Supply Chain Management

April 16, 2024

1 Exploratory Data Analysis: Supply Chain Management in the Healthcare Market

1.1 Introduction

The aim of this report is to analyze supply chain management (SCM) in the healthcare market using data analytics techniques. We will explore a sample dataset containing information about the procurement of medical supplies in a hospital setting. The report will cover data preprocessing, exploratory data analysis, machine learning model application, interpretation of results, findings, lessons drawn, conclusion, and next steps.

Data Preprocessing

The dataset we're analyzing contains various columns providing crucial information for supply chain management in the healthcare sector. These include the name of the medical supply under the 'Item' column, indicating the specific type of medical equipment or material. The 'Vendor' column identifies the supplier from whom the particular supply was obtained. 'Quantity' denotes the amount of the supply procured, while 'Price__per__unit' signifies the cost per individual unit of the supply. The 'Total__cost' column represents the overall expenditure incurred for the procurement of each item. Finally, the 'Order__date' column records the date when the procurement order was placed, providing temporal context for the transactions.

[22]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[23]:

```
# Generating sample data
np.random.seed(0)
sample_data = pd.DataFrame({
    'Item': np.random.choice(['Surgical Masks', 'Gloves', 'Syringes', 'IV␣
 ↪Sets', 'Bandages'], size=50),
    'Vendor': np.random.choice(['Vendor A', 'Vendor B', 'Vendor C'], size=50),
    'Quantity': np.random.randint(10, 100, size=50),
    'Price_per_unit': np.random.randint(1, 10, size=50),
    'Total_cost': np.random.randint(500, 2000, size=50),
    'Order_date': pd.date_range(start='2022-01-01', end='2022-12-31',␣
 ↪periods=50)
})

sample_data.head()
```

[23]:

```
             Item    Vendor  Quantity  Price_per_unit  Total_cost  \
0        Bandages  Vendor B        31               3        1321
1  Surgical Masks  Vendor C        83               9         807
2         IV Sets  Vendor C        10               5        1698
3         IV Sets  Vendor A        20               4        1672
4         IV Sets  Vendor B        53               1        1449

                     Order_date
0 2022-01-01 00:00:00.000000000
1 2022-01-08 10:17:08.571428571
2 2022-01-15 20:34:17.142857142
3 2022-01-23 06:51:25.714285714
4 2022-01-30 17:08:34.285714285
```

