mining, machine learning algorithms, and a variety of data science methods to increase delivery expediency and quality of service. Lead among these are Integer Linear Programming, Deep learning, and various Machine Learning techniques.

Retailers face critical decision-making about inventory management and the optimal time an order enters the supply chain to ensure efficient delivery. This is known as the E-commerce Shipping Problem (ESP). Each period, the retailer must make the decision whether to ship out the order or to keep it in the inventory longer depending on the priority of each order. Further analysis and decision-making is required to determine which level the order leaves the sorting centers and enters the final delivery points (FDP) where the retailer hands over deliveries to a third party logistic company such as FedEx, UPS, or USPS. E-commerce optimization increases customer satisfaction, cost management, reduces supply chain complexity, and has a competitive advantage.

According to Ponce et al., the Periodic Vehicle Routing Problem (PVRP) is pivotal in optimizing e-commerce logistics. There are two types: static and dynamic PVRPs. Orders are processed and delivery decisions are made after a period of time in static PVRP. Route optimization takes place after all orders are received in that period which Ponce et al. used Integer Linear Programming (ILP) to optimize delivery routes. In dynamic PVRP, the orders are

dealt with in real-time. Route planning constantly gets updated for the best optimized route. Four strategies were tested in California and Texas in the dynamic scenario:

1. Threshold: Makes shipping decisions based on specific criteria (a threshold).
2. Optimistic: Assumes the best case for delivery times and costs.
3. Pessimistic: Plans for the worst-case scenario, often delaying shipments to mitigate risks.
4. Minimax Regret: Tries to minimize the worst possible outcome of shipping decisions.

The study found the best strategy of dynamic PVRP are the threshold and optimistic policies. As these methods aid in dealing with the delivery process, Deep Learning is also utilized.

With the evolving need for more advanced predictive analytics regarding late deliveries in supply chains (SC), a study by Gabellini (2024) created a Long Short - Term Memory (LSTM) deep learning model to address this. Before this, most models mainly used logistic regression and decision trees, but this study utilized LSTMs to find sequential patterns in SCs data in order to predict the exact delay duration instead of classifying orders as “on time” or “late”. The biggest achievement of this study was its ability to incorporate economic indicators like inflation rates, and supply chain disruptions to have a better understanding of delivery risk based on time. This deep learning model’s key benefits are:

- The ability to produce higher accuracy in predicting delivery delays compared to former models.
- The ability to capture long term dependencies in SC performance from sequential data.
- The ability to use economic factors to create a broader view of SC trends.

This study goes hand in hand with the other information in this literature review by offering a “time series” approach that includes external economical factors and historical trends in its analysis. Although, the model performs poorly at guaranteeing actionable insights for SC

regardless of how strong its accuracy is. This is due to external factors which are out of the control of any specific SC operation. In addition, the focus on external factors means that it does not provide region specific insights thereby making it less useful in the pursuit to answer the second research question. Overall, the model has some limitations while considering the goals of this project, but it highlights the potential for improving SC analysis using deep learning to gain more precise prediction of shipping delays.

As the literature above describes the context for methods and delves into practices used to optimize aspects of the supply chain it is important to mention other work associated with the dataset chosen. In a thesis for his Master's at Rochester Institute of Technology, Mohammad Zubin Siddiqui focused on using machine learning techniques to predict late delivery analysis and demand forecasting (Siddiqui 54). Siddiqui used classification models such as logistic regression, decision trees, random forests and other secondary methods to accomplish these tasks. Building on this prior work, we hope to continue the research within this topic and expand on it further with our findings. However it is recommended to view his work as it directly correlates to ours. Given how instrumental machine learning was in his research we find it imperative to build upon his work done on these topics.

Final Research Questions

We will continue with the two research questions of our research after the preparatory work. These two questions will be analyzed to uncover potential insights that will forward our study:

Research Question 1: What are the most significant predictors of late delivery, and how can we optimize these factors to reduce delays? We will use multiple machine learning methods to identify these predictors from our shipping data.

Research Question 2: Which regions experience the highest late delivery rates, and what operational changes could improve performance in these areas? Our analysis will focus on regional data where we analyze to find the area of concern.

Exploratory Data Analysis

For this project we obtained data from kaggle.com using a dataset called DataCo Smart Supply Chain for Big Data Analysis. This is a structured dataset consisting of 180,519 observations related to specific details in supply chain operations. There is plenty of information included in this that pertains to our objective of identifying factors that contribute to delivery delays and the locations that experience these delays most strongly. Originally it contained 53 variables, but many of those were not necessary for answering our research questions. These variables were regarding customer email addresses, passwords, and other obviously irrelevant information. After understanding and cleaning the dataset to narrow down the features to what is necessary for our research, we ended up with these:

late_delivery_risk: A binary value from 0 to 1 detailing if an order is late.

days_for_shipping_real: Actual shipping days of the ordered product.

days_for_shipment_scheduled: Scheduled time for delivery of the product.

order_state: State where the order is delivered.

order_city: City where the order is delivered.

order_country: Country where the order is delivered.

order_region: A string-based value of where the customer ordered from.

shipping_date: A string-based value of exact time and date from when the order was shipped.

shipping_mode: A string with values Standard Class, First Class, Second Class, Same Day notating the method of which an order was shipped.

order_item_quantity: Integer value of the number of items within an order.

latitude_src: Float value of where the item was shipped from.

longitude_src: Float value of where the item was shipped from.

latitude_src: Float value of where the order is delivered.

longitude_src: Float value of where the order is delivered.

We then made two new variables to leverage the maximum potential this dataset holds, which assisted us while working to answer our research questions. Those variables are:

delivery_delay: Integer value that calculates the difference between days for shipping real and days for shipment scheduled.

delivery_distance: Using latitude and longitude source compared to latitude and longitude of the destination we found the distance of each delivery.

The image in appendix A shows the distribution of delivery delay by the number of days. We want to look further into the late delivery risk by the counts of the numbers of days late to identify problematic inefficiencies. The image showed the highest count of late deliveries being one-day late with approximately 60,000 counts. The second highest being on-time deliveries of approximately 33,000. Two-day late deliveries have a count of about 28,000. It is surprising to see early deliveries with about 21,000 counts for both two-day and one-day early while three and

four-days late has counts of about 6,000, which are the lowest of the groups. This trend was unexpected as we expect the higher delay days to have the most count. It's astonishing to observe the highest count being one-day late. Further analysis is necessary to seek the root cause of these minor delays.

The appendix B graph shows Average Deviation Days from Month where it measures how much actual times differ from the expected time. This graph showing monthly deviation may help to identify possible seasonal trends. The highest deviation occurs in February and August with a deviation of 0.580 and 0.577. The lowest deviation occurs in October of 0.553. The months march through July and December through January have relatively consistent deviations. Although the graph showed a prominent difference between the highest and the lowest, the range of Average Deviation is narrow with values from 0.555 to 0.580. The narrow deviation range indicates month to be an insignificant variable. Further analysis may have better insights.

While the monthly deviation graph shows how much delivery times deviate from expectations, it does not completely illustrate how delivery delays change over time. Appendix E graph illustrates the trend of average delivery delays from 2015 to early 2018, allowing us to look into seasonality patterns. From this graph we see roughly 2-3 major recurring and cyclical spikes and drops each every year suggesting that delivery delays follow a seasonal pattern. This could be influenced by holiday splurge shopping, supply chain issues, or carrier capacity constraints. Another point noting, the lack of a clear upward or downward trend over the years

suggest that delays are due to short-term rather than long-term factors. Further study is needed to determine what causes these cyclical trends.

To investigate the late delivery risk by shipping mode, a barchart was created in Python with the library matplotlib, appendix C. Late deliveries were marked red with the occurrence of "1" within the data while deliveries on time were marked blue with a "0". The x axis illustrates the different shipping modes which are Standard Class, First Class, Second Class, Same Day. The Y-axis shows the count of deliveries. Standard class is the most commonly used shipping mode and this is also shown by having the most on-time and late-deliveries. As shown in the illustration, standard class leads the category with late deliveries with the total being 41,023. First Class in second with 26,987, Second Class in third with 26,513, and Same Day with 4,454. Interestingly, first class and second class look to be the most prone to a late delivery as the number of late deliveries far surpasses the amount of normal deliveries.

Given our data it was important to find which regions experience the highest late delivery rates then research operational changes. To aid in this visualization charts were constructed in Python and Tableau under appendix F . In Tableau, for late delivery by region, orders marked with "1" in late delivery were funneled into regional categories and totaled. As noted in the graph, the top five regions for late deliveries are Central America at 15,518, Western Europe at 15,140, South America at 8,111, Oceania at 5,482, and Southeast Asia at 5,297. With such a distribution of high late deliveries, the team examined top cities with late deliveries.

Coded in Python, appendix G shows cities with late deliveries that were filtered and sorted to the top fifteen. In addition, cities with a delivery of less than 100 orders were filtered out. While we were expecting to find cities concentrated within the highest regions to be among the top, the results were surprisingly diverse. Given this information, the data seemed to indicate that late deliveries were spread across a region rather than concentrated within several cities. This then prompted an investigation into the store location of an order the distance it had to travel to a customer.

In research on the same dataset, Kaggle user Om Gupta used Python with the libraries geopandas and matplotlib to employ a geospatial map, appendix H. With aid of these libraries Gupta plotted deliveries with a Source to Destination route. Here it becomes clear that many of the source points are within the United States regions with destinations outside of the corresponding areas. Having this modeled, source destination points are shown to be a significant factor in late–deliveries and prompts for an overhaul of the current shipping system.

The visualization, appendix I, shows the state to which the stores with the most late deliveries belong, this gives crucial insights into regions with operational inefficiencies. Puerto Rico (PR) has significantly higher late deliveries compared to other states. It is followed by California (CA) and then New York (NY). From this, we understand that it suggests potential issues such as transportation inefficiencies, high demand, or supply chain disruptions in these locations. This gives us a high level overview of some possible underperforming regions and can lead us into investigating on a more granular level. From this we can research root causes in our

data and create recommendations to optimize processes, adjusting inventory counts, and finding the best locations for alternative store locations and possible places to create warehouses to centralize operations.

The visualization from appendix J shows a map of store locations across the US, including some in South America. This is an important output because it gives us a detailed view of where the majority of the stores in our dataset are located. In the US, some densely populated locations, such as Los Angeles, New York, and Chicago, have higher concentrations of stores. This could relate with higher demands on products and orders which in turn creates more logistic problems. Also, the stores that correspond to less dense areas may face their own unique problems. First, being in more remote locations, these stores most likely have to deal with longer transportation routes from store to delivery location. Second, they could hold less inventory stock and face longer wait times to restock their top selling items before getting them out to customers. Overall, this graphic assists us in understanding regional performance, identifying areas that might need improvements in infrastructure or strategic employment of distribution centers.

The next visualization, appendix K, maps the locations of all stores worldwide, which gives us an understanding of global operations. At first glance, the concentration of stores in North America should lead us to believe that the company has a focus on this market in particular. Although, from the output in appendix L, we see that only ~14% of total orders are shipped to the US, while the other 86% are filled worldwide. This suggests that most of the business operations and demand from customers is driven by international factors. This finding led us to

pursue optimizing the operations of shipping logistics in the US while keeping in mind ideas for expansion across the globe. Understanding this global dilemma is crucial for our research to improve supply chain logistics, ensuring efficiency regarding international deliveries, and identifying areas to place warehouses based on customer demand. Overall, this gives us good insights to make valuable recommendations for expanding and creating centralized distribution centers inside and outside the US, and improving regional networks to accommodate all customers.

Methodology

RQ1 - What are the most significant predictors of late delivery, and how can we optimize these factors to reduce delays?

Predictive Modeling

In order to discover the most key predictors of late delivery, we used a supervised learning approach by creating multiple classification models to identify the best performing one. Our primary model is a Random Forest classifier, because of its ability to provide feature importance rankings, find nonlinear correlations, and since it can handle high dimensional data. This method is a binary classification, which uses a target variable “Late Delivery: Yes / No”, then uses historical data to find patterns that contribute to delayed delivery times.

Supply chain operations can be complex and highly detailed, so late deliveries can be influenced by many factors such as the quantity of items ordered, the type of product ordered, the day of the week order was shipped, mode of shipping, and more. Our goal is to utilize machine learning models that are capable of identifying many different relationships while giving us accurate predictions and proving which factors are the most influential.

Preparation

In our research we used a supply chain operations dataset which contains 180,519 observations. This includes historical order information, details about each order, details for the products, shipping times, delivery location and more. To prepare this data for modeling we had to execute some specific tactics.

First, we created new variables to leverage the dataset to its best potential. This included the calculation of delivery delay, which subtracted the real delivery time from the expected, and

the shipment distance of orders. Also, we grouped observations by shipping priority and product type.

Upon obtaining our dataset we recognized some flaws, firstly there were columns containing all missing values or some missing values and these were removed. Next, we removed variables that were completely unrelated to our goals, such as customer email and customer password. On top of this, there were a few columns that were oddly written in Spanish, so these were converted into English so that we could utilize them properly.

For the purposes of our predictive modeling, we used one hot encoding upon the categorical variables to ensure that they are compatible with the models we created. Also, continuous numerical features were standardized using the MinMax scaling technique to enhance the performance of our model.

Modeling Techniques

Each of these models were trained using an 80 - 20 train test split, so that the metrics we receive ensure real world ability to predict factors. To tune the models, we improved hyperparameters using GridSearch to find the best parameters based on the performance of our validation set.

- Random Forest
    - A learning method that uses multiple decision trees and combines the outputs to enhance the accuracy of predictions.
    - Provides importance scores for the features, allowing us to rank them to find the best late delivery influencers.

    - Selects subsets randomly to train each of the trees which improves generalization and reduces overfitting the model.
- Logistic Regression
    - This model is used as our baseline for comparison, because it is simple and easily interpreted.
    - Estimates the chance of late delivery by weighting the sum of input variables.
- XGBoost
    - This algorithm uses advanced gradient boosting that targets the poor performing trees by lowering the errors between each step.
    - Is known for handling missing values well and performing efficient selection of features.
- K-Nearest Neighbors
    - Based on the majority of closest data this model classifies new orders, which creates a well performing tactic for recognizing patterns but can be affected by imbalanced data.

RQ2 - Which regions experience the highest late delivery rates, and what operational changes could improve performance in these areas?

Spatial Analysis

As the regions that experience the highest late delivery rate were discovered in the exploratory data analysis, contributing features are far more vast. Before commencing on this research, data is first filtered on five regions with the highest late deliveries. This reduces the total amount of late deliveries from 98,977 to 49,548. After this, our focus is then split into three

different categories: regional analysis, store functionality, and the relationship between store and customer.

In regional analysis, statistical testing is employed to see if there are statistically significant delivery times between regions. For this a T-Test is chosen with the standard alpha of .05. For further research, a Chi-Square Test is utilized to see if international orders have statistically significant higher late delivery rates with the same alpha of .05. By identifying potential outliers, analysis of core variables and their interactivity with each other becomes less obstructed.

After analyzing regions, clustering is used to analyze store's shipping functions. K-Means clustering is used to identify underperforming stores in their shipping features. K-Means clustering is chosen for its practicality and scalability with data such as this size. Hierarchical Clustering is then used to see if shipping patterns emerge from different stores.

Finally for the research of the relationship between store and customer, multiple regression is used to find if longer shipping distances lead to more late deliveries. By building off of Om Gupta's work as previously mentioned in our exploratory data analysis, we are able to measure the distance between where an order is placed and its shipping destination. Thereby, the metrics for multiple regression include store late deliveries, distance, and region as predictors.

Analysis

Predictive Modeling

To identify key predictors of late delivery risk, we examined a Pearson correlation matrix to get a better understanding of relationships between the numerical features. This revealed a moderate positive correlation of 0.61 between shipping delay and days_for_shipping_real, which confirms that as actual shipping duration increases, the likelihood of delay also rises. Also, days_for_shipment_scheduled showed a moderate correlation of 0.52 with days_for_shipping_real, proving to our research that discrepancies between scheduled and actual shipping times are a contributor to late deliveries.

