**Analysis Report of the Supply Chain Dataset**

**Course name**: MGMT59000 CFA Computing for Analytics

**Team members**: Xiaoyu Guan , Prathyusha Reddy Midudhula

**Submit date**: 05/04/2024

1. **Dataset Introduction**

This dataset “supply\_chain\_data.csv” is found in kaggle, collected from a Fashion and Beauty startup. The dataset is based on the supply chain of Makeup products and encapsulated in 100 entries across 24 distinct columns. Link : <https://www.kaggle.com/datasets/harshsingh2209/supply-chain-analysis/data>

1. **Integer programming for Optimal Order Quantities Calculations**

We used linear programming to determine the optimal number of units to order for each Product type and also for each SKU (Stock Keeping Unit) in a supply chain to minimize costs.

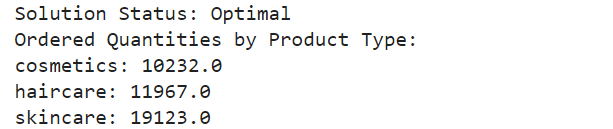
The objective is to minimize the total purchasing cost, which is calculated as the sum of products purchased multiplied by their respective prices.

The Constraints were to meet demand, where each SKU must have enough inventory to meet the sales demand. Non-Negative Orders: The decision variables (number of units ordered for each SKU) are constrained to be non-negative (you can't order a negative number of items).

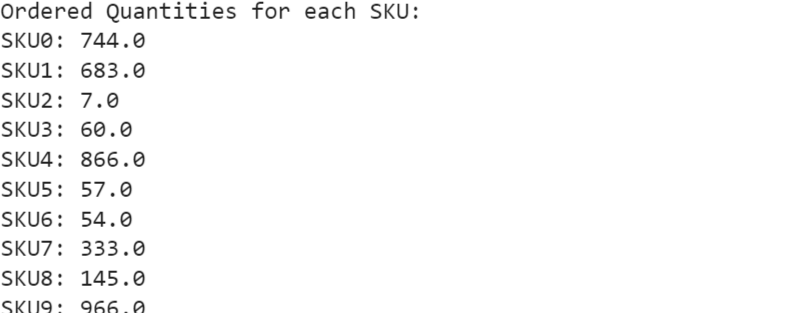
Both the objective function and constraints are linear, which is a requirement in linear programming.

The results are of two types

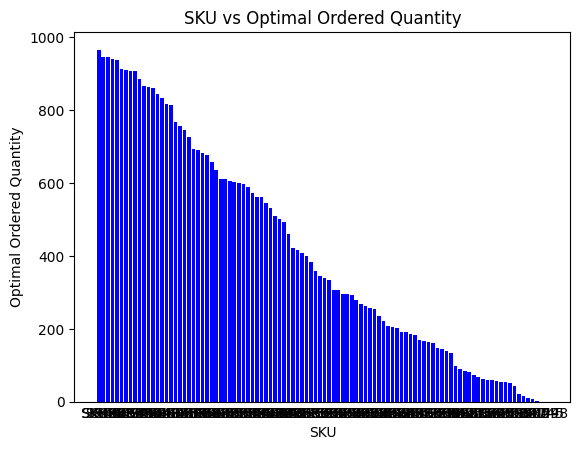
1. The overall product type Optimal order quantities:



1. Each SKU level Optimum order quantities:



Graphing the results we observe the optimal quantities varies hugely from a maximum of 966 units to a minimum of 7 units.



Key Points in the Results:

High Order Quantities: SKUs like SKU9 (966 units), SKU10 (945 units), and SKU98 (860 units) have high ordered quantities, which could indicate a high sales volume or low existing stock for these items.

Low or Zero Order Quantities: SKUs like SKU45 and SKU48 have zero ordered quantities. This could imply that the current stock levels already meet or exceed the forecasted demand for these products.

Medium Order Quantities: Most SKUs have varied quantities, indicating a diverse demand across different products, and these quantities are tailored to meet each specific demand.