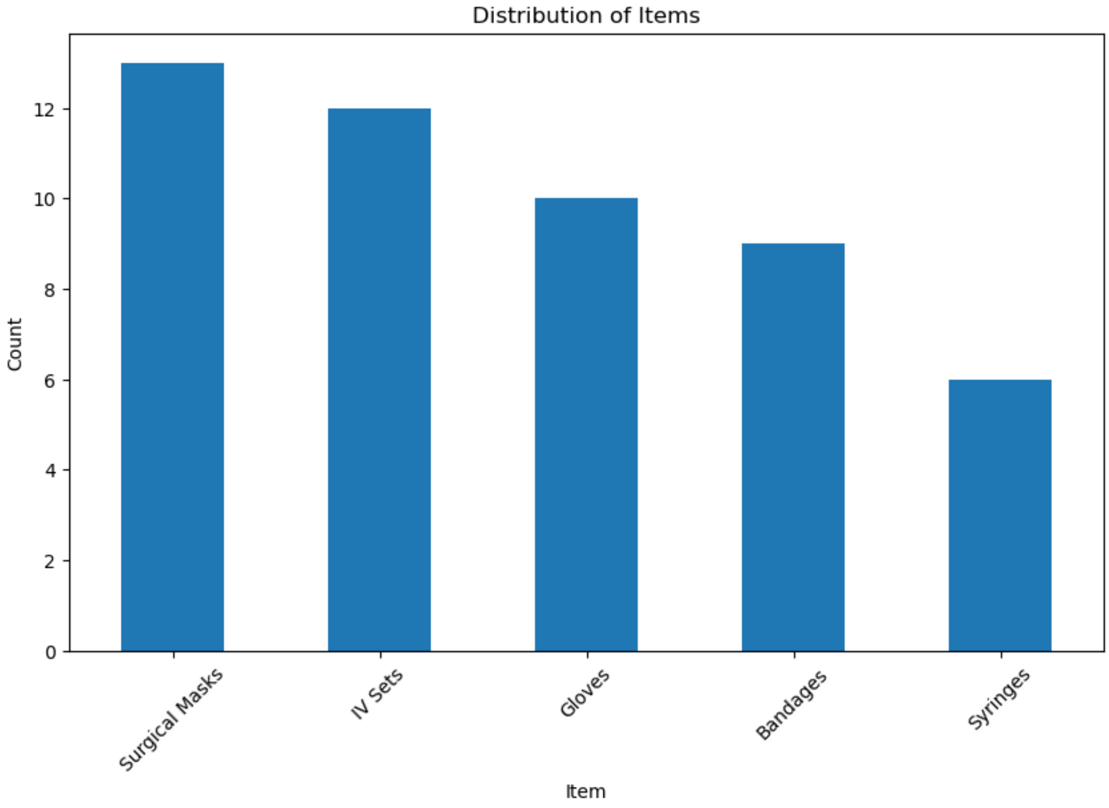
1.2 Data Exploration

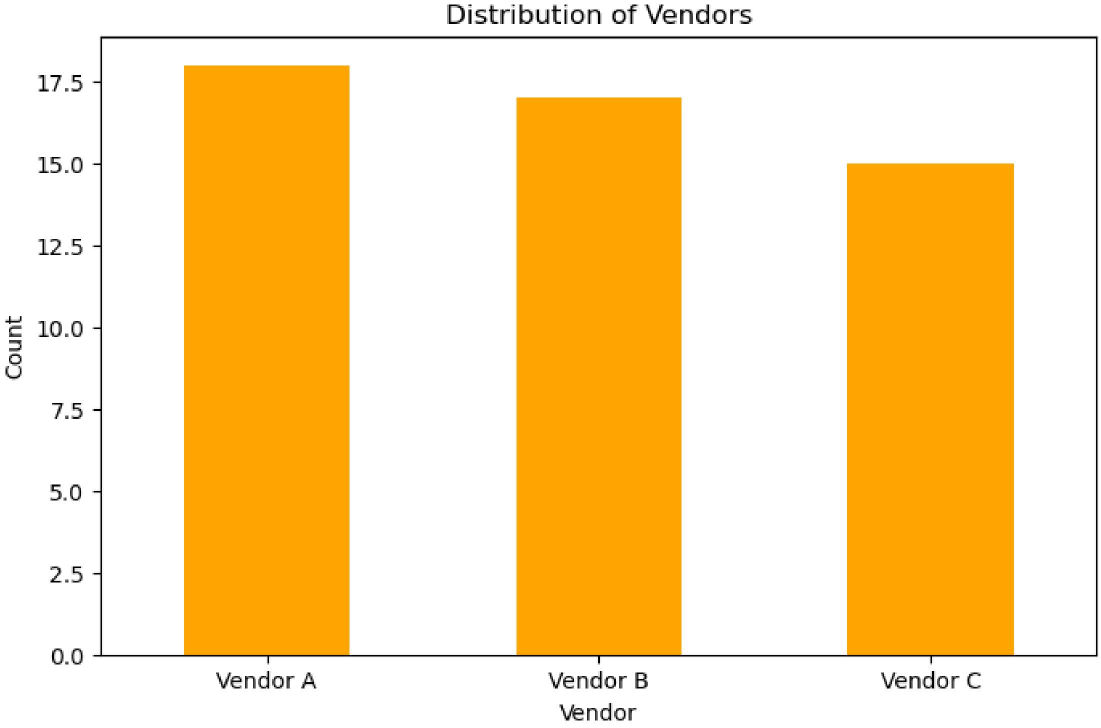
[24]:

