SUPPLY CHAIN ANALYTICS | MBA (BA) | SEMESTER 3

Supply Chain Shipment Price Forecasting

GROUP 02

Shikar Maheshwari – 20020845025
Anoushka Viegas – 20020845027
Taksande Sandeep Ravindra – 20020845030

Supply Chain Shipment Price Forecasting

Introduction

Forecasting shipments may not seem like the most important aspect of a company’s cargo journey. However, in the absence of a forecast a company may find itself having cargo but no ship to load it on — or vice versa. As a result, precise predicting has a much greater impact than it may appear.
A good forecast:

- Reduces the degree of uncertainty regarding the future and this in turn allows for a company to make better strategic choices.
- It increases customer satisfaction by increasing the ease of doing business with them.
- It increases the cost-efficiency of a company.

Role of Logistics in Supply Chain Management

Logistics - the management of physical, informational, and human flows in order to optimise them and avoid excessive waste of resources - plays a significant role in supply chains.
The goal of effective logistics management is to maximise the company's competitiveness and profitability, as well as the supply chain's whole network structure, including the end-user. In this regard, the integration and introduction of innovations into supply chain and logistics activities should strive to improve the overall productivity of all participants.

Demand Forecasting

The practise of accurately anticipating demand for products, services, and shipments along the supply chain is known as logistics demand forecasting. The forecasting model needs to be based on a combination of historical data and several external variables, to achieve this. The best model is one that allows for automatic forecast adjustments to reflect new clients or other company developments while also increasing accuracy.

Benefits of Demand Forecasting

1. Reduced operations costs
2. Dynamic pricing
3. Increased employee efficiency
4. Improved resource planning and scalability

Project

Problem:

When it comes to purchasing art, navigating the logistics can be challenging. These include the following:

- Collection management that works
- Shipping the paintings, antiques, sculptures, and other collectibles to their respective destinations after purchase

While many companies have made shipping consumer products a reasonably quick and straightforward process, the same laws do not necessarily apply when shipping paintings or transporting antiques and collectibles.

Task

A corporation buys sculptures from artists all around the world and sells them. Based on available data we have estimated the cost of shipping these sculptures to clients.

About the Data

The columns provided in the dataset are as follows:

Column name	**Description**
Customer Id	Represents the unique identification number of the customers
Artist Name	Represents the name of the artist
Artist Reputation	Represents the reputation of an artist in the market (the greater the reputation value, the higher the reputation of the artist in the market)
Height	Represents the height of the sculpture
Width	Represents the width of the sculpture
Weight	Represents the weight of the sculpture
Material	Represents the material that the sculpture is made of
Price Of Sculpture	Represents the price of the sculpture
Base Shipping Price	Represents the base price for shipping a sculpture
International	Represents whether the shipping is international
Express Shipment	Represents whether the shipping was in the express (fast) mode

Installation Included	Represents whether the order had installation included in the purchase of the sculpture
Transport	Represents the mode of transport of the order
Fragile	Represents whether the order is fragile
Customer Information	Represents details about a customer
Remote Location	Represents whether the customer resides in a remote location
Scheduled Date	Represents the date when the order was placed
Delivery Date	Represents the date of delivery of the order
Customer Location	Represents the location of the customer
Cost	Represents the cost of the order

Procedure:

We built a forecasting model using the following steps:

1. The data was loaded on Google Colab (https://colab.research.google.com/drive/1qd30Wn53EFKgzu6uJTYChoMh7xmw3yj9?usp=sharing)
2. An initial exploration of the data showed that the data comprised of 6500 rows and 20 columns.
3. A cursory observation of the data showed that some values in the "cost" column were negative.
   a. These were converted to a positive value for further use
   b. This conversion was done using the absolute method

```
[ ]  tp['Cost']=tp['Cost'].abs()
```

4. A cursory observation of the columns 'Scheduled Dates' and 'Delivery Dates' showed that while the programme did not identify the columns as dates.
   a. This was overcome by converting the scheduled and delivery date into mm/dd/yyyy format.

b. Furthermore Month, year and date were split for both

c. And Finally the initial two columns ‘Scheduled date’ and ‘Delivery date’ were dropped.

```
[ ] tp['Scheduled Date']=pd.to_datetime(tp['Scheduled Date'],format='%m/%d/%y')
    tp['Delivery Date']=pd.to_datetime(tp['Delivery Date'],format='%m/%d/%y')
```

```
[ ] tp['S_month']=tp['Scheduled Date'].dt.month
    tp['S_year']=tp['Scheduled Date'].dt.year
    tp['S_Day']=tp['Scheduled Date'].dt.day
```

```
[ ] tp['D_month']=tp['Delivery Date'].dt.month
    tp['D_year']=tp['Delivery Date'].dt.year
    tp['D_Day']=tp['Delivery Date'].dt.day
```

```
[ ] tp.drop(['Scheduled Date','Delivery Date'],1,inplace=True)
```