Another notable finding is in regards to shipping_mode, which shows a 0.57 correlation with days_for_shipping_real and a nearly perfect correlation with days_for_shipment_scheduled. This suggests that different shipping methods have a substantial influence on delivery performance.

To prevent issues with multicollinearity in our models, we eliminated redundant predictors. For example, order_item_total and sales showed an extremely high correlation of 0.99, allowing us to drop one of them. Also, features with no correlation with late_delivery_risk, such as benefit_per_order, were dropped from the analysis.

After refining our features, we trained several machine learning models to predict late delivery risk, which included a Random Forest, XGBoost, Logistic Regression, and K-Nearest Neighbors. First we analyzed each of their confusion matrices to understand the performance of the classification, by looking into true positives, false positives, true negatives, and false negatives. These insights helped assess the misclassification rates for each model. Then, we assessed the performance of each model using classification reports showing us accuracy recall

and F1 scores. XGBoost achieved the highest accuracy at 97.52%, closely followed by Random Forest which scored 95.69%. The other models' performance, KNN and Logistic Regression underperformed immensely by comparison. Because of this, we chose XGBoost and Random Forest to continue with our analysis by applying hyperparameter tuning to find the best model parameters. Although, this did not boost accuracy very much for either of the two models.

To interpret our models further, we created a feature importance ranking chart for our two selected models to show the top predictors. Days_for_shipment scheduled was the top ranking variable between both of the models, confirming its strong correlation with the target variable. Also, shipping_day_of_week had a strong impact despite its low correlation in the initial analysis. Finally, order_day_of_week performed surprisingly well, leading to further investigation. Interestingly, after hyperparameter tuning, days_for_shipment_scheduled increased in significance confirming its reliability.

To investigate how effective our models are at distinguishing between late deliveries and on time deliveries, we compared the AUC/ROC curves of the two models. The XGBoost model outperformed the Random Forest, as expected, furthering our confidence that this is the best predictive model. On top of this, we conducted a SHAP (SHapley Additive exPlanations) analysis to look into how our individual features affected the predictions in a positive or negative manner regarding the likelihood of predicting a late delivery. These values further confirmed days_for_shipment_scheduled was a dominant feature. Also, order_day_of_week was strong, proving that the day of the week an order was placed can influence shipping performance. Interestingly, shipping_day_of_week was the most influential feature from this analysis, telling us that the day an order was shipped has a significant impact on whether an order will be late.

Clustering

Clustering is used to analyze stores' shipping functions. K-Means clustering is used to identify underperforming stores in their shipping features. K-Means clustering is chosen for its practicality and scalability with data such as this size. Hierarchical Clustering is then used to see if shipping patterns emerge from different stores.

To perform this, the data was loaded in as a pandas dataframe. As previously mentioned, column 'Late_delivery_risk' was converted into integer type while column 'delivery_tim_diff' was calculated by subtracting column 'days_for_shipping_real' by column 'days_for_shipment_scheduled'. Then relevant columns were selected for store performance. These columns 'late_delivery_risk', 'days_for_shipping_real', 'delivery_time_diff' were added to a list variable named 'store_features'. To handle missing values in the relevant columns, median values were filled in the dataframe by accessing the data frame columns via the list. Using SciKit Learn's preprocessing class, the standard scaler function was imported. Using the scaler, a new pandas dataframe 'df_scaled' was created from the relevant column list. To determine the optimal number of clusters using the Elbow Method, a for-loop was created to cycle through to find the optimal number of k's. With a range of 1 to 11, the for-loop would cycle through different SciKit Learn KMeans models with an incrementing number of inertias. By using these different models, they were then implemented into a python plot illustrating the elbow method for k-selection. This plot was created with the matplotlib.pyplot library and was essential to visualize the ideal number of inertias. Here based on the elbow method and visualization, the ideal number of clusters was decided to be 3. Using the same SciKit Learn KMeans library, a new model was created with the optimal number of clusters, a dedicated random state of 42, and

an ‘n’ initialization of 1. With this model a new column named ‘cluster’ was added to the original dataframe.

To analyze these clusters, a new variable ‘cluster_means’ was created based on the original dataframe grouped by the ‘cluster columns' and the relevant ‘store_features’ columns. These variables were then printed out. Cluster 0 has a high late delivery risk (0.97), a longer shipping duration (5.35 days), and a positive delivery time difference (2.01 days), illustrating that these stores have a lot more late deliveries and take a lot longer to ship than the expected normal. Cluster 2 also has a high late delivery risk (1.00), but a shorter shipping duration (2.06 days) and a lower delivery time difference (1.00 day), indicating that these stores also experience late deliveries but in a shorter time frame. In contrast, cluster 1 has a late delivery risk of 0.00, meaning that orders in this group are delivered on time. The average shipping duration is 2.72 days, and the delivery time difference is -0.77 days, showing that deliveries arrive ahead of schedule. These results show that cluster 1 represents the regions with efficient logistic operations while Cluster 0 and Cluster 2 are high-risk groups with frequent late deliveries. We should look into analyzing these clusters to understand and be able to replicate the success of cluster 1.With these results, further visualization was desired.

A new dataframe, ‘df_sample’, was created based on 5,000 total observations with a random state of 42. Using the seaborn and matplotlib.plotly libraries, a scatter plot was created with the new sampled data. By using ‘df_sample’ as data, column ‘days_for_shipping_real’ as the x-value, and ‘delivery_time_diff’ as the y-value the data was allowed to be visualized. The scatterplot shows the relationship between actual shipping time and delivery time difference across the clusters. Delivery time difference was calculated by subtracting the scheduled shipping time from actual shipping time. Each point represents a clustered observation with

different colors distinguishing between the clusters. From the graph we see that Cluster 0 has the longest shipping times, ranging from 4 to 6 days. The observations also are all positive for difference in delivery time between 1 to 4 days showing that the orders consistently are late. Cluster 2 has shorter delivery days between 1 and 3 while all being positive as well for difference in delivery time by around 1 day, showing minor delays. Cluster 1 in contrast seems to be the best performing group. The cluster has a low difference in delivery time between -2 and 0 days, meaning the deliveries are either on time or ahead of schedule. The cluster also has short shipping times mostly under 3 days. This illustrates that cluster 0 should be prioritized in improving operational efficiency, while cluster 2 can be slightly refined, and cluster 1 replicated where possible for most efficiency.

Desiring to visualize how clusters compare in number of unique store locations, variable ‘store_counts_by_cluster’ was created by mapping value counts and sorting the index. Using the previously mentioned visualization libraries a simple bar chart was created. cluster 1 has the most number of unique store locations of nearly 6000, followed by cluster 0 at around 4000 and cluster 2 at around 2000 locations. This could indicate that cluster 2 and 0 having a smaller amount of locations are a specialized group of locations that differ significantly from cluster 1. Cluster 1 having the most amount of locations can potentially be a benchmark for an average-performing group. To examine which states have the most underperforming stores and how they are distributed across clusters, we analyzed the top ten most underperforming states. This analysis provides insights into regional logistical challenges, which can facilitate targeted operational improvements. By counting the top 10 underperforming states and grouping store locations per state by customer via the latitude source variable, a bar chart was constructed using the previously mentioned visualization libraries. stores in Cluster 0 (Slow Shipping & Late) face

issues that may include initial logistic processes such as warehousing and dispatch, whereas Cluster 2 (Fast Shipping but Late) highlights inefficiencies in last-mile delivery. Puerto Rico has an exponentially high count in both Cluster 0 and Cluster 2 with a total of almost 8000 stores, indicating severe inefficiencies and logistical problems. This high count is followed by California with a total of approximately 2000 stores.

Regional analysis

For regional analysis, data was read into a pandas dataframe. Then column 'Late_delivery_risk' was converted into integer type while column 'delivery_tim_diff' was calculated by subtracting column 'days_for_shipping_real' by column 'days_fo_shipment_scheduled'. After these two columns were modified, the column 'order_region' was then checked for a total number of unique variables. Using libraries pandas, matplotlib.pyplot, and scipy.stats, multiple statistical analysis was then performed.A Chi-Square Test was employed to see if there was a statistically significant relationship between categorical columns 'order_region' and 'Late_delivery_risk'.After using the chi2_contingency function from the spipy.stats class the calculated p-value was 0.00000011. With a standard alpha of .05, this result indicates a strong significance that late delivery rates are correlated to the region of delivery. In addition, the Chi-Square value was 74.9336, thereby supporting evidence against the null hypothesis.

In continuation of regional analysis, a plot using the matplot.lib library was constructed to visualize which regions experienced the highest rates of late delivery. With our pandas dataframe being the data, x being the order region, and y being the number of late deliveries, it became apparent that Western Europe and Central America have the highest risk of late deliveries. In order to identify why regions Western Europe and Central America experience the

highest rates of late delivery, the Kruskal-Wallis test was utilized. By still utilizing the scipy.stats class, the function kruskal was imported and used to explore the differences in distribution of delivery time differences, columns ‘delivery_time_diff’ by ‘order_region.’ The test yielded a high H-statistic value of 60.9680 and an extremely low p-value of 0.00001605, indicating statistically significant differences in median delivery times across regions. The range and median are similar across all regions except for four. Using libraries seaborn and matplotlib.pyplot a boxplot was constructed. Similar to the Chi-Square Test, data was our pandas dataframe, x was the column ‘order_region’ and y was the column ‘delivery_time_diff’. The boxplot visualization illustrates Southern Africa and Canada having the lowest median of 0 days although the range of delivery times varies significantly (from -2 to 4 days) which translates to inconsistent delivery rates. Additionally, Central Asia and South of the USA not only showed a wider range of delivery time difference but also had the highest median of one day.

Finally, a T-Test was applied to compare U.S. and global delivery performance. To accomplish this, U.S. regions were defined as variable “us_regions” within a list with string values “West of USA”, “US Center”, “East of USA”, and “South of USA”. Then we created a new dataframe with the name “us_group” by filtering order_regions in the main dataframe by the predefined list mentioned above. After creating “us_group”, another dataframe “global_group” was created by filtering order_regions by all values not within the “us_regions” list. Having these two separate dataframes a T-Test was employed. Still visualizing by libraries seaborn and matplotlib.pyplot, another boxplot was constructed to illustrate results. Overall, the test yielded a T-Statistic of 1.1016 and p-value of 0.2707.T he p-value is significantly greater than the threshold value of 0.05 indicating there is no significant difference in the mean delivery times between U.S. and international shipments. With the box plot supporting the results of the T-Test,

it is suggested that the variability of delivery times are comparable. Therefore, we can conclude that in terms of delivery time differences, the US and the rest of the world experiences are relatively similar.

Relationship of Distance between Store and Customer

To analyze the relationship between store, customer and distance, data was imported into a secondary JupyterLab python notebook. After importing the data as a pandas dataframe, the distance between the store and item delivery for every order was calculated. This was accomplished by utilizing the Haversine formula, a mathematical formula that derives the distance between two points on a sphere given their longitudes and latitudes. This calculation was added as a new column within the dataframe as ‘distance’. After this, relevant categorical variables were encoded as numerical variables with SciKit Learn label encoder. Then, irrelevant variables such customer name, customer email, were then dropped from the dataframe. Needing a target variable, late_delivery_risk was then dropped from the dataframe and was moved to a new dataframe ‘y’. After trimming the data frame, SciKit Learn’s train test split function was then used to create dataframes X_train,X_test,y_train,y_test with 30% of the data being used to train future models. This split was done to prevent overfitting. With the data split and in corresponding X and y dataframes, a random forest classifier was then deployed on the training data, the result was shocking with a perfect F1 score of ‘1’. Due to concerns of overfitting, this data was then run through a cross validation score from SciKit Learn and still received a perfect ‘1’ score.

With training data split, the linear regression class was imported from the sklearn.linear_model class. Following standard procedure, it was then fed the corresponding X train and y train datasets. After model creation and testing, accuracy was initially measured with

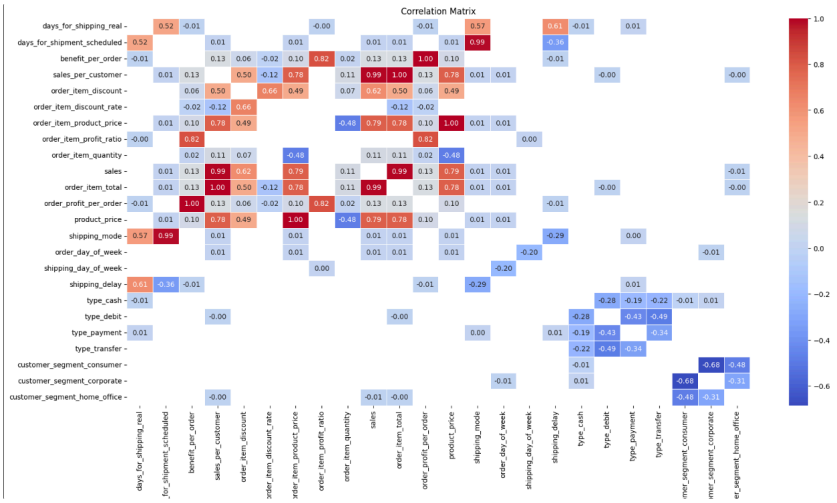
an R2 score. With distance as a factor, the model had an R2 score of 0.7773962805747521. Without distance as a factor, the model had a higher R2 score of 0.777396816883777. Desiring a more scientific comparison, statsmodels api was imported and the regression model was retested through their framework. Here a detailed list of factors and coefficients were detailed and new insights to be derived. In this model, distance was not a significant factor in the relationship between store and customer. However, many team members were curious to see what information a more specialized test might derive.

A multi-layer perceptron classifier was then implemented via SciKit Learn's neural network class. After rigorous testing, the team devised a neural network with a max iteration of 5000, adam as a solver, relu as the activation, a random state of 1, and 4 hidden layers. These hidden layers had sizes 17, 10, 10, 5 respectively and were chosen through testing via SciKit Learn's gridsearch function. Accuracy was measured using SciKit Learn's accuracy function from their metrics class. With distance as a variable, the neural network had an accuracy of 98.22%. Without distance as a variable, accuracy was at 98.96% accuracy. After these two separate evaluations, it was ruled that distance was not a significant factor in the relationship between store and customer.

Data Visualizations

Predictive Classification Modeling

To begin, we created a Pearson correlation matrix to find the numerical features that have significant relationships with late delivery and other shipping relating features. Correlations with p-values of less than 0.05 are masked in white. There is a moderate-to-strong positive correlation of 0.61 between shipping delay and days_for_shipping_real, indicating that as the actual shipping duration increases, the probability of having a delay also increases.



Days_for_shipment_scheduled has a positive moderate linear correlation of 0.52 with days_for_shipping_real, showing that discrepancies between scheduled and actual shipping contribute to delays. Another key feature is shipping_mode which has a positive moderate linear correlation of 0.57 with days_for_shipping_real and 0.99 with days_for shipment_scheduled, showing that shipping_modes heavily influence the probability of a delay.

This correlation matrix also allows us to filter out redundant predictors that might cause multicollinearity. For instance order_item_total and sales exhibit an extremely high correlation of

0.99, suggesting that one of these variables can be dropped. Features that have no correlation with late_delivery_risk such as order_day_of_week (-0.01) and shipping_day_of_week (-0.01) can also be excluded from our predictive models.

After we selected the correct variables to use in our research, we created multiple classification models to predict late delivery risk. To begin, we trained several machine learning models such as Random Forest, XGBOOST, Logistic Regression, and K - Nearest Neighbors. The first outputs we created while doing this were confusion matrices for each model shown in Appendix M. These allowed us to visualize the performance of the classification by looking into true positives, false positives, true negatives, and false negatives, giving us insights into misclassification rates for each.

