Introduction

In the dynamic world of logistics, choosing the right shipment mode is a critical decision. Our Machine Learning project is designed to streamline this process by harnessing data-driven insights. By analyzing historical data, market trends, and contextual factors, we aim to empower businesses to make optimal choices among road, sea, or air transportation modes. With advanced Machine Learning algorithms, we're poised to enhance efficiency, reduce costs, and minimize environmental impact.

Note : We've diligently performed data cleaning using Excel as a preliminary step, ensuring that the foundation of our analysis is robust and free from inconsistencies.

Importing Libraries

In [1]:

```
# Importing important libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings("ignore")
```

Data Exploration

In [2]:

```
# Creating dataframe using pandas
Shipmentmode_raw = pd.read_csv("Shipping_Mode_ML_Data.csv", sep=',', encoding='latin-1')

# Understanding data to further explore (First 5 Entries)
Shipmentmode_raw.head()
```

Out[2]:

	ID	Project Code	PQ #	PO / SO #	ASN/DN #	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipment Mode	...	Unit of Measure (Per Pack)	Line Item Quantity
0	1	100-CI-T01	Pre-PQ Process	SCMS-4	ASN-8	Côte d'Ivoire	PMO - US	Direct Drop	EXW	Air	...	30	19
1	3	108-VN-T01	Pre-PQ Process	SCMS-13	ASN-85	Vietnam	PMO - US	Direct Drop	EXW	Air	...	240	1000
2	4	100-CI-T01	Pre-PQ Process	SCMS-20	ASN-14	Côte d'Ivoire	PMO - US	Direct Drop	FCA	Air	...	100	500
3	15	108-VN-T01	Pre-PQ Process	SCMS-78	ASN-50	Vietnam	PMO - US	Direct Drop	EXW	Air	...	60	31920
4	16	108-VN-T01	Pre-PQ Process	SCMS-81	ASN-55	Vietnam	PMO - US	Direct Drop	EXW	Air	...	60	38000

5 rows × 33 columns

In [3]:
```
# Understanding data to further explore (Last 5 Entries)
Shipmentmode_raw.tail()
```

Out[3]:

	ID	Project Code	PQ #	PO / SO #	ASN/DN #	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipment Mode	...	Unit of Measure (Per Pack)	Qu
10319	86818	103-ZW-T30	FPQ-15197	SO-50020	DN-4307	Zimbabwe	PMO - US	From RDC	N/A - From RDC	Truck	...	60	
10320	86819	104-CI-T30	FPQ-15259	SO-50102	DN-4313	Côte d'Ivoire	PMO - US	From RDC	N/A - From RDC	Truck	...	60	
10321	86821	110-ZM-T30	FPQ-14784	SO-49600	DN-4316	Zambia	PMO - US	From RDC	N/A - From RDC	Truck	...	30	
10322	86822	200-ZW-T30	FPQ-16523	SO-51680	DN-4334	Zimbabwe	PMO - US	From RDC	N/A - From RDC	Truck	...	60	
10323	86823	103-ZW-T30	FPQ-15197	SO-50022	DN-4336	Zimbabwe	PMO - US	From RDC	N/A - From RDC	Truck	...	60	

5 rows × 33 columns

In [4]:
```
# Checking the shape of the Dataframe
Shipmentmode_raw.shape
```

Out[4]:
```
(10324, 33)
```

In [5]:
```
# Checking the information of the columns in the dataframe
Shipmentmode_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10324 entries, 0 to 10323
Data columns (total 33 columns):
 #   Column                        Non-Null Count  Dtype 
---  ------                        --------------  ----- 
 0   ID                            10324 non-null  int64 
 1   Project Code                  10324 non-null  object
 2   PQ #                          10324 non-null  object
 3   PO / SO #                     10324 non-null  object
 4   ASN/DN #                      10324 non-null  object
 5   Country                       10324 non-null  object
 6   Managed By                    10324 non-null  object
 7   Fulfill Via                   10324 non-null  object
 8   Vendor INCO Term              10324 non-null  object
 9   Shipment Mode                 9964 non-null   object
 10  PQ First Sent to Client Date  10324 non-null  object
 11  PO Sent to Vendor Date        10324 non-null  object
 12  Scheduled Delivery Date       10324 non-null  object
 13  Delivered to Client Date      10324 non-null  object
 14  Delivery Recorded Date        10324 non-null  object
 15  Product Group                 10324 non-null  object
```

```
 16  Sub Classification            10324 non-null  object
 17  Vendor                        10324 non-null  object
 18  Item Description              10324 non-null  object
 19  Molecule/Test Type            10324 non-null  object
 20  Brand                         10324 non-null  object
 21  Dosage                        8588 non-null   object
 22  Dosage Form                   10324 non-null  object
 23  Unit of Measure (Per Pack)    10324 non-null  int64
 24  Line Item Quantity            10324 non-null  int64
 25  Line Item Value               10324 non-null  float64
 26  Pack Price                    10324 non-null  float64
 27  Unit Price                    10324 non-null  float64
 28  Manufacturing Site            10324 non-null  object
 29  First Line Designation        10324 non-null  object
 30  Weight (Kilograms)            10324 non-null  int64
 31  Freight Cost (USD)            10324 non-null  object
 32  Line Item Insurance (USD)     10037 non-null  float64
dtypes: float64(4), int64(4), object(25)
memory usage: 2.6+ MB
```

In [6]:

```
# Checking the total null values in the dataframe column wise
Shipmentmode_raw.isnull().sum()
```

Out[6]:

```
ID                                 0
Project Code                       0
PQ #                               0
PO / SO #                          0
ASN/DN #                           0
Country                            0
Managed By                         0
Fulfill Via                        0
Vendor INCO Term                   0
Shipment Mode                    360
PQ First Sent to Client Date       0
PO Sent to Vendor Date             0
Scheduled Delivery Date            0
Delivered to Client Date           0
Delivery Recorded Date             0
Product Group                      0
Sub Classification                 0
Vendor                             0
Item Description                   0
Molecule/Test Type                 0
Brand                              0
Dosage                          1736
Dosage Form                        0
Unit of Measure (Per Pack)         0
Line Item Quantity                 0
Line Item Value                    0
Pack Price                         0
Unit Price                         0
Manufacturing Site                 0
First Line Designation             0
Weight (Kilograms)                 0
Freight Cost (USD)                 0
Line Item Insurance (USD)        287
dtype: int64
```