Key Application of the Results:

Inventory Management: The model helps in maintaining sufficient inventory to meet customer demands without overstocking, which is crucial in managing storage costs and reducing wastage due to unsold inventory.

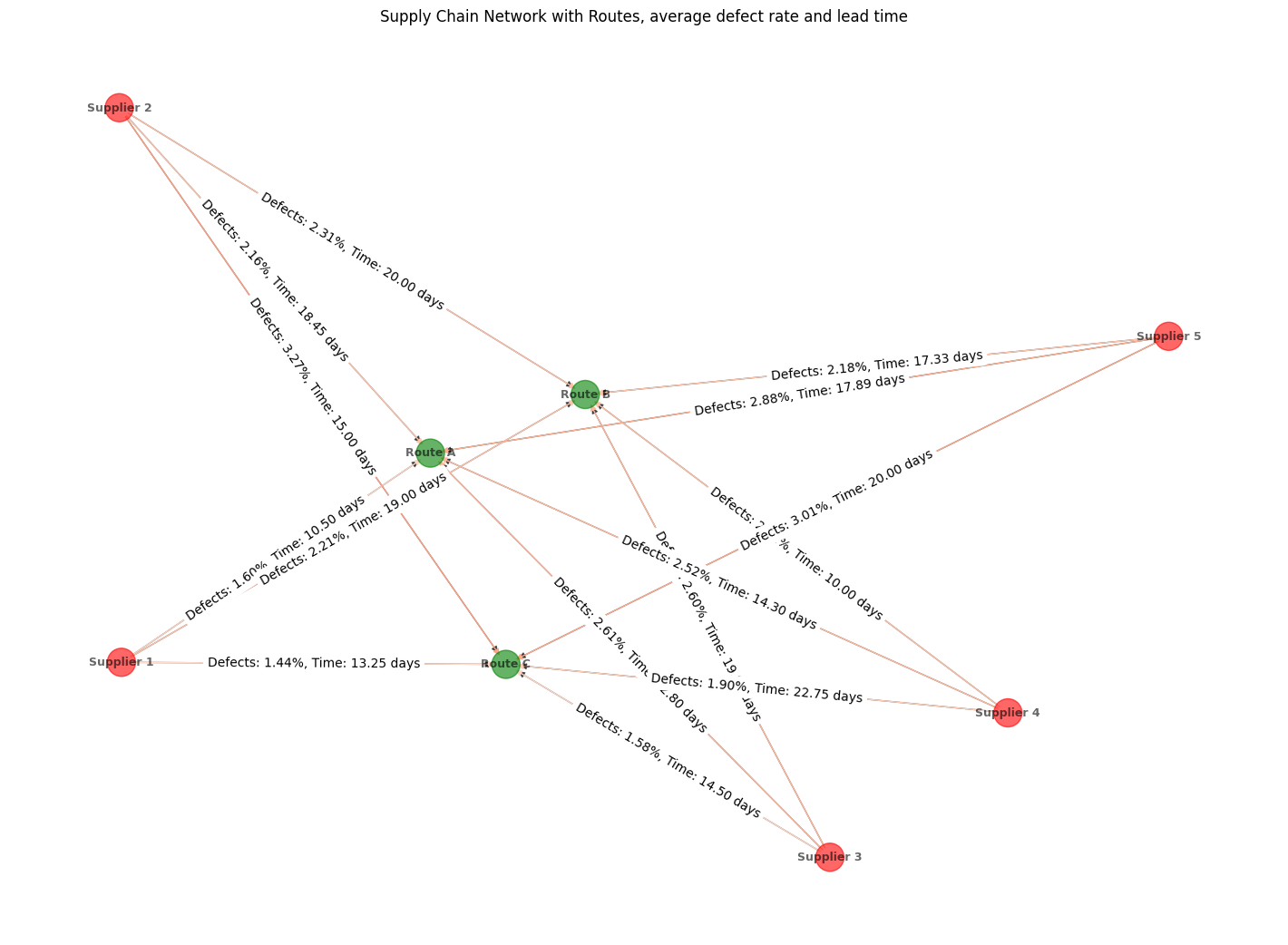
Cost Efficiency: By calculating the exact number of products needed, the model also aims to optimize spending, ensuring that investment in inventory is aligned with sales expectations.