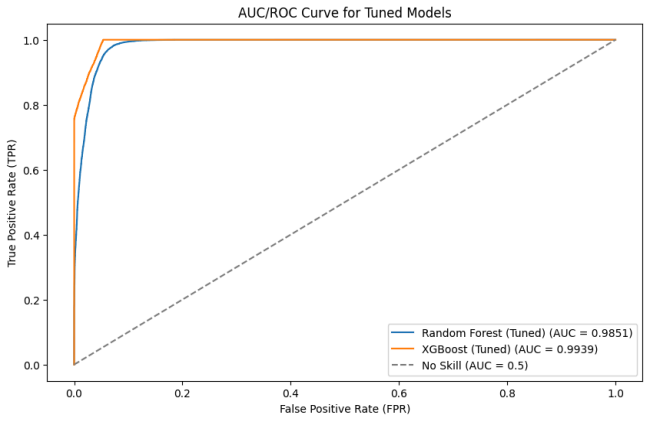
The next output we received from these tests was classification reports for each of the four models, this included metrics regarding accuracy, recall, and F-1 scores shown in appendix N. From these numbers, we were able to determine the best methods for predictive power and possible improvements to be made. These reports revealed that the XGBoost and Random Forest models gave us the best initial performance yielding accuracy scores of 0.9752 and 0.9569 respectively, leading us to choose these as the two models for hyperparameter tuning.

To get a good look into the key features that influence late deliveries, we then created a feature performance chart for our best two models shown in appendix O. These two visualizations give us insights into the ranking of the top 10 influential variables in predicting our target. We noticed that features such as days_for_shipment_scheduled, shipping_day_of_week, and order_day_of_week all performed impressively. Continuing, we will keep these variables in mind for further analysis.

Random Forest Accuracy	0.9568645554427108

XGBoost Accuracy	0.975736124365248

After creating and identifying our two best models, we used hyperparameter tuning to find the best parameters to improve our models accuracy. After finding this, the accuracy of our Random Forest model stayed almost exactly the same, although our XGBoost model went up a small amount to 0.9757. Then we created charts for the feature importances of each of these models again to see the results after tuning. The results were very similar, although on the Random Forest model, the variable days_for_shipment_scheduled increased giving us more confidence for predicting late delivery with it.



From our final models, we created a graph to compare the AUC/ROC curves between them both. This allowed us to evaluate their ability to distinguish between late and on time deliveries. In this visualization, the higher the AUC value the better the performance of the model, proving that XGBoost performed better.

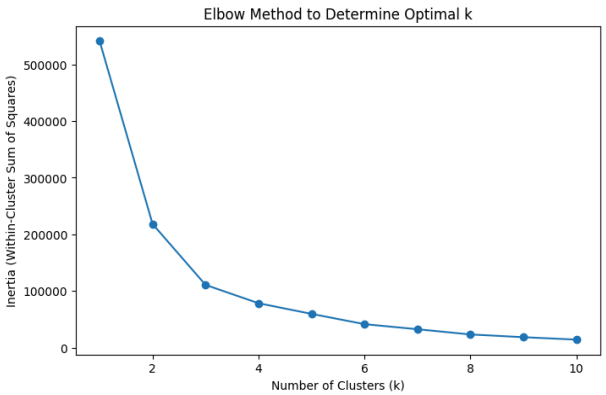
In order to interpret the behavior of our model further, SHAP (SHapley Additive exPlanations) to gain insights into the impact of individual features on the prediction shown in appendix P. In this visualization, SHAP values show how variables contributed to a prediction being classified as late or on time. The SHAP summary plot shows the overall impact of the variables. Then, the SHAP value plot for shipping_day_of_week explains how this variable

specifically changed predictions. Finally, the SHAP value plot for days_for_shipment_scheduled was included to do the same thing.

Cluster Analysis

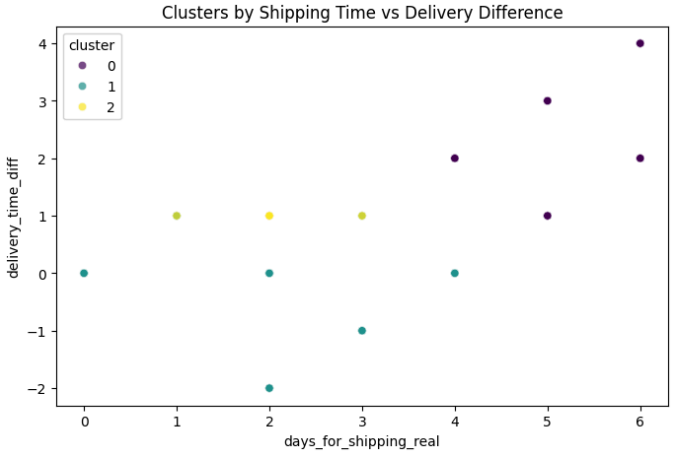
The Elbow graph below helps us determine the number of clusters (k) for k-means clustering. The elbow point in this graph is k = 3, which is the point where adding more clusters does not significantly reduce inertia. For our analysis we will thus use 3 clusters. This will minimize the variance within the clusters while keeping the model as simple as possible.

cluster	late_delivery_risk	days_for_shipping_real	delivery_time_diff
0	0.975878	5.349940	2.012539
1	0.000000	2.722135	-0.773529
2	1.000000	2.061098	1.000000



The Cluster Summary Table above provides a breakdown of late_delivery_risk, days_for_shipping_real, and delivery_time_diff for each cluster. Stores have been grouped together based on their shipping behavior to represent a cluster. Cluster 0 has a high late delivery risk (0.97), a longer shipping duration (5.35 days), and a positive delivery time difference (2.01 days), illustrating that these stores have a lot more late deliveries and take a lot longer to ship than the expected normal. Cluster 2 also has a high late delivery risk (1.00), but a shorter shipping duration (2.06 days) and a lower delivery time difference (1.00 day), indicating that these stores also experience late deliveries but in a shorter time frame. In contrast, cluster 1 has a late delivery risk of 0.00, meaning that orders in this group are delivered on time. The average shipping duration is 2.72 days, and the delivery

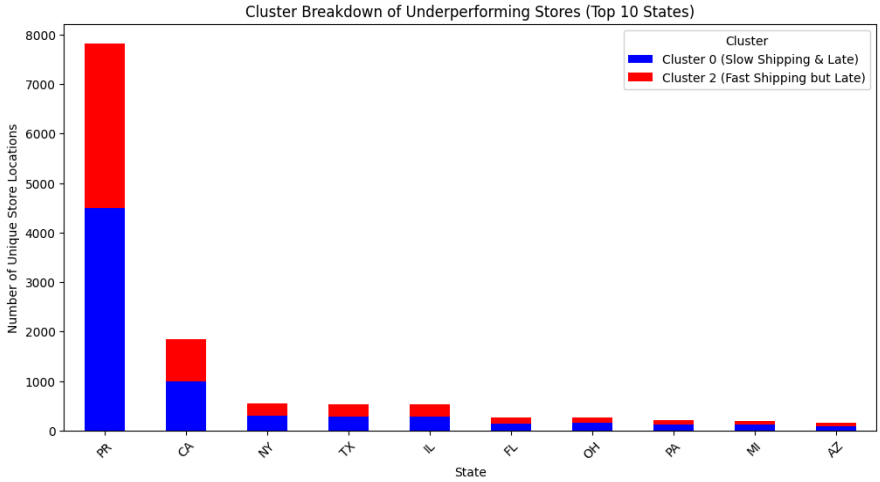
time difference is -0.77 days, showing that deliveries arrive ahead of schedule. These results show that cluster 1 represents the regions with efficient logistic operations while Cluster 0 and Cluster 2 are high-risk groups with frequent late deliveries. We should look into analyzing these clusters to understand and be able to replicate the success of cluster 1.



The scatterplot shows the relationship between actual shipping time and delivery time difference across the clusters. Delivery time difference was calculated by subtracting the scheduled shipping time from actual shipping time. Each point represents a clustered observation with different colors distinguishing between the clusters. From the graph we see that Cluster 0 has the longest shipping times, ranging from 4 to 6 days. The observations also are all positive for difference in delivery time between 1 to 4 days showing that the orders consistently are late. Cluster 2 has shorter delivery days between 1 and 3 while all being positive as well for difference in delivery time by around 1 day, showing minor delays. Cluster 1 in contrast seems to be the best performing group. The cluster has a low difference in delivery time between -2 and 0 days, meaning the deliveries are either on time or ahead of schedule. The cluster also has short shipping times mostly under 3 days. This illustrates that cluster 0 should be prioritized in improving operational efficiency, while cluster 2 can be slightly refined, and cluster 1 replicated where possible for most efficiency.

The bar chart visualizes how the clusters compare in number of unique store locations. As we can see cluster 1 has the most number of unique store locations of nearly 6000, followed by cluster 0 at around 4000 and cluster 2 at around 2000 locations. This could indicate that cluster 2 and 0 having a smaller amount of locations are a specialized group of locations that differ significantly from cluster 1. Cluster 1 having the most amount of locations can potentially be a benchmark for an average-performing group.

The Three boxplots in appendix Q illustrate how the clusters vary in distribution for delivery time difference, actual shipping days, and late delivery risk. For the delivery time difference boxplot, cluster 0 appears to be the most problematic with late deliveries mostly ranging from 1 to 3 days. Cluster 1 has high variability from -2 to 2 days, and cluster 2 has very stable slightly delayed deliveries of 1 day. For the late delivery risk graph, cluster 0 and 2 have near-consistent late delivery risk of 1 (100%), meaning that these clusters almost always arrive late. In contrast, cluster 1 has a late delivery risk of 0 (0%), meaning that deliveries are consistently on time. For the actual shipping by cluster graph, cluster 0 experiences the longest shipping delays between 5 to 6 days, contributing to its late deliveries. Cluster 1 again shows very high variability with shipping times ranging from 1 to 5 days. This could suggest that some stores are efficient while others are not. Cluster 2 has the lowest variability with shipping times



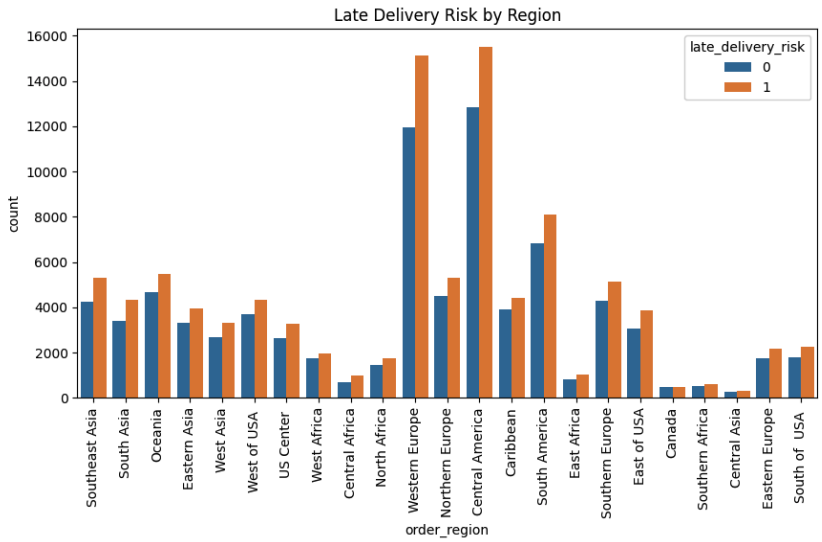
of around 2 days, and only a few outliers. This shows cluster 2 as having a more controlled and predictable shipping process.

To examine which states have the most underperforming stores and how they are distributed across clusters, we analyzed the top ten most underperforming states in the above graph. This analysis provides insights into regional logistical challenges, which can facilitate targeted operational improvements. Specifically, stores in Cluster 0 (Slow Shipping & Late) face issues that may include initial logistic processes such as warehousing and dispatch, whereas Cluster 2 (Fast Shipping but Late) highlights inefficiencies in last-mile delivery. Puerto Rico has an exponentially high count in both Cluster 0 and Cluster 2 with a total of almost 8000 stores, indicating severe inefficiencies and logistical problems. This high count is followed by California with a total of approximately 2000 stores.

Regional Analysis

Chi-Square Statistics	74.9336
Chi-Square p-value	0.00000011
Kruskal-Wallis H-Statistics	60.9680
Kruskal-Wallis p-value	0.00001605
T-Test Statistics	1.1016
T-Test p-value	0.2707

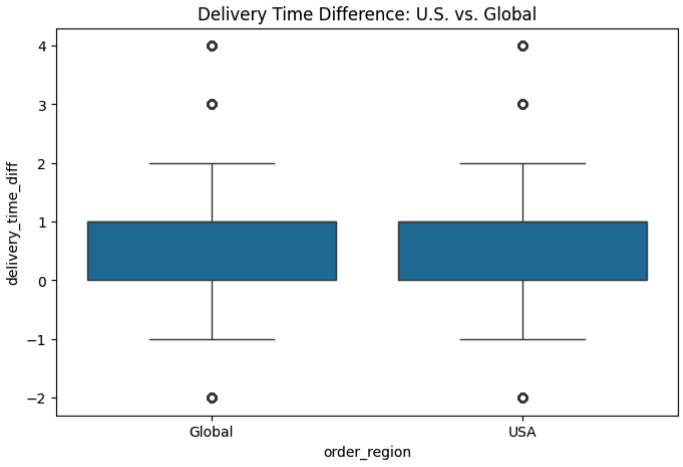
We want to determine if there is a statistically significant relationship between 'order_region', where the order is delivered, and 'Late_delivery_risk'. Since both are categorical variables, the Chi-Square test is performed. The result indicates an extremely low p-value of 0.00000011 indicating a strong significance where the likelihood of late delivery rates are, indeed, related to the region where packages are delivered. A relatively high Chi-Square value of 74.9336 also supports evidence against our null hypothesis.



In continuation of the Chi-Square test, we aim to identify which regions have the highest cases of 'Late_delivery_risk' by graphing the regions by count above. It was observed

that Western Europe and Central America have the highest risk of late deliveries. Although they also recorded the highest number of packages delivered on time, the discrepancy between on time (0) and late deliveries (1) is noticeable. On the other hand, the regions with the lowest rates of late deliveries include Canada, Southern Africa, and Central Asia with low counts of all deliveries. Further analysis is needed to understand why Western Europe and Central America experience the highest rates of late deliveries. We can also explore the possible reasons related to late deliveries such as infrastructure, economic factors, or climate conditions factors.

After confirming a relationship between 'order_region' and 'Late_delivery_risk', a Kruskal-Wallis test was performed to further explore the differences in the distribution of delivery time differences, 'delivery_time_diff' by 'order_region.' The test yielded a high H-statistic value of 60.9680 and an extremely low p-value of 0.00001605, indicating statistically significant differences in median delivery times across regions. The range and median are similar across all regions except for four. The boxplot visualization showed Southern Africa and Canada has the lowest median of 0 days although the range of delivery times varies significantly (from -2 to 4 days) which translates to inconsistent delivery rates. Additionally, Central Asia and South of the USA not only showed a wider range of delivery time difference but also had the highest median of one day. The findings of Kruskal-Wallis test show areas of different results than the initial 'Late Deliveries by Regions' graph.



The T-test comparing U.S. and global delivery performance yields a T-statistic of 1.1016 and p-value of 0.2707. The

p-value is significantly greater than the threshold value of 0.05 indicating there is no significant difference in the mean delivery times between U.S. and international shipments. The boxplot above supports the results of the T-test where U.S. and international delivery time differences showed similar distributions, suggesting that the variability of delivery times are comparable. Therefore, we can conclude that in terms of delivery time differences, the US and the rest of the world experiences are relatively similar.