In [7]:

```
# Checking for total number of unique values in each column
Shipmentmode_raw.nunique()
```

Out[7]:

```
ID                              10324
Project Code                      142
PQ #                             1237
PO / SO #                        6233
ASN/DN #                         7030
```

```
Country                            43
Managed By                          4
Fulfill Via                         2
Vendor INCO Term                    8
Shipment Mode                       4
PQ First Sent to Client Date      765
PO Sent to Vendor Date           1881
Scheduled Delivery Date          2006
Delivered to Client Date         2093
Delivery Recorded Date           2042
Product Group                       5
Sub Classification                  6
Vendor                             73
Item Description                  184
Molecule/Test Type                 86
Brand                              48
Dosage                             54
Dosage Form                        17
Unit of Measure (Per Pack)         31
Line Item Quantity               5065
Line Item Value                  8741
Pack Price                       1175
Unit Price                        183
Manufacturing Site                 88
First Line Designation              2
Weight (Kilograms)               3388
Freight Cost (USD)               6290
Line Item Insurance (USD)        6722
dtype: int64
```

Data Transformation

In [8]:

```
# Filling null values with the relevant values
Shipmentmode_raw['Shipment Mode'].fillna(value="Other", inplace=True)
Shipmentmode_raw['Line Item Insurance (USD)'].fillna(value=0, inplace=True)
Shipmentmode_raw.isnull().sum()
```

Out[8]:

```
ID                                 0
Project Code                       0
PQ #                               0
PO / SO #                          0
ASN/DN #                           0
Country                            0
Managed By                         0
Fulfill Via                        0
Vendor INCO Term                   0
Shipment Mode                      0
PQ First Sent to Client Date       0
PO Sent to Vendor Date             0
Scheduled Delivery Date            0
Delivered to Client Date           0
Delivery Recorded Date             0
Product Group                      0
Sub Classification                 0
Vendor                             0
Item Description                   0
Molecule/Test Type                 0
Brand                              0
Dosage                          1736
Dosage Form                        0
Unit of Measure (Per Pack)         0
Line Item Quantity                 0
Line Item Value                    0
Pack Price                         0
```

```
Unit Price                        0
Manufacturing Site                0
First Line Designation            0
Weight (Kilograms)                0
Freight Cost (USD)                0
Line Item Insurance (USD)         0
dtype: int64
```

In [9]:
```
# Assigning integer value to string as needed to make the column mutual
Shipmentmode_raw['Freight Cost (USD)'] = Shipmentmode_raw['Freight Cost (USD)'].replace(
```

In [10]:
```
# Checking the format and type of the date column
Shipmentmode_raw["PO Sent to Vendor Date"].unique()
```

Out[10]:
```
array(['04-03-2006', '16-08-2006', '29-05-2006', ..., '24-05-2014',
       '03-02-2015', '08-01-2015'], dtype=object)
```

In [11]:
```
# Converting into Date time format from the string format
Shipmentmode_raw["PO Sent to Vendor Date"]= pd.to_datetime(Shipmentmode_raw["PO Sent to
Shipmentmode_raw["Scheduled Delivery Date"]= pd.to_datetime(Shipmentmode_raw["Scheduled
Shipmentmode_raw["Delivered to Client Date"]= pd.to_datetime(Shipmentmode_raw["Delivered
Shipmentmode_raw["Delivery Recorded Date"]= pd.to_datetime(Shipmentmode_raw["Delivery Re

# converting Frieght cost column to float type from the string type
Shipmentmode_raw['Freight Cost (USD)'] = Shipmentmode_raw['Freight Cost (USD)'].astype(f
Shipmentmode_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10324 entries, 0 to 10323
Data columns (total 33 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          10324 non-null  int64
 1   Project Code                10324 non-null  object
 2   PQ #                        10324 non-null  object
 3   PO / SO #                   10324 non-null  object
 4   ASN/DN #                    10324 non-null  object
 5   Country                     10324 non-null  object
 6   Managed By                  10324 non-null  object
 7   Fulfill Via                 10324 non-null  object
 8   Vendor INCO Term            10324 non-null  object
 9   Shipment Mode               10324 non-null  object
 10  PQ First Sent to Client Date  10324 non-null  object
 11  PO Sent to Vendor Date      10324 non-null  datetime64[ns]
 12  Scheduled Delivery Date     10324 non-null  datetime64[ns]
 13  Delivered to Client Date    10324 non-null  datetime64[ns]
 14  Delivery Recorded Date      10324 non-null  datetime64[ns]
 15  Product Group               10324 non-null  object
 16  Sub Classification          10324 non-null  object
 17  Vendor                      10324 non-null  object
 18  Item Description            10324 non-null  object
 19  Molecule/Test Type          10324 non-null  object
 20  Brand                       10324 non-null  object
 21  Dosage                      8588 non-null   object
 22  Dosage Form                 10324 non-null  object
 23  Unit of Measure (Per Pack)  10324 non-null  int64
 24  Line Item Quantity          10324 non-null  int64
 25  Line Item Value             10324 non-null  float64
 26  Pack Price                  10324 non-null  float64
 27  Unit Price                  10324 non-null  float64
 28  Manufacturing Site          10324 non-null  object
 29  First Line Designation      10324 non-null  object
 30  Weight (Kilograms)          10324 non-null  int64
 31  Freight Cost (USD)          10324 non-null  float64
 32  Line Item Insurance (USD)   10324 non-null  float64
```

```
dtypes: datetime64[ns](4), float64(5), int64(4), object(20)
memory usage: 2.6+ MB
```