Operational Decisions:

Purchasing Decisions: These results can be directly used to place orders with suppliers, ensuring that each product type is replenished according to the predicted demand.

Financial Planning: Understanding the quantity and cost associated with each SKU aids in better financial planning and budget allocation for inventory purchasing.

1. **Supply Chain Network Graph with Route level average defect rate and lead time**

Route level average defect rate and lead time

The graph shows a network of suppliers connected to various routes, indicating how products are distributed from these suppliers through specific pathways. Each supplier is linked to multiple routes, suggesting a complex distribution network.

Edges are labeled with defect rates and lead times, which vary across routes. For instance, routes connected to Supplier 3 and Supplier 5 exhibit higher defect rates and longer lead times, suggesting issues in quality control or logistical efficiency in these segments.

The variability in defect rates and lead times across different routes can point to inconsistent quality control processes or varying efficiencies in logistics. For example, some routes like those connected to Supplier 1 have lower defect rates and shorter lead times, indicating more reliable performance.

Key Application of the Results:

1. Quality Control Improvement: We can focus on routes with higher defect rates for targeted quality control improvements. This could involve enhancing inspection processes or re-evaluating supplier contracts and manufacturing practices.
2. Logistics Optimization: Analysing routes with longer lead times can help in identifying bottlenecks in the supply chain. We can consider alternative routing options, adjust inventory levels, or negotiate better shipping terms to improve efficiency.
3. Risk Management: Understanding which routes and suppliers are linked to higher defect rates and longer lead times can help in developing risk mitigation strategies. This might include diversifying suppliers or creating contingency plans for routes that frequently encounter delays or quality issues.
4. Strategic Planning: The network visualization supports strategic decision-making by clearly highlighting the strengths and weaknesses of different parts of the supply chain. This can guide investments in logistics infrastructure, technology upgrades, or partnerships.

This graph is useful in analyzing the complexities of supply chain logistics, that can help in optimizing operations and improving overall supply chain resilience and efficiency.

1. **Machine learning : neural network implementation to Predict Delays**

Built a model for predicting delays in shipments, using neural networks.

Used 80% data for training the model and 20% to test the model.

Components of the Neural Network in the Code:

Input Layer: The features (Shipping times, Lead times, Shipping costs) are treated as input neurons. These inputs are fed directly to the output layer without any intermediate processing, which is typical in a single-layer perceptron.

Weights: The code initializes a set of weights (weights = 2 \* np.random.random((X\_train.shape[1], 1)) - 1) that connect each input neuron to the output neuron. These weights are adjusted during training to minimize the prediction error.

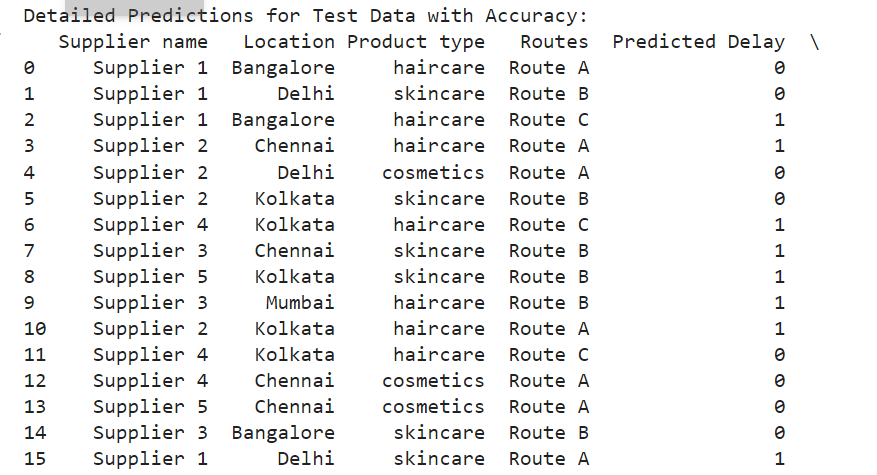
Activation Function: A sigmoid function is used as the activation function (def sigmoid(x): return 1 / (1 + np.exp(-x))). The sigmoid function outputs a value between 0 and 1, making it suitable for binary classification tasks such as predicting whether a shipment will be delayed.