Relationship between Store and Customer

The OLS multiple regression model (Appendix R) was used to find if shipping distances lead to a greater number of late deliveries. Ultimately, the model with distances resulted in an adjusted R^2 of 0.776, suggesting a strong relationship between independent variables and late delivery risk. However, without distance as a variable, the model still presented the same adjusted R^2 of 0.776. Through more testing, distance was also shown to have a p-value of .820 (Appendix S) within the regression analysis leading to the conclusion that within OLS multiple regression, distance is not a significant predictor. The F-statistic of 1.507e+04 is highly significant ($p<0.0001$), illustrating the overall model significantly predicts late delivery risk. Despite the strong adjusted R^2, the model shows signs of multicollinearity. This is illustrated by small eigenvalues (6.11e-20) and high p-values for the features. Thereby indicating that certain features are highly correlated with one another. However, after removing redundant features by excluding variables with a VIF over 10, the regression model(Appendix S) performed significantly worse with an adjusted R^2 of 0.214 and F-statistic of 2460. As such we decided to try another approach with stepwise regression (Appendix U).

Both the full and stepwise regression model share the same adjusted R-squared of 0.776, i.e., they explain the same amount of variance in late delivery risk. The stepwise model's F-statistic (6.243 e+04) is larger than the full model's F-statistic (1.457e+04), though both are very significant ($p<0.001$), indicating high predictive capability in both. The complete model has a lot of predictors that are not necessarily adding useful information. The stepwise model, on the other hand, has the same explanatory power using fewer variables, which minimizes the possibility of overfitting and simplifies the model to interpret. Other measures such as AIC and BIC are nearly identical for the two models, indicating that neither model is significantly better from an information-criterion perspective. Based on this, stepwise is greatly favored as it combines predictability at its best with simplicity. Its elegance and good performance in general mean that it is the more realistic model to make predictions about late delivery.

After utilizing multiple regression for predicting late deliveries by distance, an inquiry was raised to see if distance combined with other factors could be more accurate for predicting late deliveries within a neural network. Via SciKit Learn, an artificial neural network (Appendix T) was created with four hidden layers. Layers and corresponding neurons were decided upon after initial trial and error in accuracy rates. Layer #1 having 17 neurons, layer #2 having 10, layer #3 having 10, and finally layer #4 having 5.

Including distance as a variable, the model presented an 98.22% accuracy via SciKit Learns accuracy function with an R2 score of 0.77. Subtracting distance, the model improved minorly to 98.96% accuracy with the same R2 score of 0.77. In conclusion, distance was not found to aid in prediction of late deliveries within multiple machine learning modalities.

Ethical Recommendations

E-commerce has been involved in the everyday lives of the consumers for the past decade and are becoming progressively more active. The companies have also expanded their services to meet the high demands of consumers and grab a share of the market in this steep competition such as increasing shipping rate from multiple days to one-day and same day services. The e-commerce sector is evolving rapidly in the era of big data and new technologies. It is important to address the ethical concerns in this sector. Consumers are concerned about their private information such as how they are managed, collected, stored, and transparency. The environmental impact of e-commerce cannot be overlooked as predictions and understanding late delivery rates can significantly reduce this impact. Companies also have to keep up with the changes and adjustments in the regulatory policies in order to meet the updated requirements. Ethical considerations such as data privacy, environmental responsibility, and regulatory compliance will gain consumer trust and improve reputation.

Data Privacy

Consumers are concerned about the increasing volume of data collection. In our dataset, critical information about the consumers' names, emails, addresses, and passwords was collected. This sensitive information should only be collected with full explicit consent from the customers before the information collection. Once collected, consumers are also free to withdraw their consent at their will. Companies should be transparent about how the data will be used and shared. The privacy policies should be clear, concise and user friendly. This includes avoiding the use of small prints and technical jargon that might obscure the term meanings. However, the personal information of email address, password and names are not relevant to our analysis, such

data can be minimized or omitted in the dataset for our specific purpose. They were masked in our case to prevent identification.

Data minimization can prevent possible data breach and misuse. Improper anonymization increases the risk of cyberattacks and identity thefts due to the possibility of information exposure. Possibility of unethical data use may occur concerning selling consumer data to others without consent and excessive consumer profiling for misleading advertisements. Adhering to these principles will protect the consumers' private information which will increase the trust between the consumers and the company. However, e-commerce needs to strike a delicate balance between protecting consumer privacy and facilitating data-driven innovation since data is necessary to drive business insights, improving customer experience, and gaining competitive advantage (Nalla & Reddy, 2023)

Environmental Responsibility

E-commerce companies are facing increasing pressure to adopt environmentally friendly actions and policies for a greener earth. Carbon footprints increase significantly with fast shipping. Fast shipping produces significantly higher carbon dioxide emission by 15% since it's challenging to consolidate a full cargo and increase cost up to 68% (Muñoz-Villamizar et al., 2021). Various strategies have been rolled out from optimizing delivery routes and offering incentives for customers to reducing packaging waste. E-commerce platforms have given consumers an extra 1% cash back for choosing a single delivery date for multiple orders to decrease delivery frequency. Small orders are consolidated into a single large package to reduce cardboard use and minimizing route to reduce carbon prints.

Route optimization is another critical strategy to implement in order to reduce carbon footprint. If our model was successful, efficient routes reduce fuel consumption by reducing

travel distance and improve delivery efficiency. In return, the operation cost will also decrease. Improvement of route optimization using machine learning algorithms also indirectly improves customer experience by ensuring packages arrive predictably and on-time. Things will go in the opposite direction if our model had failed. A failed model can have a great negative impact for both the company and consumers. Poor predictions may cause warehouse congestions where inventory stocks may be mismanaged creating supply chain disruptions. Some deliveries may be over-prioritized while the prioritized ones may be delayed.

Integrating this research in the e-commerce sector increases a better ethical practice and simultaneously improves operational methods. The primary concerns relating to e-commerce are waste of excessive packaging and carbon emissions from deliveries. By analyzing historical data and using predictive modeling to increase efficiency of delivery routes alleviates such ethical problems. Decreasing late deliveries risks using this research also solidifies the credibility of the company in the market which in turn decreases operational cost. Beyond environmental responsibility, the company should be keen on protecting consumers' information. Under current regulation, there isn't a federal law to protect consumers privacy in the same way as the California Consumer Privacy Act. Most companies follow strict privacy protection for the consumers' benefit and to simplify the compliance aspects of regulations.

Challenges

During the timeline of the project numerous challenges occurred, most notably were in regards to interpreting the data set, handling uncommon variables, and the coding methods used for the analysis of the relationship between store and customer. During the analysis of the relationship between store and customer the team tried to reduce the number of factors in regression. Initial multiple regression model attempts were plagued by multicollinearity as indicated by the low eigenvalues and high VIF values, which indicated high intercorrelations between predictors. Multiple tests were done to reduce the model to only the most significant factors however the R2 score fell by at least 10%. The final stepwise regression procedure helped to solve the above problem but the challenge was still in retaining high predictive ability in the reduced model. In addition, it was theorized that logistic regression could be a better fit for the binary nature of our target y-value; however, R2 scores were still much lower than the initial regression model even with rigorous testing. Transitioning into the neural network, the team desired to switch the foundation of the model from SciKit Learn to the Keras library. Having more customizability could allow for a better fit to be found however it fell outside the scope of the project.

One of the biggest challenges was understanding and interpreting the data set due to irregular variable nomenclature. Some variables had names that were misleading, which forced us to carry out more in depth investigations. For example, the variable ‘customer_state’ seemingly represents the state to which the customer ordered a product, but after reviewing the documentation, we discovered that it actually referred to the store's location. Also, some variables were originally in the Spanish language and required translation to understand. Next, our data saved precise store locations with latitude and longitude coordinates and did not contain

any type of store ID variable. This spatial data required a different and new approach than the team was formerly familiar with. Getting past these challenges required detailed exploration of the data, referencing from the description of the dataset, and performing manual validations to ensure the data was understood in the correct way. Overall, resolving these problems was crucial in ensuring that our research and insights were based on the correct interpretation of the data.

Recommendations

Improvements

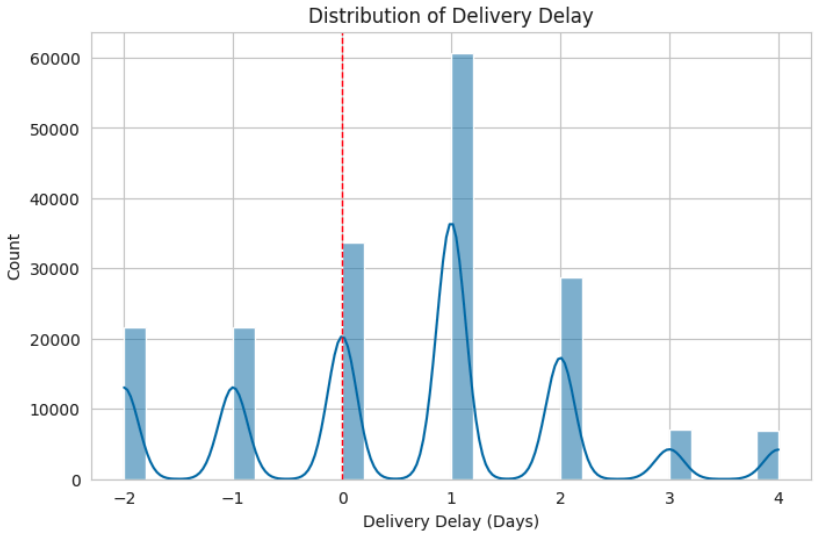
Looking back at our models and source code, several areas have the potential for improvement. Implementing cross-validation techniques, especially on the final models random forest and XGBoost, will generalize well on unseen data. While cross-validation can be computationally time-consuming and expensive for large models, we could choose a smaller fold (3 or 5) to improve the model without significantly increasing computational time. As for our clustering model, a silhouette score can be added to measure how distinct our clusters are from each other.

Next Step

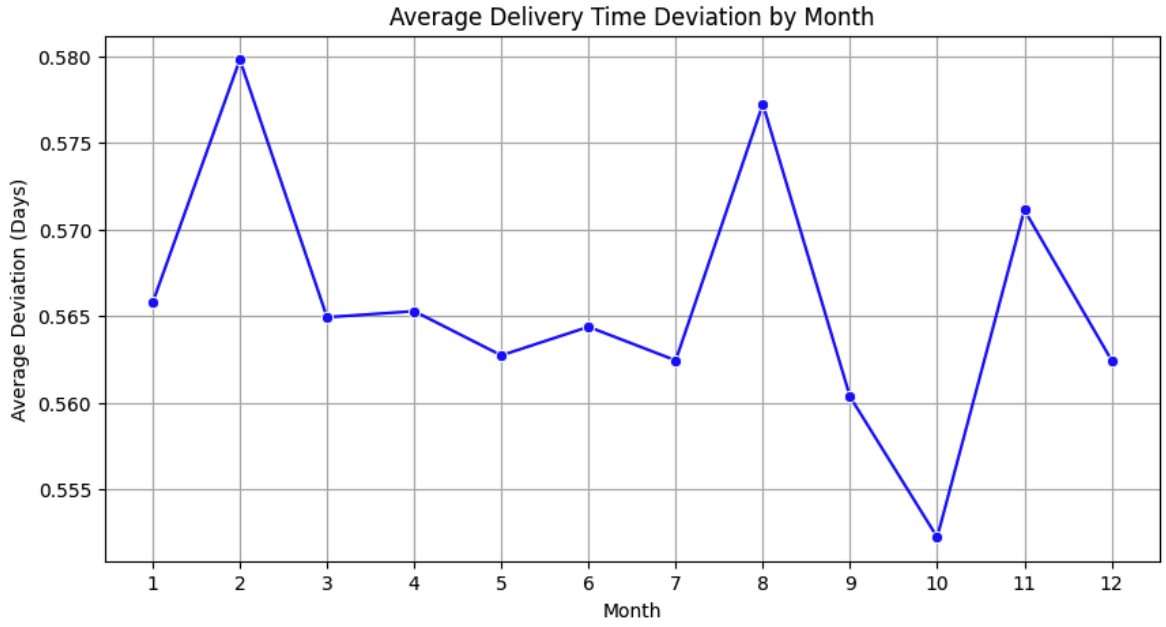
After implementing fine-tuning improvements, our next step is to prepare the model for deployment after it meets the performance criteria that is. The model will be used to make predictions on new and unseen data once deployed. The model will be continuously monitored to make sure it remains accurate. Regular maintenance will be performed as new data is gathered and emerged. We can also enhance the model's analysis by incorporating new data variables, such as delivery times, for further analysis and insights. These variables will be in the new data collection. This approach will ensure the model remains effective and continue to provide valuable predictions in the ever changing environment.