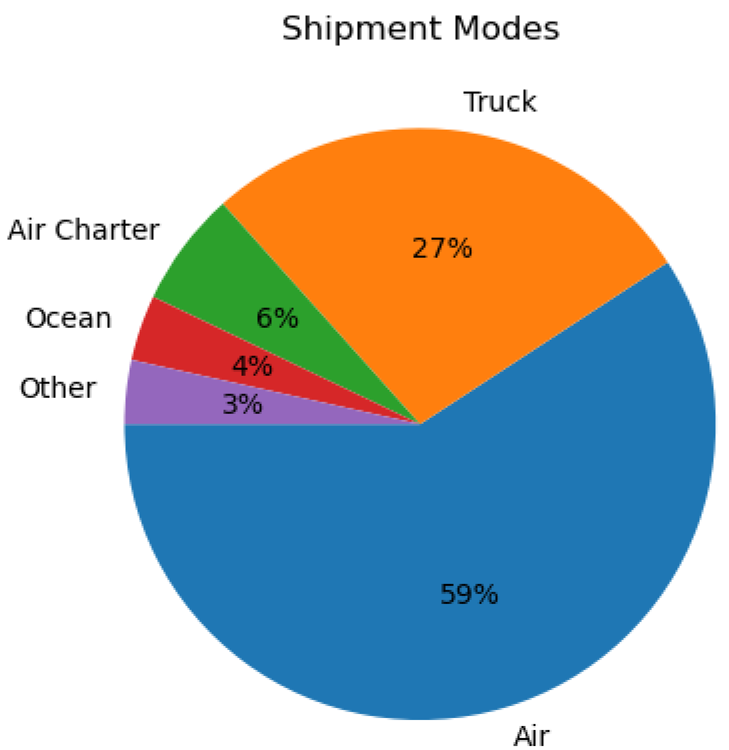
In [12]:

```python
# Replacing string value to make the column mutual
Shipmentmode_raw['Vendor'] = Shipmentmode_raw['Vendor'].replace(['ABBVIE LOGISTICS (FORM
Shipmentmode_raw['Vendor'] = Shipmentmode_raw['Vendor'].replace(['ABBOTT LOGISTICS BV'],
Shipmentmode_raw['Vendor'] = Shipmentmode_raw['Vendor'].replace(['MYLAN LABORATORIES LTD
```

Exploratory Data Analysis

In [13]:

```python
# Creating visualisation to check the density of the modes
ShippingMode = Shipmentmode_raw["Shipment Mode"].value_counts()
mlabels = (np.array(ShippingMode.index))
sizes = (np.array((ShippingMode / ShippingMode.sum())*100))
plt.pie(sizes, labels=mlabels, autopct='%0.f%%', startangle=180)
plt.title('Shipment Modes')
plt.show()
```

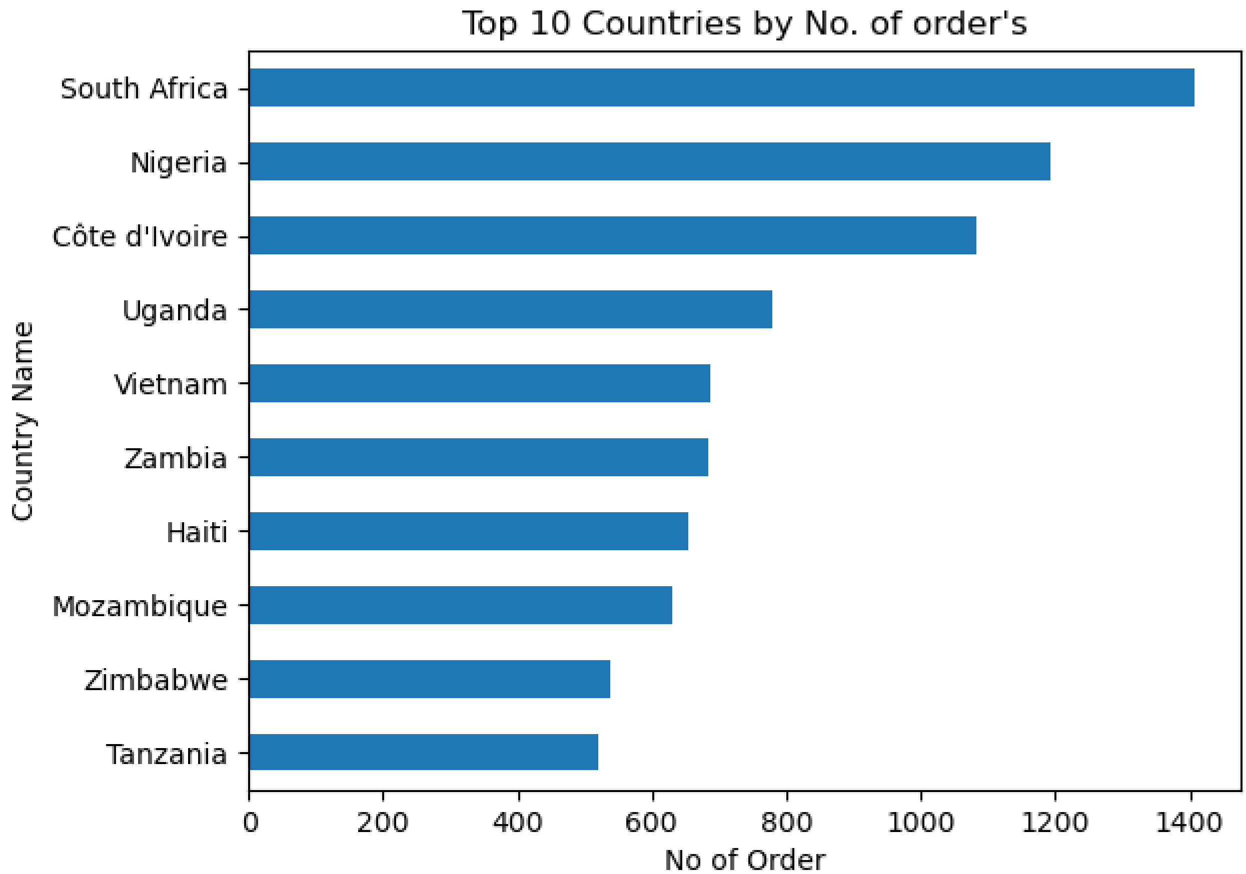


In [14]:

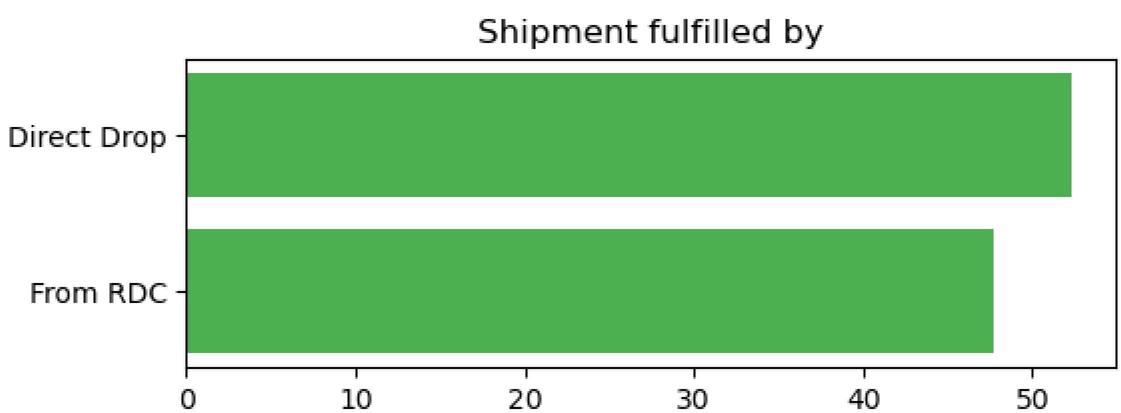
```python
# Creating the visualisation to check the top 10 countries by their no of orders
top10country = Shipmentmode_raw["Country"].value_counts().iloc[:10]
top10country_sorted = top10country.sort_values(ascending=True)
top10country_sorted.plot(kind='barh')
plt.title('Top 10 Countries by No. of order\'s')
plt.xlabel('No of Order')
plt.ylabel('Country Name')
```

Out[14]:

```
Text(0, 0.5, 'Country Name')
```

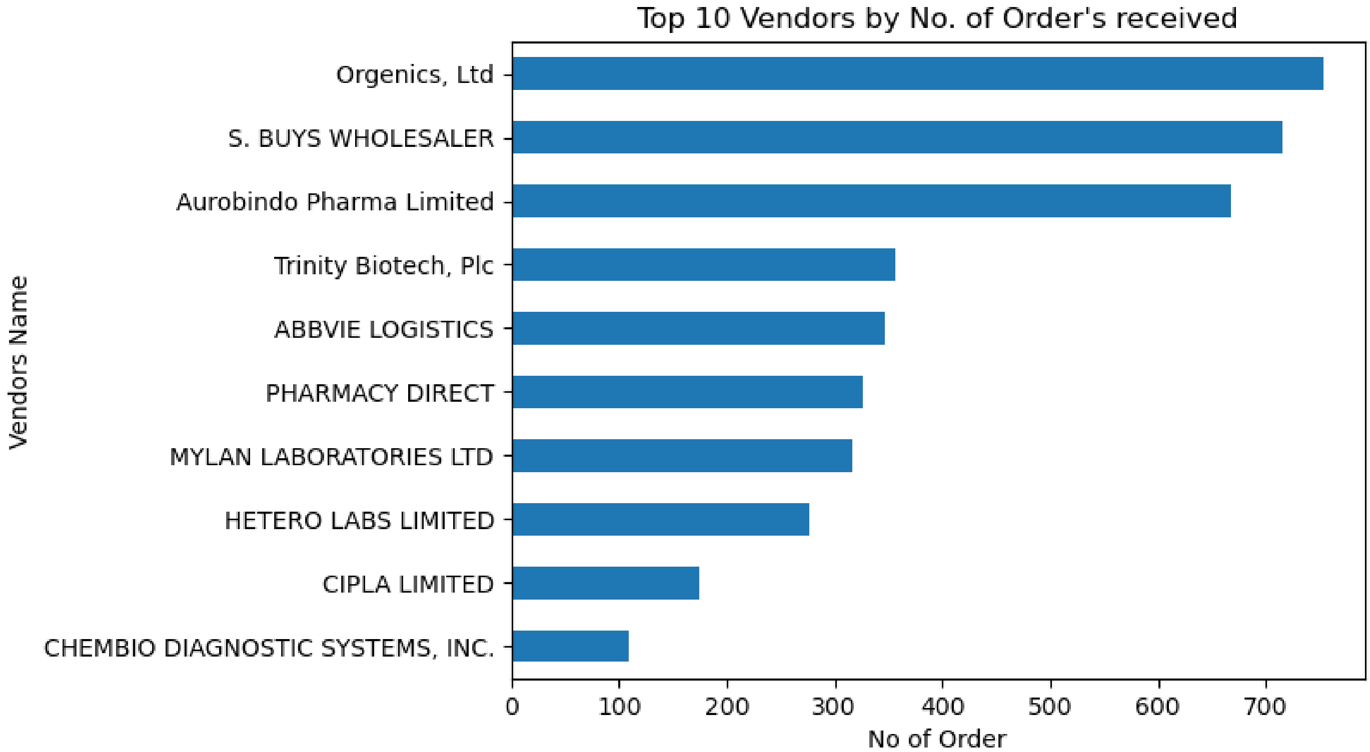


In [15]:
```
# Creating Visuvalisation to check the the method of the shipment
FulfillVia = Shipmentmode_raw["Fulfill Via"].value_counts()
mlabels = (np.array(FulfillVia.index))
sizes = (np.array((FulfillVia / FulfillVia.sum())*100))
plt.figure(figsize=(6,2))
plt.barh(mlabels, sorted(sizes), color = "#4CAF50")
plt.title('Shipment fulfilled by')
plt.show()
```