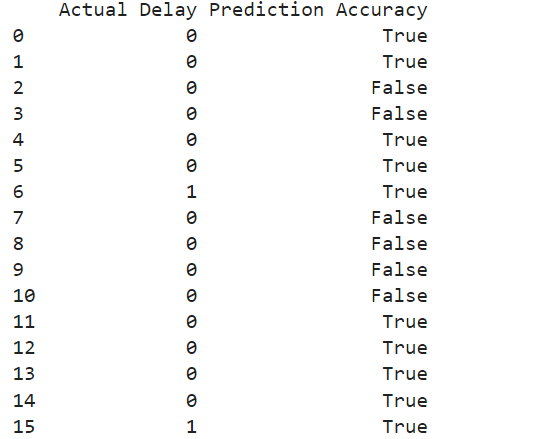
Output Layer: The output layer consists of a single neuron that uses the sigmoid activation function to produce the final prediction. This output can be interpreted as the probability of a shipment being delayed.

Training (Learning) Process: The network is trained using a simple form of the gradient descent algorithm. In each iteration of the training loop, the following steps occur:

* Forward Propagation: \* Calculate the predicted output (outputs = sigmoid(np.dot(input\_layer, weights))) based on the current weights.
* Error Calculation:\* Determine the difference between the actual labels and the predicted labels (error = y\_train - outputs.squeeze()). 3.Backpropagation: Compute the gradient of the loss function with respect to the weights. This involves taking the derivative of the sigmoid function (sigmoid\_derivative(outputs)) and adjusting the weights based on the gradient (weights += np.dot(input\_layer.T, adjustments)).

Prediction: After training, the network uses the learned weights to make predictions on the test data. The output neuron's activation provides the probability that a given shipment will be delayed.





We ended with a model that predicted if there is a delay accurately 70% of the time.

Error Analysis:

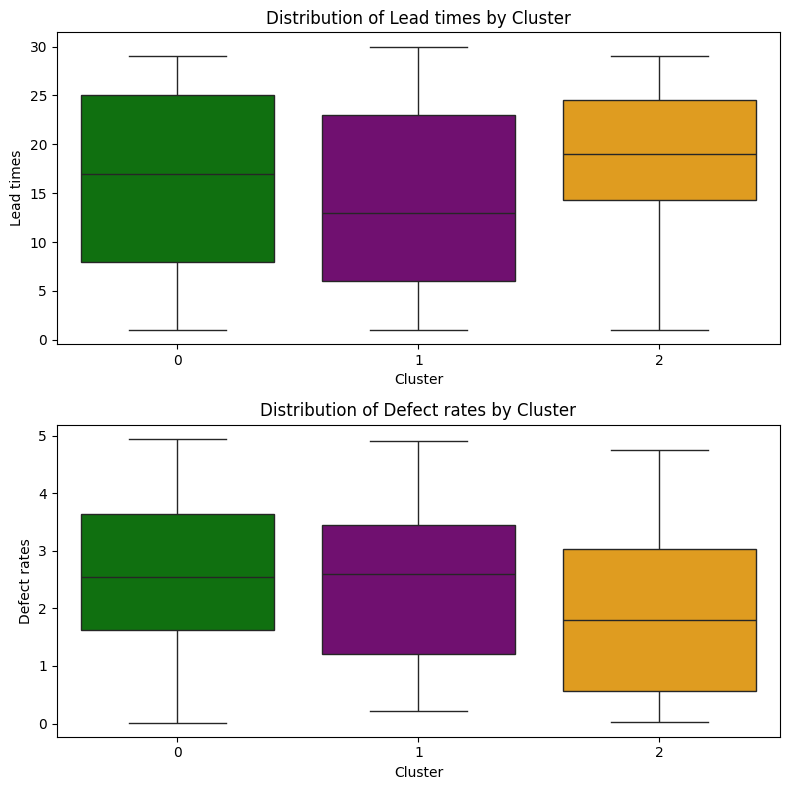
False Predictions: Notably, predictions involving Route A and Route B (e.g., entries 2, 3, 7, 8, 9, 10) frequently resulted in false negatives, where delays were predicted but did not occur. This indicates potential overfitting to certain features or noise within the data affecting these specific routes.