Appendix

A. Distribution of Delivery Delay counted for each order

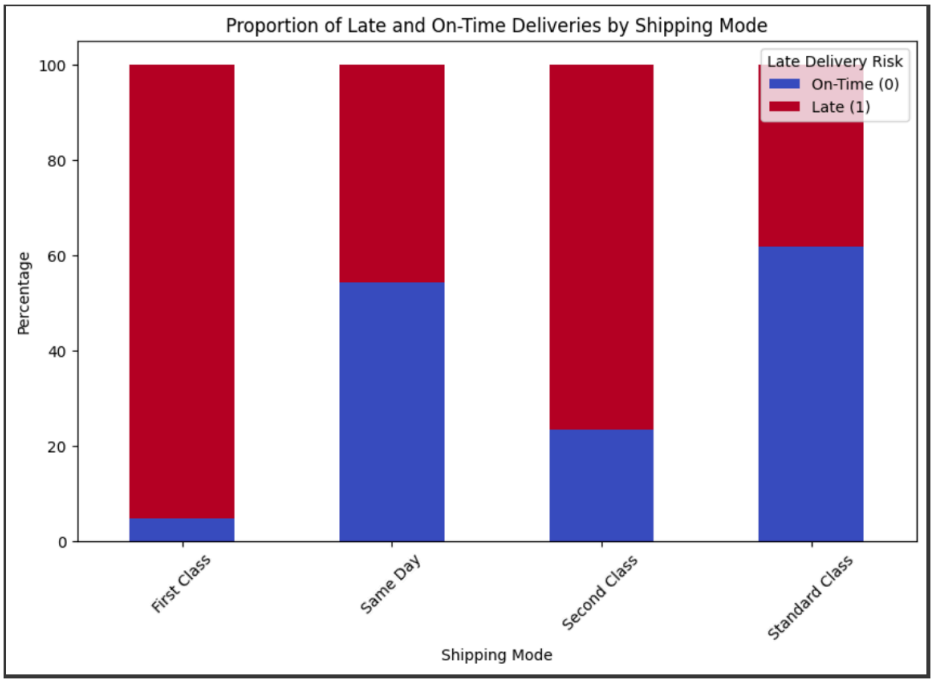


B. Average Delivery Time Deviation by Month

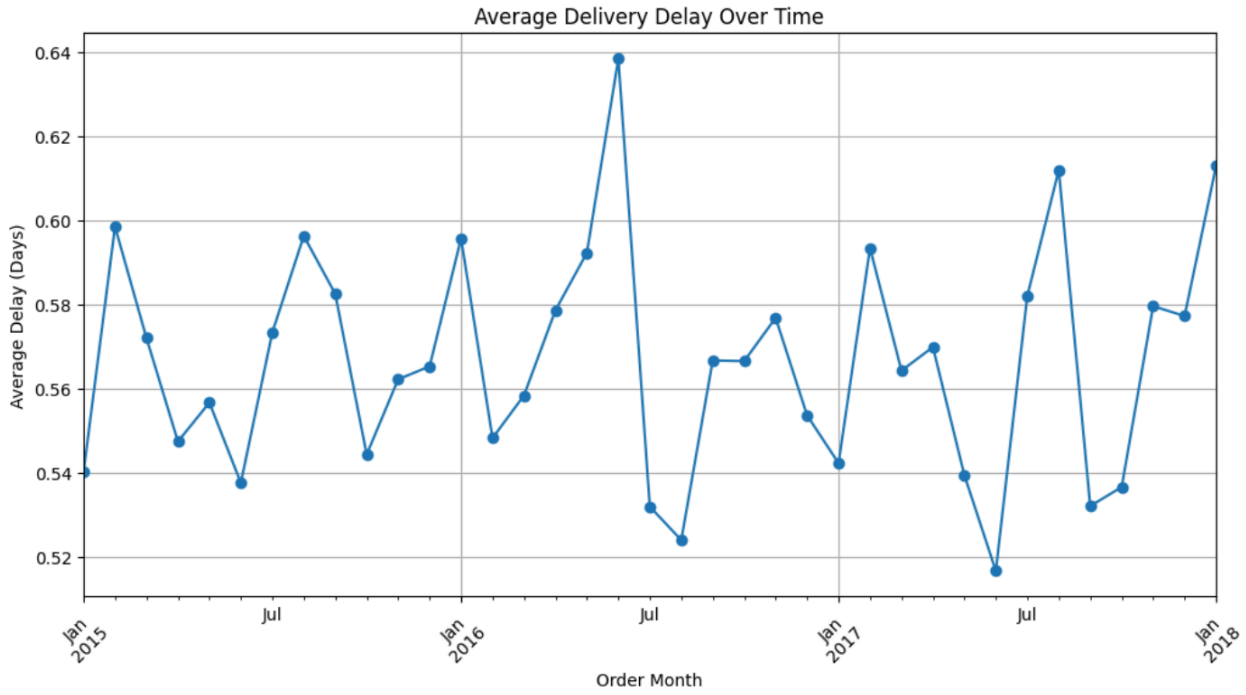


C. Late Delivery Risk by Shipping Mode

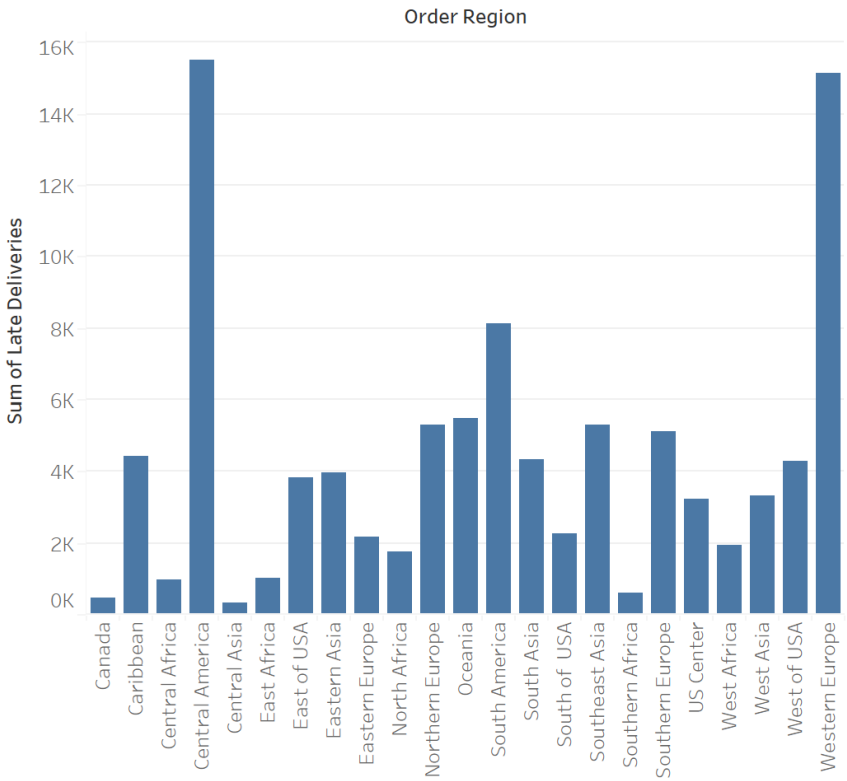
D. Proportion of Late and On-Time Deliveries by Shipping Mode



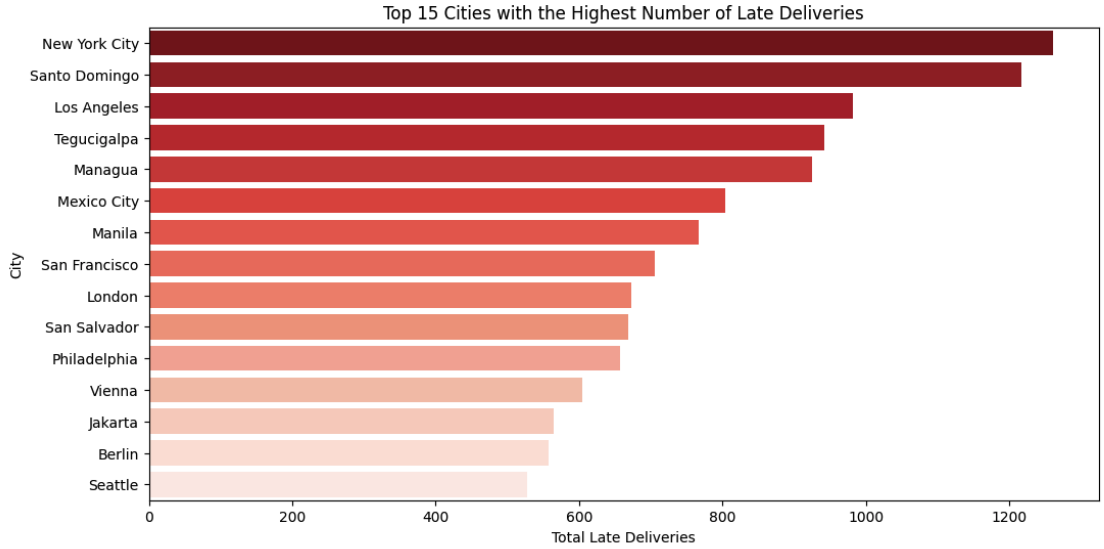
E.Average delivery Delay Over Time



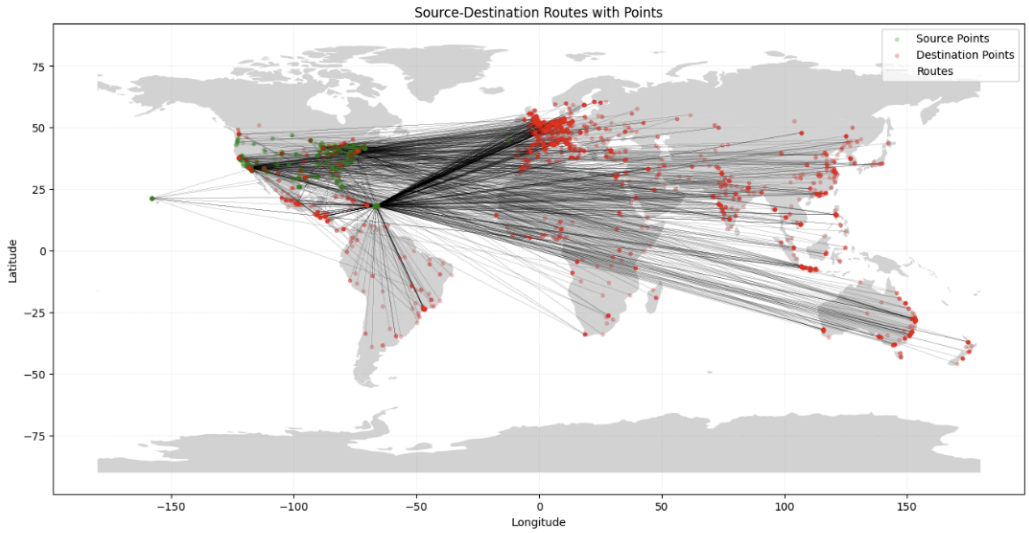
F. Late Deliveries by Order Region



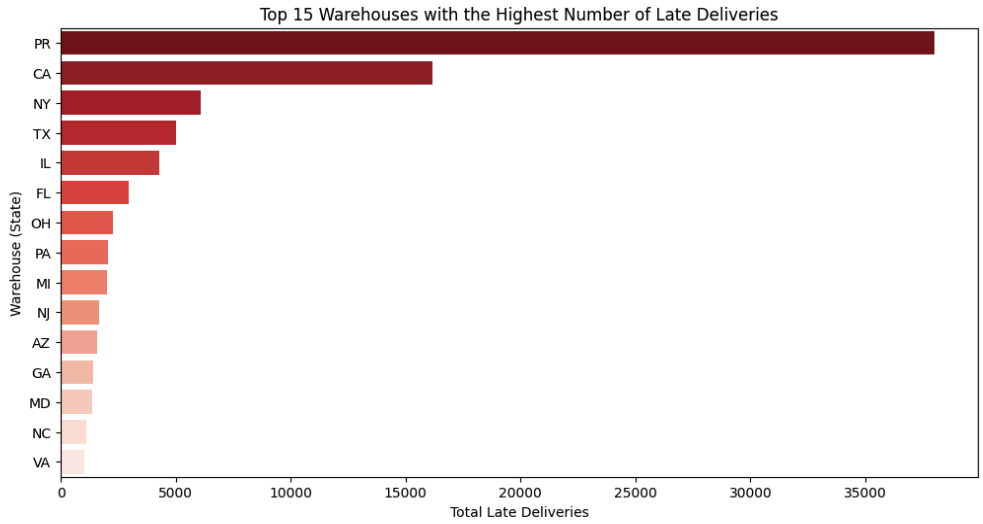
G. Top 15 Cities with Highest Late Deliveries



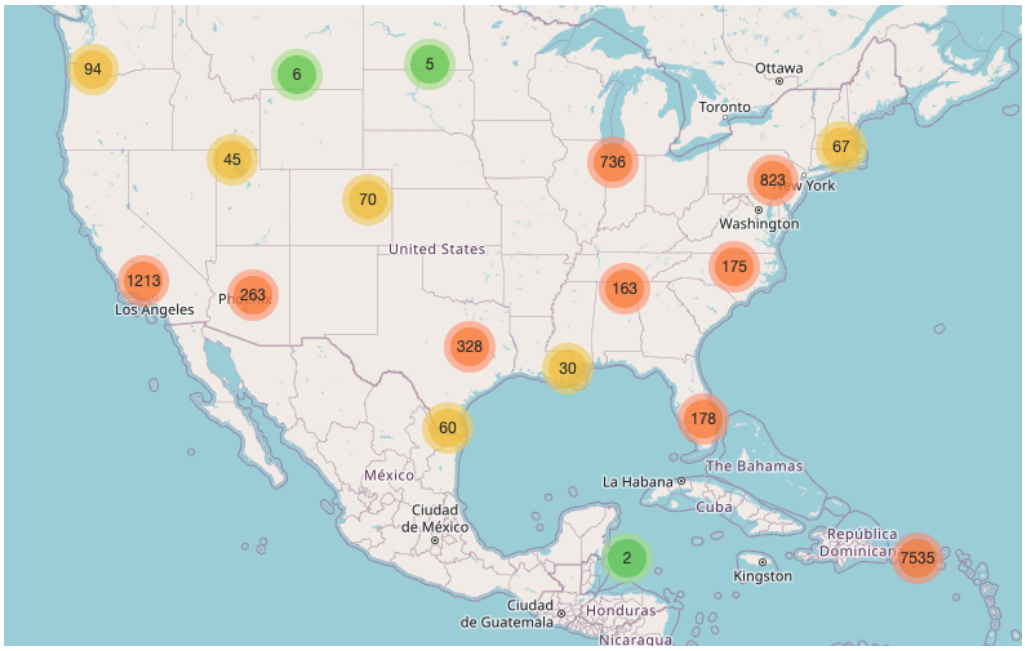
H. Routes from Store Location to Delivery Location



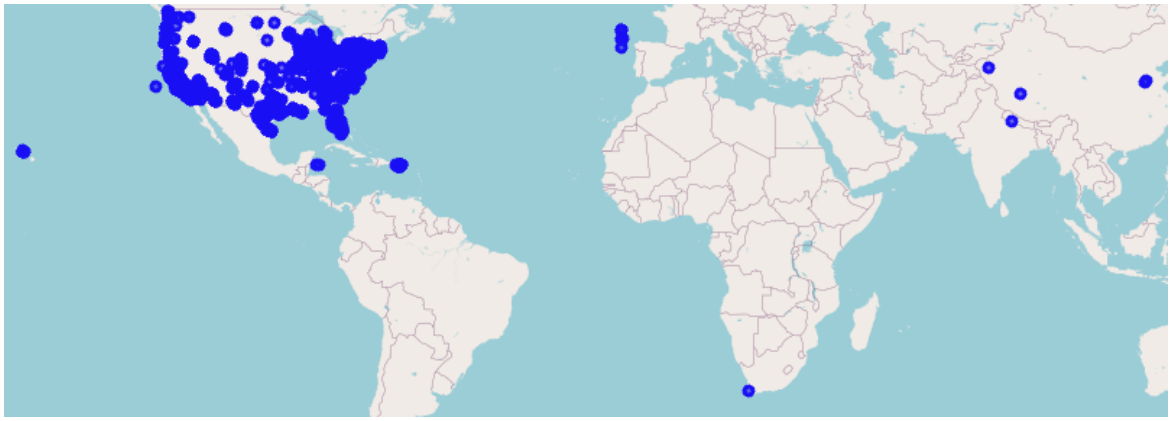
I. Top 15 Stores with Most Late Deliveries



J. Store Locations in US



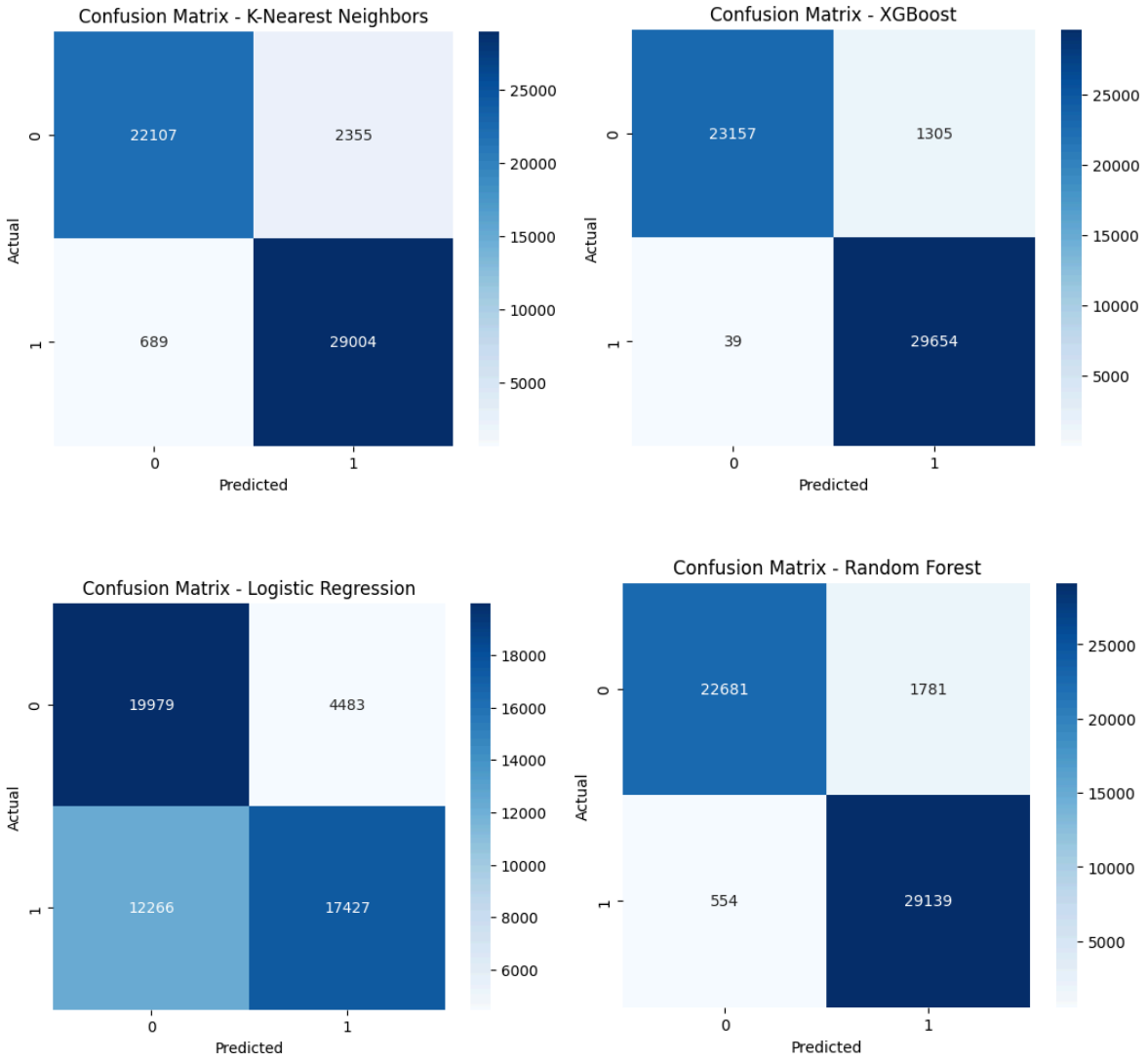
K. Store Locations Worldwide



L. Orders in US vs Global

US Orders	24840
Non-US orders	155679
Total Orders	180519
Percentage US Orders	13.76
Percentage Non-US Orders	86.24

M.