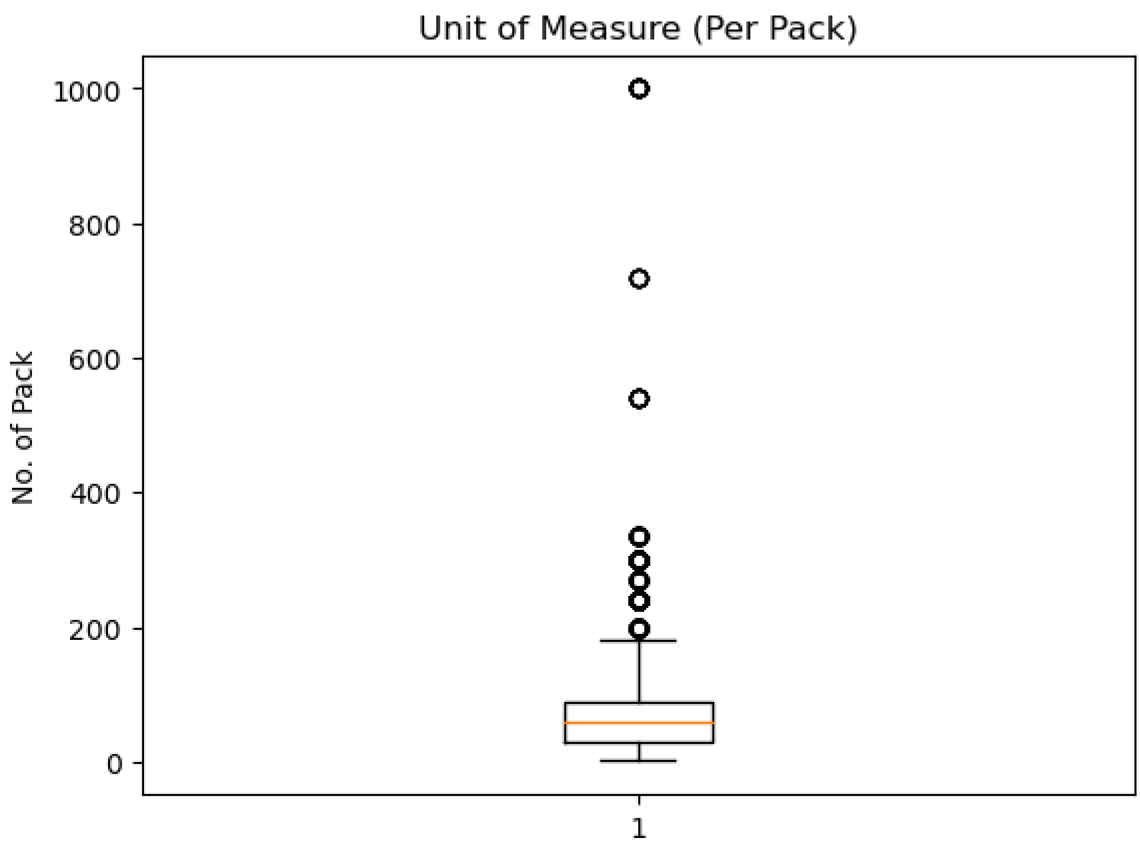


In [16]:
```
# Creating a visualisation to know Top 10 vendors after
top10vendor = (Shipmentmode_raw['Vendor'].value_counts()).iloc[1:11]
top10vendor_sorted = top10vendor.sort_values(ascending=True)
top10vendor_sorted.plot(kind='barh')
plt.title('Top 10 Vendors by No. of Order\'s received')
plt.xlabel('No of Order')
plt.ylabel('Vendors Name')
```

Out[16]:
```
Text(0, 0.5, 'Vendors Name')
```

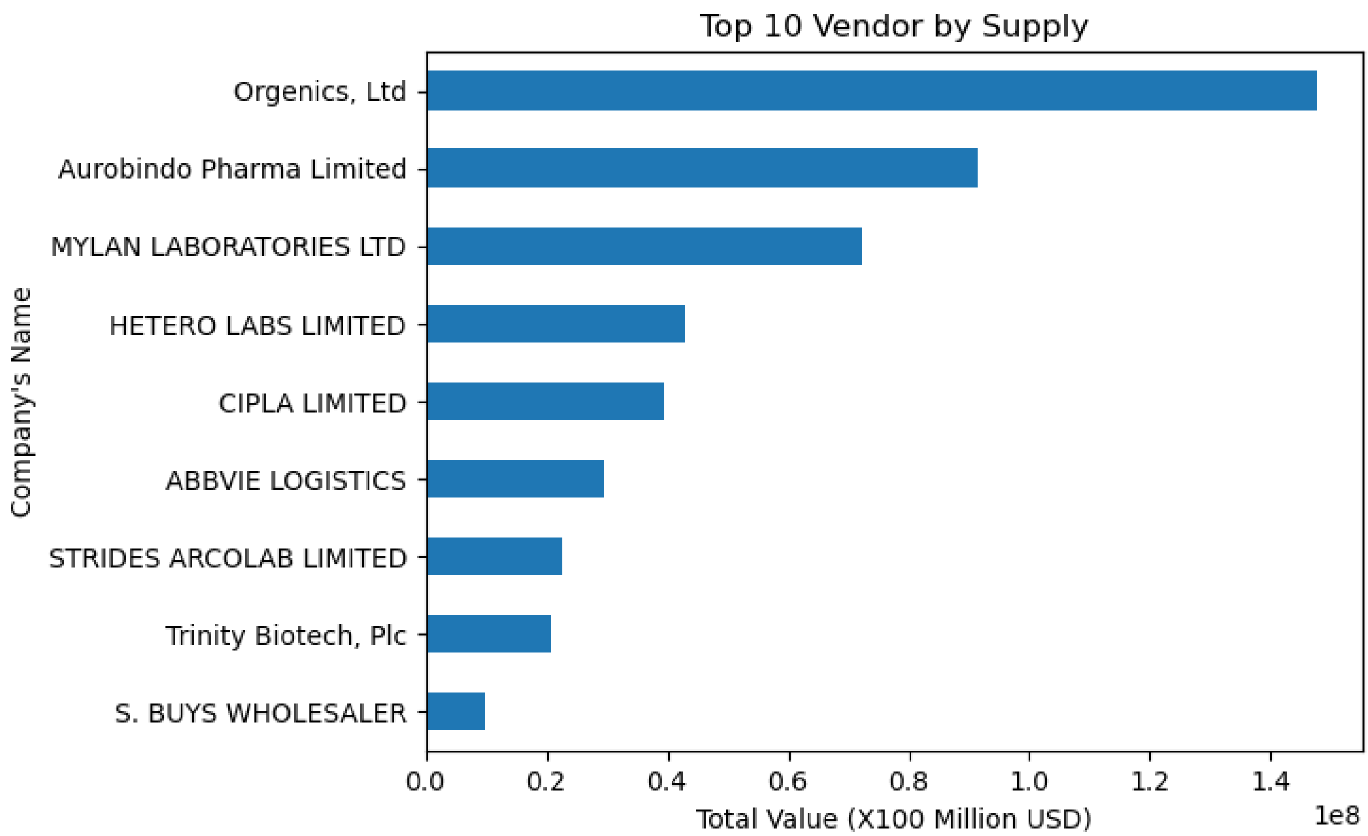


In [17]:
```
# Creating visualisation to check the outliars
plt.boxplot(Shipmentmode_raw['Unit of Measure (Per Pack)'])
plt.title('Unit of Measure (Per Pack)')
plt.ylabel('No. of Pack')
plt.show()
```