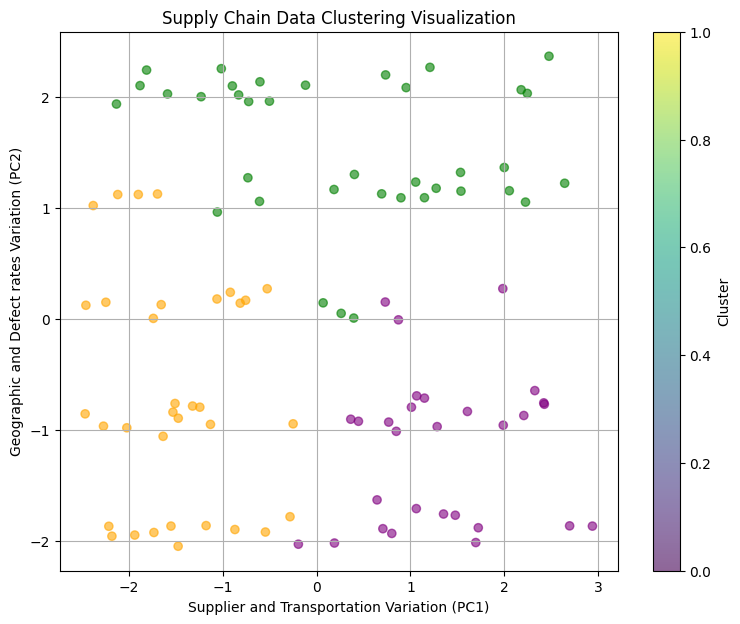
Successful Predictions: The model successfully predicted delays and non-delays across various suppliers and routes (e.g., entries 0, 1, 6, 15, 16), particularly where actual delays matched the prediction, suggesting that under certain conditions the model performs well.

With more data this model can be further improved and can be used for accurate predictions and thus help the business by making the necessary adjustments or changes in the supply chain routes for better outcomes.

1. **Suppliers Clustering**

We applied the k-means technique to cluster the suppliers into three groups according to their lead time and defect rates. This model leverages the balance of product quality and efficiency and could potentially help the company to choose more proper suppliers in the future. Here are the results present in the box and PCA plot.





Explanation for different group:

* **Cluster 0 (green)**: Present a lowest average lead time and average defect rate in the box plot. PCA plot shows a cluster on the upper likely signifies a more concentrated location and higher defect rate, positioned centrally along PC1 also implies this cluster might have suppliers with a balanced range of supplier names or transportation modes.
* **Cluster 1 (purple)**: Box plot shows a lowest average lead time and average defect rate. PCA implies this group consists of a diverse geographic location but concentrates supplier names within the dataset.
* **Cluster 2 (orange):** This group has the highest average lead time and lowest defect rate in the box plot, which also presents a relative concentrate trend for these two features. PCA indicates suppliers in this group with characteristics or transportation methods that are less common compared to the other clusters.

Based on the clustering result, suppliers in cluster 2 have the longest lead time but higher average quality. Cluster 1 has the lowest lead time but higher defect rate, it’s also mainly concentrated on two suppliers, which are supplier 1 and 2. Cluster 0 has a moderate lead time but also a relatively higher defect rate, which may need to be considered with other factors to evaluate.