Confusion Matrix - K-Nearest Neighbors
22107
2355
689
29004
Actual
Predicted
Confusion Matrix - XGBoost
23157
1305
39
29654
Confusion Matrix - Logistic Regression
19979
4483
12266
17427
Confusion Matrix - Random Forest
22681
1781
554
29139

N.

K-Nearest Neighbors Model Performance:
Accuracy: 0.9438
Classification Report:

	precision	recall	f1-score	support
0	0.97	0.90	0.94	24462
1	0.92	0.98	0.95	29693
accuracy			0.94	54155
macro avg	0.95	0.94	0.94	54155
weighted avg	0.95	0.94	0.94	54155

XGBoost Model Performance:
Accuracy: 0.9752
Classification Report:

	precision	recall	f1-score	support
0	1.00	0.95	0.97	24462
1	0.96	1.00	0.98	29693
accuracy			0.98	54155
macro avg	0.98	0.97	0.97	54155
weighted avg	0.98	0.98	0.98	54155

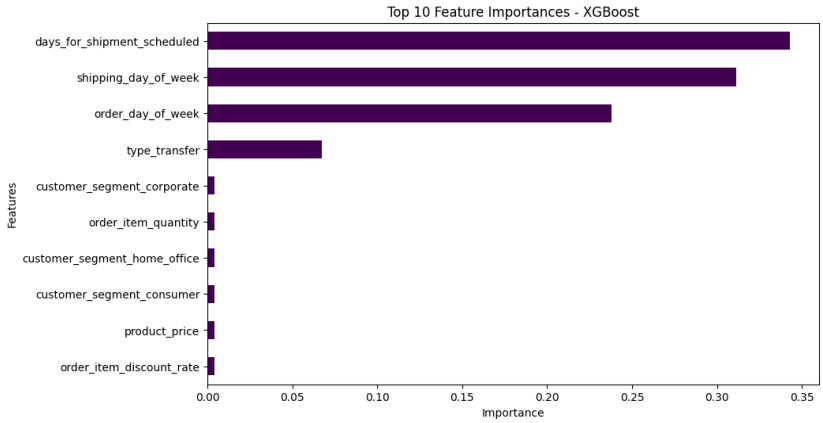
Logistic Regression Model Performance:
Accuracy: 0.6907
Classification Report:

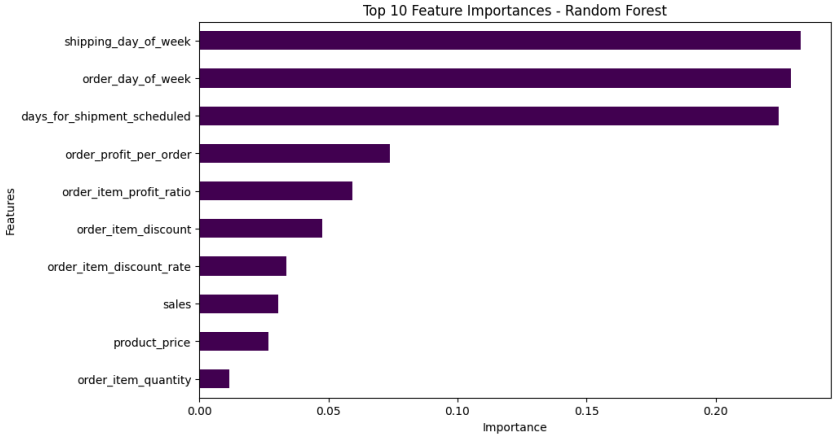
	precision	recall	f1-score	support
0	0.62	0.82	0.70	24462
1	0.80	0.59	0.68	29693
accuracy			0.69	54155
macro avg	0.71	0.70	0.69	54155
weighted avg	0.72	0.69	0.69	54155

Random Forest Model Performance:
Accuracy: 0.9569
Classification Report:

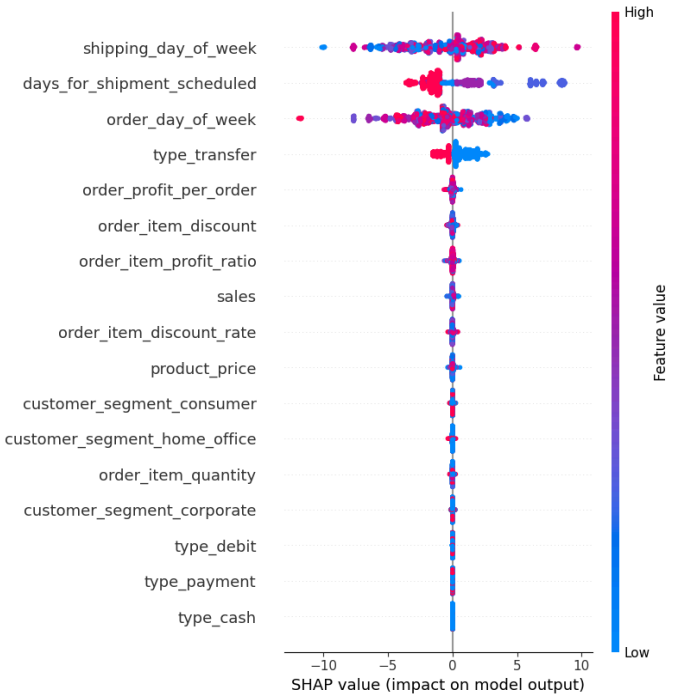
	precision	recall	f1-score	support
0	0.98	0.93	0.95	24462
1	0.94	0.98	0.96	29693
accuracy			0.96	54155
macro avg	0.96	0.95	0.96	54155
weighted avg	0.96	0.96	0.96	54155

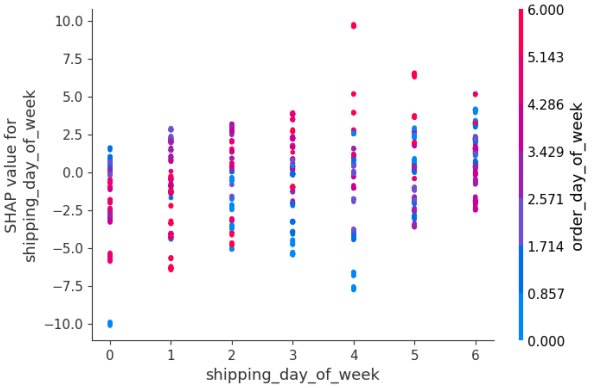
O.

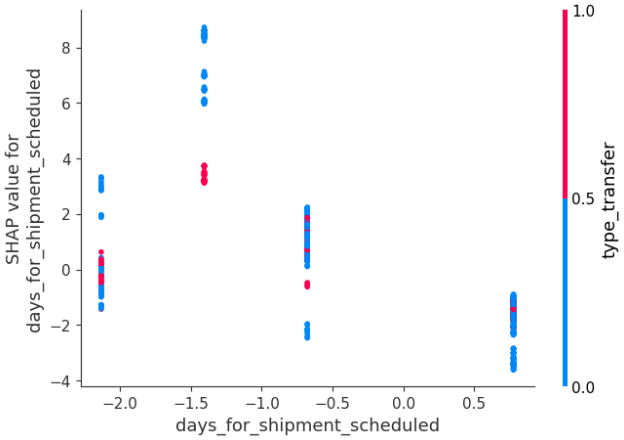

Top 10 Feature Importances - XGBoost
days_for_shipment_scheduled
shipping_day_of_week
order_day_of_week
type_transfer
customer_segment_corporate
order_item_quantity
customer_segment_home_office
customer_segment_consumer
product_price
order_item_discount_rate
Features
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Importance


Top 10 Feature Importances - Random Forest
shipping_day_of_week
order_day_of_week
days_for_shipment_scheduled
order_profit_per_order
order_item_profit_ratio
order_item_discount
order_item_discount_rate
sales
product_price
order_item_quantity
Features
0.00
0.05
0.10
0.15
0.20
Importance

P.

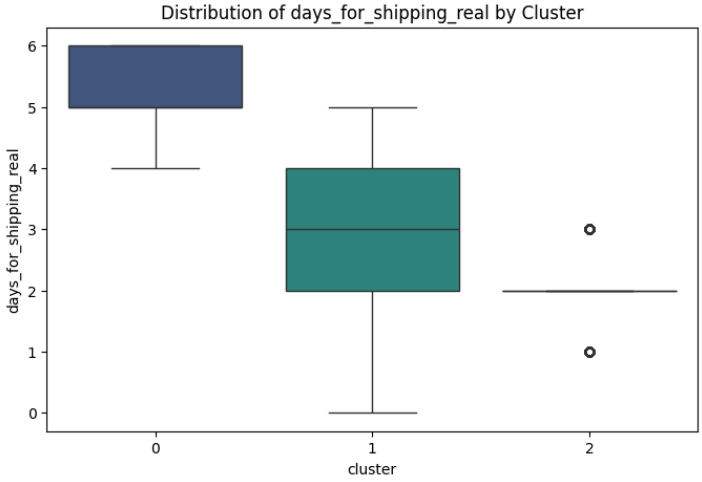

shipping_day_of_week
days_for_shipment_scheduled
order_day_of_week
type_transfer
order_profit_per_order
order_item_discount
order_item_profit_ratio
sales
order_item_discount_rate
product_price
customer_segment_consumer
customer_segment_home_office
order_item_quantity
customer_segment_corporate
type_debit
type_payment
type_cash
−10
−5
0
5
10
SHAP value (impact on model output)
High
Feature value
Low

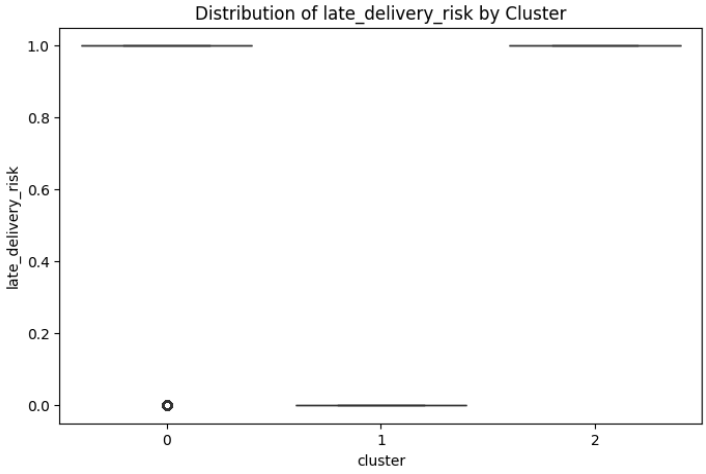

SHAP value for
shipping_day_of_week
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
0
1
2
3
4
5
6
shipping_day_of_week
6.000
5.143
4.286
3.429
2.571
1.714
0.857
0.000
order_day_of_week


SHAP value for
days_for_shipment_scheduled
8
6
4
2
0
−2
−4
−2.0
−1.5
−1.0
−0.5
0.0
0.5
days_for_shipment_scheduled
1.0
0.5
0.0
type_transfer

Q.


Distribution of delivery_time_diff by Cluster
delivery_time_diff
cluster
4
3
2
1
0
−1
−2
0
1
2


Distribution of days_for_shipping_real by Cluster
days_for_shipping_real
cluster
6
5
4
3
2
1
0
0
1
2


Distribution of late_delivery_risk by Cluster
late_delivery_risk
cluster
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2

R.

```
             Feature       VIF
0               type  3.574924
1    delivery_status  2.347569
2      category_name  4.484860
3      customer_city  3.309192
4   customer_country  3.216703
5     customer_state  6.903893
6    department_name  2.751253
7             market  4.627192
8      shipping_mode  4.578071
9   order_country_en  3.953590
10    order_state_en  4.539914
11     order_city_en  4.342407
12  customer_segment  1.715871
13         distance  3.438118
```

OLS Regression Results

Dep. Variable:	late_delivery_risk	R-squared:	0.214
Model:	OLS	Adj. R-squared:	0.214
Method:	Least Squares	F-statistic:	2460.
Date:	Mon, 17 Feb 2025	Prob (F-statistic):	0.00
Time:	03:23:12	Log-Likelihood:	-75939.
No. Observations:	126363	AIC:	1.519e+05
Df Residuals:	126348	BIC:	1.521e+05
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.1288	0.008	142.247	0.000	1.113	1.144
type	-0.0260	0.001	-20.852	0.000	-0.028	-0.024
delivery_status	-0.1137	0.001	-89.528	0.000	-0.116	-0.111
category_name	-0.0001	8.95e-05	-1.587	0.113	-0.000	3.34e-05
customer_city	8.727e-06	9.58e-06	0.911	0.362	-1.01e-05	2.75e-05
customer_country	0.0012	0.004	0.322	0.748	-0.006	0.008
customer_state	4.676e-05	0.000	0.392	0.695	-0.000	0.000
department_name	0.0004	0.001	0.877	0.380	-0.001	0.001
market	-0.0011	0.001	-0.937	0.349	-0.003	0.001
shipping_mode	-0.1893	0.001	-167.379	0.000	-0.192	-0.187
order_country_en	2.992e-05	2.73e-05	1.097	0.273	-2.35e-05	8.34e-05
order_state_en	1.135e-05	4.3e-06	2.637	0.008	2.91e-06	1.98e-05
order_city_en	2.653e-06	1.29e-06	2.055	0.040	1.22e-07	5.18e-06
customer_segment	0.0006	0.002	0.357	0.721	-0.003	0.004
distance	6.546e-07	2.9e-07	2.254	0.024	8.54e-08	1.22e-06

Omnibus:	768320.983	Durbin-Watson:	2.001
Prob(Omnibus):	0.000	Jarque-Bera (JB):	11794.742
Skew:	0.029	Prob(JB):	0.00
Kurtosis:	1.504	Cond. No.	5.60e+04

S.