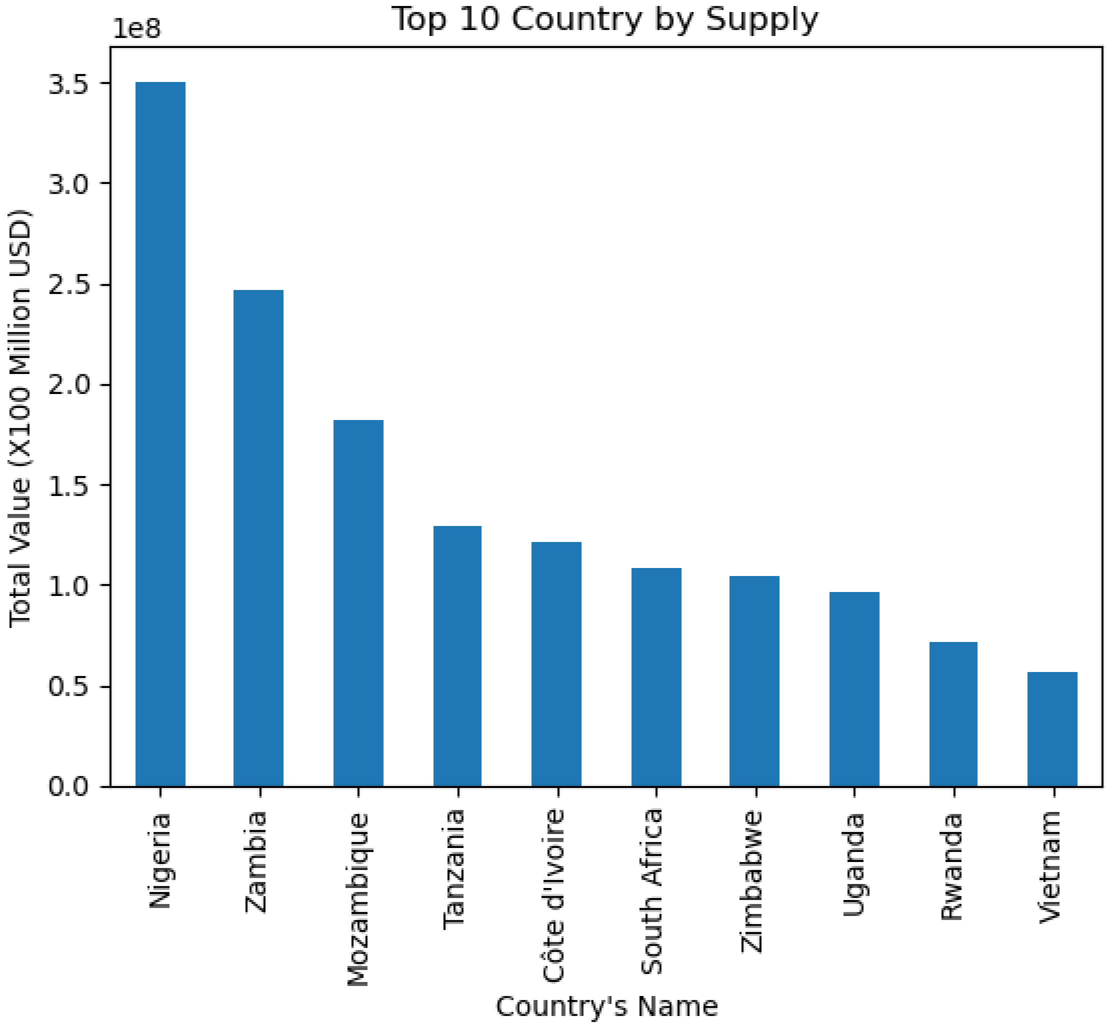


In [18]:
```
# Creating Visualisation to check for the top 10 company by their supply (Except RDC)
top10vendor = Shipmentmode_raw.groupby('Vendor')['Line Item Value'].sum().nlargest(10).i
top10vendor_sorted = top10vendor.sort_values(ascending=True)
top10vendor_sorted.plot(kind='barh')
plt.title('Top 10 Vendor by Supply')
plt.xlabel('Total Value (X100 Million USD)')
```

```
plt.ylabel('Company\'s Name')
plt.show()
```