5. The Data was then checked for any Null values and missing values

a. Missing values were filled in accordingly:

```
[ ] tp['Transport'].fillna("missingTransport",inplace=True)
    tp['Material'].fillna("missingMaterial",inplace=True)
    tp['Remote Location'].fillna("missingRemoteLocation",inplace=True)
```

```
[ ] tp['Artist Reputation'].fillna(value=tp['Artist Reputation'].mean(), inplace=True)
    tp['Height'].fillna(value=tp['Height'].mean(), inplace=True)
    tp['Weight'].fillna(value=tp['Weight'].mean(), inplace=True)
    tp['Width'].fillna(value=tp['Width'].mean(), inplace=True)
```

b. Null Values were identified using:

```
tp.isnull().sum()

Customer Id              0
Artist Name              0
Artist Reputation        0
Height                   0
Width                    0
Weight                   0
Material                 0
Price Of Sculpture       0
Base Shipping Price      0
International            0
Express Shipment         0
Installation Included    0
Transport                0
Fragile                  0
Customer Information     0
Remote Location          0
Customer Location        0
Cost                     0
S_month                  0
S_year                   0
S_Day                    0
D_month                  0
D_year                   0
D_Day                    0
dtype: int64
```

6. Some columns were dropped as they did not contribute to the goal of this exercise

a. Only the following variables were taken for further analysis - ['Height', 'Width', 'Weight','International', 'Express Shipment', 'Base Shipping Price','Installation Included','Transport', 'Fragile', 'Remote Location', 'S_month', 'S_year', 'S_Day', 'D_month','D_year', 'D_Day', 'Cost']]

7. Since the Variables ‘International’ and ‘Express Shipment’ contains Yes/ No values these were converted into True/False respectively (Boolean data)

```
[ ] tp=tp.replace({'Yes': True,'No':False,"missingRemoteLocation":False})
```

8. The Transport column was converted into a dummy variable.

a. Since this column is a categorical variable and initially comprised of these options Airways, Roadways, Waterways and Missing Values (NaN)

```
[ ] tp = pd.get_dummies(tp,prefix=['Transport'], columns = ['Transport'], drop_first=True)
```

```
[ ] tp.head()
```

Fragile	Remote Location	S_month	S_year	S_Day	D_month	D_year	D_Day	Cost	Transport_Roadways	Transport_Waterways	Transport_missingTransport
False	False	6	2015	7	6	2015	3	283.29	0	0	0
False	False	3	2017	6	3	2017	5	159.96	1	0	0
True	True	3	2015	9	3	2015	8	154.29	1	0	0
False	True	5	2015	24	5	2015	20	161.16	0	0	1
False	False	12	2016	18	12	2016	14	159.23	0	0	0

9. Pre-processing of data was completed once reached this:

```
tp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6500 entries, 0 to 6499
Data columns (total 19 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Height                      6500 non-null   float64
 1   Width                       6500 non-null   float64
 2   Weight                      6500 non-null   float64
 3   International               6500 non-null   bool
 4   Express Shipment            6500 non-null   bool
 5   Base Shipping Price         6500 non-null   float64
 6   Installation Included       6500 non-null   bool
 7   Fragile                     6500 non-null   bool
 8   Remote Location             6500 non-null   bool
 9   S_month                     6500 non-null   int64
 10  S_year                      6500 non-null   int64
 11  S_Day                       6500 non-null   int64
 12  D_month                     6500 non-null   int64
 13  D_year                      6500 non-null   int64
 14  D_Day                       6500 non-null   int64
 15  Cost                        6500 non-null   float64
 16  Transport_Roadways          6500 non-null   uint8
 17  Transport_Waterways         6500 non-null   uint8
 18  Transport_missingTransport  6500 non-null   uint8
dtypes: bool(5), float64(5), int64(6), uint8(3)
memory usage: 609.5 KB
```

10. We then built a Machine learning model to forecast demand.

a. This Machine learning model was trained on 80% of the data

```
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import mean_squared_log_error

features = tp.drop("Cost",axis=1)
labels = np.array(tp['Cost'])
train_features, test_features, train_labels, test_labels = train_test_split(features, labels, test_size = 0.2, random_state = 20)
```

11. We then tested the trained MLP regressor model on the 20% test data and got 89% accuracy

```python
from sklearn.neural_network import MLPRegressor
lr = MLPRegressor(random_state=1, max_iter=500000)

lr.fit(train_features, train_labels)

print("Analysis of Testing Dataset")
y_pred = lr.predict(test_features)
print('Accuracy {}'.format(lr.score(test_features, test_labels)))
print("This is the predicted model",y_pred)
print("This is the original trained labels",test_labels)
```

```
Analysis of Testing Dataset
Accuracy 0.8913213397253492
This is the predicted model [15199.67423321   274.45015445   370.49763504 ... 14918.10285468
 62439.86907958 15145.75852469]
This is the original trained labels [1.5766000e+02 3.2940000e+02 1.4346000e+02 ... 1.7370800e+03 3.2664808e+05
 1.7118000e+02]
```

Refences:

- https://artelogic.net/blog/the-role-of-logistics-in-a-supply-chain-management
- https://throughput.world/blog/industry/logistics-forecasting/