OLS Regression Results

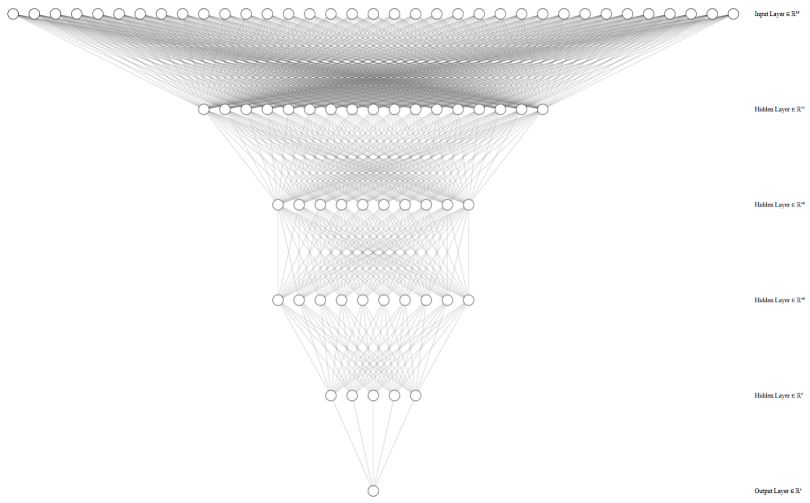
Dep. Variable:	late_delivery_risk	R-squared:	0.776
Model:	OLS	Adj. R-squared:	0.776
Method:	Least Squares	F-statistic:	1.457e+04
Date:	Sun, 16 Feb 2025	Prob (F-statistic):	0.00
Time:	19:53:35	Log-Likelihood:	3295.0
No. Observations:	126363	AIC:	-6528.
Df Residuals:	126332	BIC:	-6226.
Df Model:	30		
Covariance Type:	nonrobust		

```
==============================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                          0.7743      0.011     72.819      0.000       0.753       0.795
type                          -0.0251      0.001    -37.540      0.000      -0.026      -0.024
days_for_shipping_real         0.2714      0.000    560.880      0.000       0.270       0.272
days_for_shipment_scheduled   -0.1623      0.001   -126.787      0.000      -0.165      -0.160
benefit_per_order           4.794e-06   5.73e-06      0.836      0.403   -6.44e-06    1.6e-05
sales_per_customer             0.4962      0.220      2.259      0.024       0.066       0.927
delivery_status               -0.1666      0.001   -236.081      0.000      -0.168      -0.165
category_name                 -0.0001   5.27e-05     -2.714      0.007      -0.000   -3.97e-05
customer_city               1.058e-05   5.14e-06      2.060      0.039    5.11e-07    2.06e-05
customer_country               0.0049      0.004      1.172      0.241      -0.003       0.013
customer_segment              -0.0014      0.001     -1.615      0.106      -0.003       0.000
customer_state              5.457e-05   7.29e-05      0.748      0.454   -8.84e-05       0.000
department_id                  0.0009      0.001      1.818      0.069   -7.31e-05       0.002
department_name            -8.826e-05      0.000     -0.301      0.763      -0.001       0.000
latitude_src                   0.0003      0.000      1.619      0.105   -5.86e-05       0.001
longitude_src              -1.09e-05   4.94e-05     -0.220      0.826      -0.000     8.6e-05
market                        -0.0005      0.001     -0.775      0.439      -0.002       0.001
order_customer_id             -1.8e-07   1.64e-07     -1.100      0.271   -5.01e-07    1.41e-07
order_item_discount            0.9924      0.439      2.259      0.024       0.131       1.853
order_item_discount_rate      -0.0022      0.017     -0.127      0.899      -0.036       0.032
order_item_product_price    8.617e-07    1.1e-05      0.079      0.937   -2.06e-05    2.24e-05
order_item_profit_ratio       -0.0024      0.003     -0.935      0.350      -0.007       0.003
order_item_quantity           -0.0001      0.001     -0.086      0.932      -0.003       0.002
sales                         -0.9925      0.439     -2.259      0.024      -1.853      -0.131
order_item_total               0.4962      0.220      2.259      0.024       0.066       0.927
order_profit_per_order      4.794e-06   5.73e-06      0.836      0.403   -6.44e-06    1.6e-05
product_price               8.617e-07    1.1e-05      0.079      0.937   -2.06e-05    2.24e-05
product_status              3.456e-14   1.53e-14      2.259      0.024    4.57e-15    6.46e-14
shipping_mode                 -0.2136      0.002   -136.140      0.000      -0.217      -0.211
order_country_en            2.053e-05   1.61e-05      1.275      0.202    -1.1e-05    5.21e-05

order_city_en               5.794e-08    6.9e-07      0.084      0.933   -1.29e-06    1.41e-06
latitude_dest              -1.623e-05   3.18e-05     -0.511      0.610   -7.85e-05    4.61e-05
longitude_dest              9.741e-06   2.06e-05      0.473      0.636   -3.06e-05    5.01e-05
distance                    8.303e-08   3.65e-07      0.228      0.820   -6.32e-07    7.98e-07
==============================================================================
Omnibus:                    14248.022   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            30780.184
Skew:                          -0.702   Prob(JB):                         0.00
Kurtosis:                       4.969   Cond. No.                     1.10e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.29e-19. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

T.



U.

```
Selected Features after Stepwise Regression:
['shipping_mode', 'days_for_shipping_real', 'delivery_status', 'days_for_shipment_scheduled', 'type', 'order_state_en', 'customer_city']

--- Final Model Summary (Train Set) ---
                            OLS Regression Results
==============================================================================
Dep. Variable:     late_delivery_risk   R-squared:                       0.776
Model:                            OLS   Adj. R-squared:                  0.776
Method:                 Least Squares   F-statistic:                 6.243e+04
Date:                Mon, 17 Feb 2025   Prob (F-statistic):               0.00
Time:                        03:35:19   Log-Likelihood:                 3279.1
No. Observations:              126363   AIC:                            -6542.
Df Residuals:                  126355   BIC:                            -6464.
Df Model:                           7
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                            0.7829      0.003    288.618      0.000       0.778       0.788
shipping_mode                   -0.2136      0.002   -136.151      0.000      -0.217      -0.210
days_for_shipping_real           0.2714      0.000    560.929      0.000       0.270       0.272
delivery_status                 -0.1666      0.001   -236.113      0.000      -0.168      -0.165
days_for_shipment_scheduled     -0.1623      0.001   -126.831      0.000      -0.165      -0.160
type                            -0.0251      0.001    -37.544      0.000      -0.026      -0.024
order_state_en                5.941e-06   2.21e-06      2.684      0.007     1.6e-06    1.03e-05
customer_city                 9.626e-06   4.12e-06      2.336      0.020    1.55e-06    1.77e-05
==============================================================================
Omnibus:                    14244.725   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            30753.598
Skew:                          -0.702   Prob(JB):                         0.00
Kurtosis:                       4.967   Cond. No.                     2.72e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.72e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

--- Test Set Performance ---
R² (Test): 0.7774
Adjusted R² (Test): 0.7773
```

Code

Exploratory Data Analysis

RQ1 Analysis

```
# -*- coding: utf-8 -*-
"""RQ1 Final Analysis.ipynb
Automatically generated by Colab.
Original file is located at
    https://colab.research.google.com/drive/1xPHq-AKWhcEAWfnrOOVmtMn3G-xqTVZr
"""
!pip install --upgrade xgboost  # Upgrade XGBoost
import xgboost
print(xgboost.__version__)  # Confirm the version

# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV, StratifiedKFold #
Import StratifiedKFold here
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder, StandardScaler # Import
StandardScaler here
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, roc_curve, auc,
confusion_matrix, roc_auc_score # Import roc_auc_score here
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import cross_val_score # Import cross_val_score here
from sklearn.metrics import ConfusionMatrixDisplay # Import ConfusionMatrixDisplay

# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Set the file path and load
file_path = '/content/drive/My Drive/Colab Notebooks/DAT 490
CAPSTONE/DataCoSupplyChainDatasetRefined.csv'
df = pd.read_csv(file_path)
irrelevant_columns = ["customer_email", "customer_password", "product_image"]
df = df.drop(columns=irrelevant_columns, errors='ignore')
df.customer_lname.fillna(value="", inplace=True)
df.dropna(subset=['customer_zipcode'], inplace=True)
df.dropna(axis=1, inplace=True)
date_columns = ["order_date_dateorders", "shipping_date_dateorders"]
for col in date_columns:
    df[col] = pd.to_datetime(df[col], errors='coerce')
df['order_day_of_week'] = df['order_date_dateorders'].dt.dayofweek
df['shipping_day_of_week']= df['shipping_date_dateorders'].dt.dayofweek
df['shipping_delay'] = df.days_for_shipping_real - df.days_for_shipment_scheduled
```

```
df.drop(columns=date_columns, inplace=True)
useful_cat_cols = ['type', 'customer_segment']
df = pd.get_dummies(df, columns=useful_cat_cols, dtype=int)

# Standardising the column names again after encoding
df.columns = df.columns.str.lower().str.strip().str.replace(' ',
'_').str.replace(r'[()]', '', regex=True)
df.filter(regex='*|'.join(useful_cat_cols)).head()
shipping_priority = ['Same Day', 'First Class', 'Second Class', 'Standard Class']
shipping_mapping = {mode: idx for idx, mode in enumerate(shipping_priority)}
df.shipping_mode = df.shipping_mode.map(shipping_mapping)
quantitative_columns = [
    'benefit_per_order', 'days_for_shipping_real', 'days_for_shipment_scheduled',
    'sales_per_customer', 'order_item_discount', 'order_item_product_price',
    'order_item_profit_ratio', 'order_item_quantity', 'sales',
    'order_item_total', 'order_profit_per_order', 'product_price'
]
scaler = StandardScaler()
df[quantitative_columns] = scaler.fit_transform(df[quantitative_columns])
plt.figure(figsize=(10, 6))
sns.histplot(df['shipping_delay'], bins=30, kde=True, color='blue')
plt.title("Distribution of Shipping Delays")
plt.xlabel("Shipping Delay (days)")
plt.ylabel("Frequency");
target = df['late_delivery_risk']
useless_num = [
    'category_id', 'customer_id', 'customer_zipcode', 'department_id',
    'order_id', 'order_item_cardprod_id', 'order_item_id', 'order_customer_id',
    'product_card_id', 'product_category_id', 'product_status',
    'latitude_src', 'longitude_src', 'latitude_dest', 'longitude_dest'
]
useless_cat = [
    'category_name', 'customer_city', 'customer_country', 'customer_fname',
    'customer_lname','customer_state', 'customer_street',
    'order_status', 'order_region', 'order_state', 'order_city', 'order_country',
    'product_name', 'department_name', 'market', 'delivery_status', 'address_dest',
    'order_country_en', 'order_state_en', 'order_city_en'
]
useless_features = useless_cat + useless_num
features = df.drop(columns=['late_delivery_risk']+useless_features, errors='ignore')
print(features.shape)
correlation_matrix = features.corr()
plt.figure(figsize=(20, 12))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f',
linewidths=0.5)
plt.title("Correlation Matrix")
plt.show()
features.drop(columns=[
    'shipping_delay',
    'days_for_shipping_real',
    'shipping_mode',
    'benefit_per_order',
    'sales_per_customer',
    'order_item_total',
    'order_item_product_price',
```

```
], inplace=True, errors='ignore')
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.3,
random_state=42, stratify=target)
target_distribution = y_train.value_counts()
plt.figure(figsize=(8, 6))
sns.barplot(x=target_distribution.index, y=target_distribution.values,
palette='viridis')
plt.title('Target Variable Distribution (Class Imbalance Check)')
plt.xlabel('Class')
plt.ylabel('Frequency')
plt.show()
print("Target Variable Class Distribution:")
print(target_distribution)
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Define models
models = {
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42),
    "Logistic Regression": LogisticRegression(max_iter=1000, random_state=42),
    "XGBoost": XGBClassifier(use_label_encoder=False, eval_metric='logloss',
random_state=42),
    "K-Nearest Neighbors": KNeighborsClassifier(n_neighbors=5)
}

# Train and evaluate each model
results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    results[name] = accuracy
    print(f"\n{name} Model Performance:")
    print(f"Accuracy: {accuracy:.4f}")
    print("Classification Report:\n", classification_report(y_test, y_pred))
    # Plot Confusion Matrix
    import seaborn as sns
    import matplotlib.pyplot as plt
    plt.figure(figsize=(6, 5))
    sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
    plt.title(f"Confusion Matrix - {name}")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.show()

# Feature Importance for Random Forest and XGBoost
for name, model in models.items():
    if name in ["Random Forest", "XGBoost"]:
        feature_importances = pd.Series(model.feature_importances_,
index=X_train.columns)
        feature_importances.nlargest(10).sort_values().plot(kind='barh', figsize=(10,
6), colormap='viridis')
```

```
        plt.title(f"Top 10 Feature Importances - {name}")
        plt.xlabel("Importance")
        plt.ylabel("Features")
        plt.show()

"""THIS IS THE POINT WHERE WE TUNE / REFER TO NOTES FOR HYPERPARAMETER CODE THAT WAS
DELETED"""
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
import matplotlib.pyplot as plt
import pandas as pd
# Use Best Parameters Directly (Skipping Tuning)
best_rf = RandomForestClassifier(
    n_estimators=100,
    min_samples_split=5,
    max_depth=None,
    random_state=42
)
best_rf.fit(X_train, y_train)
best_xgb = XGBClassifier(
    n_estimators=200,
    max_depth=6,
    learning_rate=0.1,
    eval_metric='logloss',
    use_label_encoder=False,
    random_state=42
)
best_xgb.fit(X_train, y_train)

# Print Model Accuracy
print("Random Forest Accuracy:", best_rf.score(X_test, y_test))
print("XGBoost Accuracy:", best_xgb.score(X_test, y_test))

# Feature Importance Analysis
for name, model in {"Random Forest": best_rf, "XGBoost": best_xgb}.items():
    feature_importances = pd.Series(model.feature_importances_, index=X_train.columns)
    feature_importances.nlargest(10).sort_values().plot(kind='barh', figsize=(10, 6),
colormap='viridis')
    plt.title(f"Top 10 Feature Importances - {name} (Final Model)")
    plt.xlabel("Importance")
    plt.ylabel("Features")
    plt.show()
from sklearn.metrics import roc_curve, auc

# Function to plot AUC/ROC curve
def plot_auc_roc(model, X_test, y_test, model_name):
    y_probs = model.predict_proba(X_test)[:, 1]  # Get probability scores for the
positive class
    fpr, tpr, _ = roc_curve(y_test, y_probs)
    auc_score = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f"{model_name} (AUC = {auc_score:.4f})")

# Plot ROC curves for both tuned models
plt.figure(figsize=(10, 6))
plot_auc_roc(best_rf, X_test, y_test, "Random Forest (Tuned)")
```

```
plot_auc_roc(best_xgb, X_test, y_test, "XGBoost (Tuned)")
# Plot the reference "no skill" classifier
plt.plot([0, 1], [0, 1], linestyle='--', color='gray', label="No Skill (AUC = 0.5)")
plt.xlabel("False Positive Rate (FPR)")
plt.ylabel("True Positive Rate (TPR)")
plt.title("AUC/ROC Curve for Tuned Models")
plt.legend()
plt.show()
import shap
explainer = shap.TreeExplainer(best_xgb)
shap_values = explainer.shap_values(X_test[:2000], approximate=True)  # Limit to 2,000
rows
shap.summary_plot(shap_values, X_test[:2000])  # Plot only for limited data

# Ensure X_test is limited to the same size as shap_values
X_test_sample = X_test[:2000]
shap.dependence_plot("days_for_shipment_scheduled", shap_values, X_test_sample)
shap.dependence_plot("shipping_day_of_week", shap_values, X_test_sample)
```

RQ2 Clustering

```
# -*- coding: utf-8 -*-
"""RQ2 Final Analysis - Clustering.ipynb
Automatically generated by Colab.
Original file is located at
    https://colab.research.google.com/drive/1c8XEPAMWn9m7u8i5bdowOEqWJ7866oH4
"""
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from google.colab import drive
drive.mount('/content/drive')

# Set the file path and load
file_path = '/content/drive/My Drive/Colab Notebooks/DAT 490
CAPSTONE/DataCoSupplyChainDatasetRefined.csv'
df = pd.read_csv(file_path)

# Ensure relevant columns are in the right format
df['late_delivery_risk'] = df['late_delivery_risk'].astype(int)
df['delivery_time_diff'] = df['days_for_shipping_real'] -
df['days_for_shipment_scheduled']

# Select relevant columns for store performance (adjust based on data)
store_features = ['late_delivery_risk', 'days_for_shipping_real',
'delivery_time_diff']

# Handle missing values by filling with median values
df[store_features] = df[store_features].fillna(df[store_features].median())

# Standardize the data
```

```
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df[store_features])

# Determine the optimal number of clusters using the Elbow Method
inertia = []
K_range = range(1, 11)  # Try different cluster counts
for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(df_scaled)
    inertia.append(kmeans.inertia_)