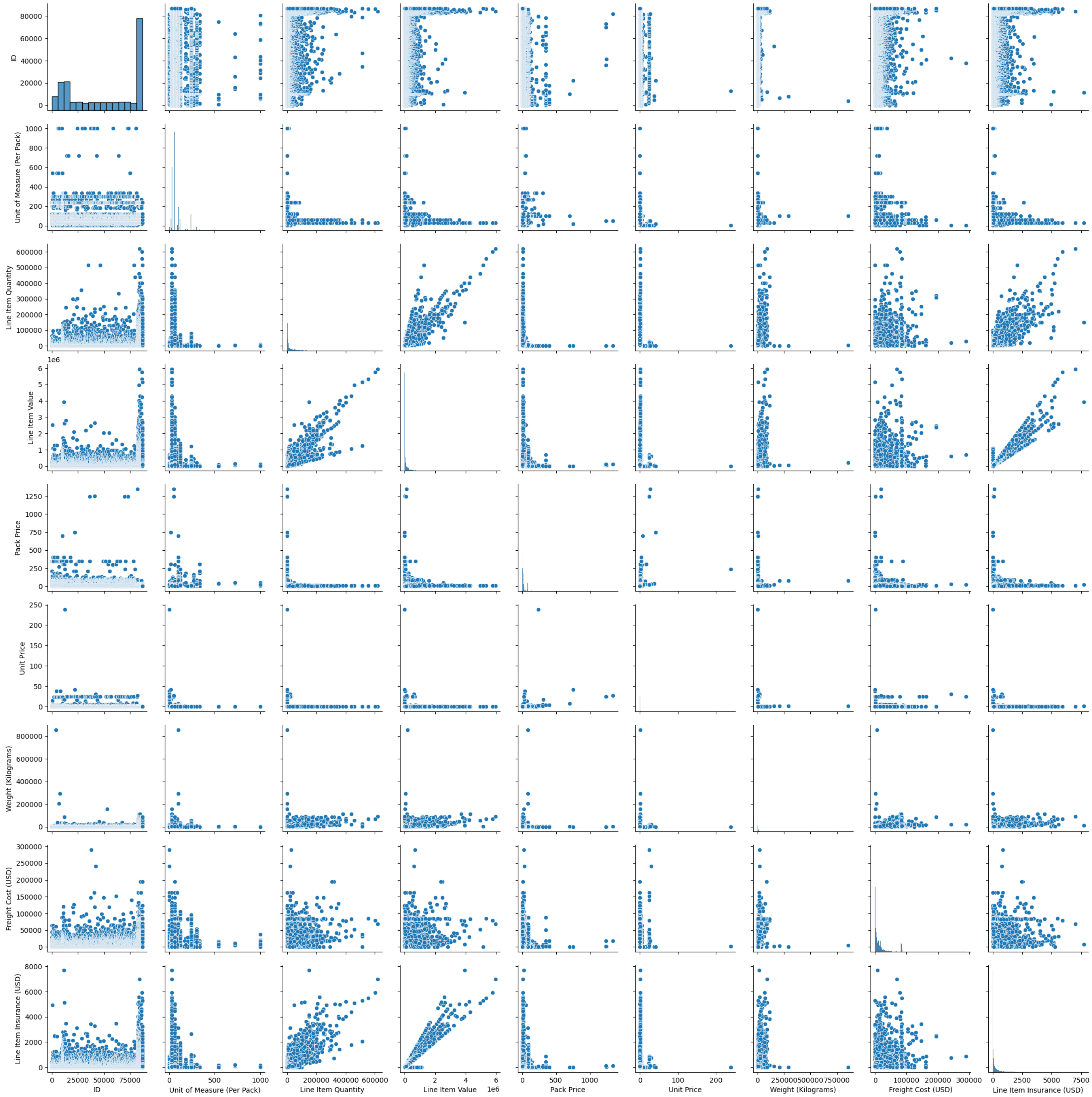


In [19]:
```
# Creating Visualisation to check for top 10 country by their supply
top10Country = Shipmentmode_raw.groupby('Country')['Line Item Value'].sum().nlargest(10)
top10Country_sorted = top10Country.sort_values(ascending=False)
top10Country_sorted.plot(kind='bar')
plt.title('Top 10 Country by Supply')
plt.xlabel('Country\'s Name')
plt.ylabel('Total Value (X100 Million USD)')
plt.show()
```



In [20]:

```
# Creating Visualisation to check the relation between columns in dataframe
sns.pairplot(Shipmentmode_raw)
plt.show()
```



Feature Engineering

In [21]:

```python
# Finding necessary data for the Machine Learning Model which will effect the prediction
Shipmentmode_raw['Estimated delivery days'] = (Shipmentmode_raw['Scheduled Delivery Date
Shipmentmode_raw['Actual delivery days'] = (Shipmentmode_raw['Delivered to Client Date']

Shipmentmode_raw['Total Cost'] = (Shipmentmode_raw['Line Item Value'] + Shipmentmode_raw

Shipmentmode_raw.head()
```

Out[21]:

	ID	Project Code	PQ #	PO / SO #	ASN/DN #	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipment Mode	...	Pack Price	Unit Price	Manufa
0	1	100-CI-T01	Pre-PQ Process	SCMS-4	ASN-8	Côte d'Ivoire	PMO - US	Direct Drop	EXW	Air	...	29.00	0.97	Ranb Chem
1	3	108-VN-	Pre-PQ Process	SCMS-13	ASN-85	Vietnam	PMO - US	Direct Drop	EXW	Air	...	6.20	0.03	Aurobi

		T01												
2	4	100-CI-T01	Pre-PQ Process	SCMS-20	ASN-14	Côte d'Ivoire	PMO - US	Direct Drop	FCA	Air	...	80.00	0.80	ABBVI Wi
3	15	108-VN-T01	Pre-PQ Process	SCMS-78	ASN-50	Vietnam	PMO - US	Direct Drop	EXW	Air	...	3.99	0.07	I Paonta
4	16	108-VN-T01	Pre-PQ Process	SCMS-81	ASN-55	Vietnam	PMO - US	Direct Drop	EXW	Air	...	3.20	0.05	Aurobi

5 rows × 36 columns

In [22]:
```
#Checking for all the column name
Shipmentmode_raw.columns
```

Out[22]:
```
Index(['ID', 'Project Code', 'PQ #', 'PO / SO #', 'ASN/DN #', 'Country',
       'Managed By', 'Fulfill Via', 'Vendor INCO Term', 'Shipment Mode',
       'PQ First Sent to Client Date', 'PO Sent to Vendor Date',
       'Scheduled Delivery Date', 'Delivered to Client Date',
       'Delivery Recorded Date', 'Product Group', 'Sub Classification',
       'Vendor', 'Item Description', 'Molecule/Test Type', 'Brand', 'Dosage',
       'Dosage Form', 'Unit of Measure (Per Pack)', 'Line Item Quantity',
       'Line Item Value', 'Pack Price', 'Unit Price', 'Manufacturing Site',
       'First Line Designation', 'Weight (Kilograms)', 'Freight Cost (USD)',
       'Line Item Insurance (USD)', 'Estimated delivery days',
       'Actual delivery days', 'Total Cost'],
      dtype='object')
```

In [23]:
```
#Shipmentmode_raw.to_csv('final_dataset.csv')
```

Feature Selection

In [24]:
```
#Selecting Feature and Label data to apply Machine Learning Model
shipmentmode = Shipmentmode_raw[['Country', 'Fulfill Via', 'Vendor INCO Term', 'Vendor',
                 'Estimated delivery days', 'Actual delivery days', 'Total Cost', 'Shipme

# Checking New dataframe
shipmentmode.head()
```

Out[24]:

	Country	Fulfill Via	Vendor INCO Term	Vendor	Weight (Kilograms)	Estimated delivery days	Actual delivery days	Total Cost	Shipment Mode
0	Côte d'Ivoire	Direct Drop	EXW	RANBAXY Fine Chemicals LTD.	13	90	90	1331.34	Air
1	Vietnam	Direct Drop	EXW	Aurobindo Pharma Limited	358	90	90	10721.50	Air
2	Côte d'Ivoire	Direct Drop	FCA	Abbott GmbH & Co. KG	171	90	90	41653.78	Air
3	Vietnam	Direct Drop	EXW	SUN PHARMACEUTICAL INDUSTRIES LTD (RANBAXY LAB...	1855	90	90	143367.86	Air
4	Vietnam	Direct Drop	EXW	Aurobindo Pharma Limited	7590	90	90	167050.08	Air

Encoding

In [25]:
```
# Transforming string data to numeric data using One Hot encoding for applying Machine L
shipmentmode_f = pd.get_dummies(shipmentmode, columns=["Country", "Fulfill Via", "Vendor
```

In [26]:
```
# Transforming Label data to integer value using Label encoding
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
shipmentmode_f["Shipment Mode"]= label_encoder.fit_transform(shipmentmode["Shipment Mode

shipmentmode_f["Shipment Mode"].unique()
```

Out[26]:
```
array([0, 3, 4, 1, 2])
```

In [27]:
```
# Checking transformed data to apply Machine Learning Models
shipmentmode_f.head()
```

Out[27]:

	Weight (Kilograms)	Estimated delivery days	Actual delivery days	Total Cost	Shipment Mode	Country_Afghanistan	Country_Angola	Country_Belize
0	13	90	90	1331.34	0	0	0	0
1	358	90	90	10721.50	0	0	0	0
2	171	90	90	41653.78	0	0	0	0
3	1855	90	90	143367.86	0	0	0	0
4	7590	90	90	167050.08	0	0	0	0

5 rows × 131 columns

Dividing into Train and Test

In [28]:
```
# Marking data to feature and Label data
X = shipmentmode_f.drop('Shipment Mode',axis=1)
Y= shipmentmode_f['Shipment Mode']
```

In [29]:
```
# Dividing feature and Label data to further Train and test data
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.20, random_state=1
```

Machine Learning

In [30]:
```
# Importing Machine learning models and validation libraries
from sklearn.model_selection import cross_val_score, cross_val_predict
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report
```

In [31]:
```python
# Creating list of all Machine learning Models which will be applied
models = [
    ("Logistic Regression", LogisticRegression(max_iter=1000)),
    ("Decision Tree", DecisionTreeClassifier()),
    ("Random Forest", RandomForestClassifier()),
    ("Support Vector Machine", SVC()),
    ("K-Nearest Neighbors", KNeighborsClassifier())
]
```

In [32]:
```python
# Applying Machine Learning models and Validation techniques using for loop
for model_name, model in models:
    scores = cross_val_score(model, X, Y, cv=5)  # 5-fold cross-validation
    avg_score = np.mean(scores)

    predicted = cross_val_predict(model, X, Y, cv=5)  # Predictions for calculating accu

    print(f"{model_name} - Cross-Validation Scores: {scores}")
    print(f"{model_name} - Average Score: {avg_score:.2f}\n")

    class_report = classification_report(Y, predicted)
    print(f"{model_name} - Classification Report:\n{class_report}\n")
```

```
Logistic Regression - Cross-Validation Scores: [0.64116223 0.74188862 0.6401937  0.65036
32  0.62839147]
Logistic Regression - Average Score: 0.66

Logistic Regression - Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.95      0.78      6113
           1       0.17      0.01      0.03       650
           2       0.00      0.00      0.00       371
           3       0.00      0.00      0.00       360
           4       0.65      0.36      0.47      2830

    accuracy                           0.66     10324
   macro avg       0.30      0.26      0.25     10324
weighted avg       0.58      0.66      0.59     10324


Decision Tree - Cross-Validation Scores: [0.71138015 0.83099274 0.88523002 0.87215496 0.
68895349]
Decision Tree - Average Score: 0.80

Decision Tree - Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.80      0.84      6113
           1       0.77      0.81      0.79       650
           2       0.72      0.80      0.76       371
           3       0.36      0.43      0.39       360
           4       0.73      0.82      0.77      2830

    accuracy                           0.79     10324
   macro avg       0.69      0.73      0.71     10324
weighted avg       0.80      0.79      0.80     10324


Random Forest - Cross-Validation Scores: [0.7433414  0.86198547 0.89588378 0.89491525 0.
72771318]
Random Forest - Average Score: 0.82

Random Forest - Classification Report:
              precision    recall  f1-score   support
```

```
           0       0.89      0.84      0.86      6113
           1       0.81      0.84      0.82       650
           2       0.86      0.83      0.84       371
           3       0.41      0.37      0.39       360
           4       0.75      0.84      0.79      2830

    accuracy                           0.82     10324
   macro avg       0.74      0.74      0.74     10324
weighted avg       0.83      0.82      0.83     10324


Support Vector Machine - Cross-Validation Scores: [0.59661017 0.57433414 0.60871671 0.61
549637 0.62063953]
Support Vector Machine - Average Score: 0.60

Support Vector Machine - Classification Report:
              precision    recall  f1-score   support

           0       0.61      0.98      0.75      6113
           1       0.17      0.01      0.01       650
           2       0.00      0.00      0.00       371
           3       0.00      0.00      0.00       360
           4       0.43      0.08      0.14      2830

    accuracy                           0.60     10324
   macro avg       0.24      0.21      0.18     10324
weighted avg       0.49      0.60      0.49     10324


K-Nearest Neighbors - Cross-Validation Scores: [0.64745763 0.72687651 0.61937046 0.60823
245 0.56879845]
K-Nearest Neighbors - Average Score: 0.63

K-Nearest Neighbors - Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.86      0.77      6113
           1       0.14      0.08      0.10       650
           2       0.14      0.06      0.08       371
           3       0.18      0.03      0.06       360
           4       0.56      0.42      0.48      2830

    accuracy                           0.63     10324
   macro avg       0.34      0.29      0.30     10324
weighted avg       0.58      0.63      0.60     10324
```

Conclusion

In our exploration of various machine learning models for the task of shipment mode classification, we have observed diverse performances across different algorithms.

- Logistic Regression, while delivering an average accuracy of 66%, faces challenges in handling the imbalanced nature of the dataset, resulting in limited success in identifying minority classes.
- Decision Trees, on the other hand, demonstrated a promising average accuracy of 80%, showing robustness in capturing complex decision boundaries. It exhibited decent precision and recall for most classes, particularly class 0.

- Random Forest, with an average accuracy of 82%, further improved upon the decision tree model. Its ensemble nature helped enhance generalization, providing a strong balance between precision and recall for multiple classes.

- Support Vector Machine, despite having a comparatively lower average accuracy of 60%, excelled in precision for class 0. However, it struggled with minority classes due to class imbalance.

- K-Nearest Neighbors, with an average accuracy of 63%, performed reasonably well in classifying class 0 but encountered challenges in distinguishing minority classes.

In summary, the **Random Forest** model emerged as the top performer in this classification task, offering a robust balance between precision and recall for multiple classes. However, addressing class imbalance remains a crucial challenge across all models. Further refinement and optimization are essential to improve the classification of minority classes, ultimately enhancing the practical utility of these models in real-world shipment mode selection scenarios.