# Plot the Elbow Method graph
plt.figure(figsize=(8,5))
plt.plot(K_range, inertia, marker='o', linestyle='-')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia (Within-Cluster Sum of Squares)')
plt.title('Elbow Method to Determine Optimal k')
plt.show()

# Choose the best k (based on the Elbow Method) and apply K-Means clustering
optimal_k = 3  # Adjust based on the elbow method result
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
df['cluster'] = kmeans.fit_predict(df_scaled)

# Analyze the clusters
cluster_means = df.groupby('cluster')[store_features].mean()
print(cluster_means)

# Sample a subset of the data for visualization
df_sample = df.sample(n=5000, random_state=42)

# Generate a pairplot with sampled data
sns.pairplot(df_sample, hue='cluster', diag_kind='kde', palette='viridis')
plt.show()
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
sns.scatterplot(data=df_sample, x='days_for_shipping_real', y='delivery_time_diff',
hue='cluster', palette='viridis', alpha=0.7)
plt.title("Clusters by Shipping Time vs Delivery Difference")
plt.show()

# Count unique store locations using both latitude & longitude
total_unique_stores = df[['latitude_src', 'longitude_src']].drop_duplicates().shape[0]
print(f"Total number of unique stores: {total_unique_stores}")
import matplotlib.pyplot as plt

# Assign each store (lat/long) to its most frequent cluster
store_cluster_mapping = (
    df.groupby(['latitude_src', 'longitude_src'])['cluster']
    .agg(lambda x: x.value_counts().idxmax())  # Assign the most frequent cluster per
store
    .reset_index()
)
```

```
# Count unique stores per cluster
store_counts_by_cluster = store_cluster_mapping['cluster'].value_counts().sort_index()

# Plot the corrected cluster distribution
plt.figure(figsize=(8,5))
plt.bar(store_counts_by_cluster.index, store_counts_by_cluster.values, color='blue',
alpha=0.7)
plt.xlabel('Cluster')
plt.ylabel('Number of Unique Store Locations')
plt.title('Number of Unique Stores per Cluster (Using Lat & Long)')
plt.xticks(store_counts_by_cluster.index)
plt.show()

# Print the sum of unique stores across clusters to verify it matches total stores
print(f"Sum of stores across clusters: {store_counts_by_cluster.sum()} (Should match
{total_unique_stores})")
import seaborn as sns

# Plot boxplots for each feature by cluster
for feature in store_features:
    plt.figure(figsize=(8,5))
    sns.boxplot(x='cluster', y=feature, data=df, palette='viridis')
    plt.title(f'Distribution of {feature} by Cluster')
    plt.show()
import matplotlib.pyplot as plt

# Count unique store locations per state
underperforming_store_counts = (
    df[df['cluster'].isin([0, 2])]
    .groupby('customer_state')[['latitude_src', 'longitude_src']]
    .nunique()
    .sum(axis=1)  # Sum unique latitudes and longitudes as an approximation for unique
stores
    .sort_values(ascending=False)
)

# Filter to the **Top 10 states**
top_10_states = underperforming_store_counts.head(10)

# Plot the corrected graph
plt.figure(figsize=(10,5))
top_10_states.plot(kind='bar', color='red', alpha=0.7)
plt.title('Top 10 States with the Most Underperforming Stores (Based on Unique
Locations)')
plt.xlabel('State')
plt.ylabel('Number of Unique Store Locations')
plt.xticks(rotation=45)
plt.show()
import matplotlib.pyplot as plt

# Count unique store locations per state by cluster (using only latitude)
cluster_state_counts = (
    df[df['cluster'].isin([0, 2])]
    .groupby(['customer_state', 'cluster'])['latitude_src']
    .nunique()
```

```
    .unstack()  # Create separate columns for Cluster 0 and Cluster 2
    .fillna(0)  # Replace NaN values with 0
    .astype(int)  # Convert to integers
)

# Select the top 10 states with the most underperforming stores
top_10_states = cluster_state_counts.sum(axis=1).nlargest(10).index
filtered_cluster_counts = cluster_state_counts.loc[top_10_states]

# Plot the corrected stacked bar chart
filtered_cluster_counts.plot(kind='bar', stacked=True, figsize=(12,6), color=['blue',
'red'])
plt.title('Cluster Breakdown of Underperforming Stores (Top 10 States)')
plt.xlabel('State')
plt.ylabel('Number of Unique Store Locations')
plt.xticks(rotation=45)
plt.legend(title="Cluster", labels=["Cluster 0 (Slow Shipping & Late)", "Cluster 2
(Fast Shipping but Late)"])
plt.show()
```

RQ2 Chi Square and T test

```
# -*- coding: utf-8 -*-
"""RQ2 Final Analysis - Chi Square & T-test.ipynb
Automatically generated by Colab.
Original file is located at
    https://colab.research.google.com/drive/15nbK-RP6ZVF75u7305xYqeQGdlF0GvGs
"""
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import ttest_ind, chi2_contingency, kruskal
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from geopy.distance import geodesic
from google.colab import drive
drive.mount('/content/drive')
# Set the file path and load
file_path = '/content/drive/My Drive/Colab Notebooks/DAT 490
CAPSTONE/DataCoSupplyChainDatasetRefined.csv'

df = pd.read_csv(file_path)
print(df.head())
print(df.info())

# Ensure relevant columns are in the right format
df['late_delivery_risk'] = df['late_delivery_risk'].astype(int)
df['delivery_time_diff'] = df['days_for_shipping_real'] -
df['days_for_shipment_scheduled']

# Check unique regions
print(df['order_region'].value_counts())
```

```
### Chi-Square Test: Late Delivery Risk Across Regions ###
contingency_table = pd.crosstab(df['order_region'], df['late_delivery_risk'])
chi2, p, dof, expected = chi2_contingency(contingency_table)
print("\nChi-Square Test Results:")
print(f"Chi-Square Statistic: {chi2:.4f}, p-value: {p:.8f}")
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='order_region', hue='late_delivery_risk')
plt.xticks(rotation=45)
plt.title("Late Delivery Risk by Region")
plt.show()

### Comparing Delivery Time Differences Across Regions ###
# Kruskal-Wallis test (non-parametric alternative to ANOVA)
groups = [df[df['order_region'] == region]['delivery_time_diff'].dropna() for region
in df['order_region'].unique()]
h_stat, p_value = kruskal(*groups)
print("\nKruskal-Wallis Test Results:")
print(f"H-statistic: {h_stat:.4f}, p-value: {p_value:.8f}")

# Boxplot of delivery time differences by region
plt.figure(figsize=(12,6))
sns.boxplot(data=df, x='order_region', y='delivery_time_diff')
plt.xticks(rotation=45)
plt.title("Delivery Time Difference by Region")
plt.show()

# Define U.S. regions (Adjust as needed based on your dataset)
us_regions = ["West of USA", "US Center", "East of USA", "South of USA"]

# Create two groups: U.S. vs. Global
us_group = df[df['order_region'].isin(us_regions)]['delivery_time_diff'].dropna()
global_group = df[~df['order_region'].isin(us_regions)]['delivery_time_diff'].dropna()

# Perform an independent t-test (assuming unequal variances)
t_stat, p_value = ttest_ind(us_group, global_group, equal_var=False)
print("\nT-test: U.S. vs. Global Delivery Performance")
print(f"T-statistic: {t_stat:.4f}, p-value: {p_value:.4f}")

# Visualizing the comparison
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x=df['order_region'].apply(lambda x: "USA" if x in us_regions
else "Global"), y="delivery_time_diff")
plt.title("Delivery Time Difference: U.S. vs. Global")
plt.show()
```

RQ2 Relationship between Store and Customer

```
# -*- coding: utf-8 -*-
"""distanceNeuralNetworksRegression final.ipynb
Automatically generated by Colab.
Original file is located at
    https://colab.research.google.com/drive/1t7Ivc20aLJL6nsA32JoIBvnnjO1V6sti
"""
import pandas as pd
```

```
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from math import cos, asin, sqrt, pi

# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Set the file path and load
file_path = '/content/drive/My Drive/Colab Notebooks/DAT 490
CAPSTONE/DataCoSupplyChainDatasetRefined.csv'
df = pd.read_csv(file_path)
df.head()

#Stackoverflow used for basis of code, then improved/modified
#Haversine formula to calculate distance between two points on the earth (specified in
decimal degrees)
def distance(col):
    val = 0.5 - cos((col['latitude_dest']-col['latitude_src'])*(pi / 180))/2 +
cos(col['latitude_src']*(pi / 180)) * cos(col['latitude_dest']*(pi / 180)) *
(1-cos((col['longitude_dest']-col['longitude_src'])*(pi / 180)))/2
    return asin(sqrt(val)) * 12742
df['distance'] = df.apply(distance, axis=1)

# Encode categorical columns
labelencoder = LabelEncoder()
columns_to_encode = [
    'type', 'delivery_status', 'category_name', 'customer_city',
    'customer_country', 'customer_state', 'department_name', 'market',
    'order_region', 'shipping_mode', 'order_country_en', 'order_state_en',
    'order_city_en', 'customer_segment'
]
for column in columns_to_encode:
    df[column] = labelencoder.fit_transform(df[column])

# Drop irrelevant variables
df = df.drop(['category_id',
'customer_email','customer_fname','customer_id','customer_lname','customer_password','
customer_street','customer_zipcode','order_date_dateorders','order_id','order_item_car
dprod_id','order_item_id','order_status','order_zipcode','product_card_id','product_ca
tegory_id','product_image','product_name','shipping_date_dateorders','address_dest','o
rder_state','order_region','order_country','order_city'], axis=1)

# Prepare features and target variable for model training
X = df
X = X.drop('late_delivery_risk', axis=1)
X = X.drop('distance', axis=1)
y = df['late_delivery_risk']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.30,random_state=44)
```

```
# Configurations for hidden layers
hidden_layers_configs = [
    (17,10,10),
    (17,10,10, 5),
    (100, 50)
]

# Initialize and train the neural network classifier with different hidden layer sizes
for config in hidden_layers_configs:
    neural_network = MLPClassifier(max_iter=5000, activation="relu", solver='adam',
hidden_layer_sizes=config, random_state=1)
    neural_network.fit(X_train, y_train)
    predictions = neural_network.predict(X_test)
    # Calculate and print the model's accuracy
    accuracy = accuracy_score(y_test, predictions)
    print(f"Configuration {config} - Model Accuracy: {accuracy:.4f}")

#Accuracy WITH distance    96.53962626486447
#Accuracy WITHOUT distance 98.85331265233769
```

Stepwise Regression:

```
labelencoder = LabelEncoder()

categorical_features = [
    "type", "delivery_status", "category_name", "customer_city", "customer_country",
    "customer_state", "department_name", "market", "shipping_mode",
    "order_country_en", "order_state_en", "order_city_en", "customer_segment"
]

for col in categorical_features:
    if col in df.columns:
        df[col] = labelencoder.fit_transform(df[col])

# 3. Define Your Target and Potential Predictors
target = "late_delivery_risk"
predictors = [
    "type", "days_for_shipping_real", "days_for_shipment_scheduled", "benefit_per_order",
    "sales_per_customer", "shipping_mode", "distance", "delivery_status", "customer_segment",
    "latitude_src", "market", "order_item_discount", "order_item_product_price",
    "order_item_profit_ratio", "order_item_quantity", "order_item_total",
    "order_profit_per_order", "product_price", "order_country_en", "order_state_en",
    "order_city_en", "latitude_dest", "longitude_dest", "product_status", "category_name",
    "customer_city", "customer_country", "customer_state", "department_name"
]
predictors = [p for p in predictors if p in df.columns]

X_train, X_test, y_train, y_test = train_test_split(
    df[predictors], df[target], test_size=0.3, random_state=44
)

# 5. Stepwise Selection Function
def stepwise_selection(X, y,
                       initial_list=None,
                       threshold_in=0.05,
                       threshold_out=0.05,
                       verbose=True):
```

```
    if initial_list is None:
        included = []
    else:
        included = list(initial_list)

    while True:
        changed = False

        # Step 1: Forward Step
        excluded = list(set(X.columns) - set(included))
        new_pval = pd.Series(index=excluded, dtype=float)
        for new_col in excluded:
            model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included + [new_col]]))).fit()
            new_pval[new_col] = model.pvalues[new_col]
        best_pval = new_pval.min()
        if best_pval < threshold_in:
            best_feature = new_pval.idxmin()
            included.append(best_feature)
            changed = True
            if verbose:
                print(f"Add  {best_feature} p-value = {best_pval:.6f}")

        # Step 2: Backward Step
        model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included]))).fit()
        # Use all the pvalues except intercept
        pvalues = model.pvalues.iloc[1:]
        worst_pval = pvalues.max()
        if worst_pval > threshold_out:
            worst_feature = pvalues.idxmax()
            included.remove(worst_feature)
            changed = True
            if verbose:
                print(f"Drop {worst_feature} p-value = {worst_pval:.6f}")

        if not changed:
            break
```

```
        # Use all the pvalues except intercept
        pvalues = model.pvalues.iloc[1:]
        worst_pval = pvalues.max()
        if worst_pval > threshold_out:
            worst_feature = pvalues.idxmax()
            included.remove(worst_feature)
            changed = True
            if verbose:
                print(f"Drop {worst_feature} p-value = {worst_pval:.6f}")

        if not changed:
            break

    return included

selected_features = stepwise_selection(X_train, y_train, threshold_in=0.05, threshold_out=0.05, verbose=True)
print("\nSelected Features after Stepwise Regression:")
print(selected_features)

X_train_sm = sm.add_constant(X_train[selected_features])
X_test_sm = sm.add_constant(X_test[selected_features])
final_model = sm.OLS(y_train, X_train_sm).fit()
print("\n--- Final Model Summary (Train Set) ---")
print(final_model.summary())


y_pred = final_model.predict(X_test_sm)

from sklearn.metrics import r2_score
r2_test = r2_score(y_test, y_pred)
n_test = X_test.shape[0]
p_test = len(selected_features)
adjusted_r2_test = 1 - ((1 - r2_test) * (n_test - 1) / (n_test - p_test - 1))

print("\n--- Test Set Performance ---")
print(f"R² (Test): {r2_test:.4f}")
print(f"Adjusted R² (Test): {adjusted_r2_test:.4f}")
```

References

Dataset. Retrieved from: https://www.kaggle.com/datasets/shashwatwork/dataco-smart-supply-chain-for-big-data-analysis?select=DescriptionDataCoSupplyChain.csv

International Trade Administration. (2021). *eCommerce Sales & Size Forecast*. Www.trade.gov. https://www.trade.gov/ecommerce-sales-size-forecast

Matteo Gabellini, Civolani, L., Calabrese, F., & Bortolini, M. (2024). A Deep Learning Approach to Predict Supply Chain Delivery Delay Risk Based on Macroeconomic Indicators: A Case Study in the Automotive Sector. *Applied Sciences*, *14*(11), 4688–4688. https://doi.org/10.3390/app14114688

Muñoz-Villamizar, A., Velázquez-Martínez, J. C., Haro, P., Ferrer, A., & Mariño, R. (2021). The environmental impact of fast shipping ecommerce in inbound logistics operations: A case study in Mexico. *Journal of Cleaner Production*, *283*(283), 125400. https://doi.org/10.1016/j.jclepro.2020.125400

Nalla, L., & Reddy, V. (2023). Data Privacy and Security in E-commerce: Modern Database Solutions. *International Journal of Advanced Engineering Technologies and Innovations*, *01*(3), 3.

Ponce, D., Contreras, I., & Laporte, G. (2020). E-commerce shipping through a third-party supply chain. *Transportation Research Part E: Logistics and Transportation Review*, *140*, 101970. https://doi.org/10.1016/j.tre.2020.101970

Siddiqui, Mohammad Zubin, "Optimizing Supply Chain Dynamics using Machine Learning" (2024). Thesis. Rochester Institute of Technology. Accessed from https://repository.rit.edu/cgi/viewcontent.cgi?article=12919&context=theses