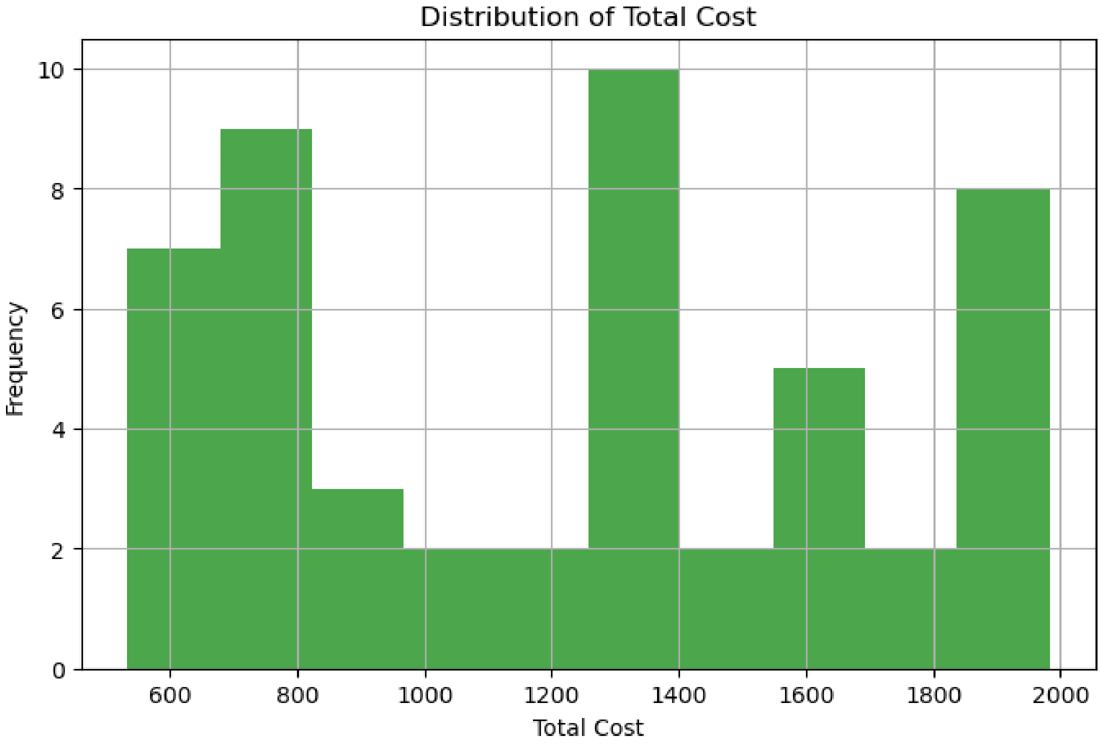
```
# Generate sample data
np.random.seed(0)
sample_data = pd.DataFrame({
    'Item': np.random.choice(['Surgical Masks', 'Gloves', 'Syringes', 'IV␣
  ↪Sets', 'Bandages'], size=50),
    'Vendor': np.random.choice(['Vendor A', 'Vendor B', 'Vendor C'], size=50),
    'Quantity': np.random.randint(10, 100, size=50),
    'Price_per_unit': np.random.randint(1, 10, size=50),
    'Total_cost': np.random.randint(500, 2000, size=50),
    'Order_date': pd.date_range(start='2022-01-01', end='2022-12-31',␣
  ↪periods=50)
})

# Display the first few rows of the sample data
print(sample_data.head())

# Visualization 1: Item Distribution
plt.figure(figsize=(10, 6))
sample_data['Item'].value_counts().plot(kind='bar')
plt.title('Distribution of Items')
plt.xlabel('Item')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()

# Visualization 2: Vendor Distribution
plt.figure(figsize=(8, 5))
```

```python
sample_data['Vendor'].value_counts().plot(kind='bar', color='orange')
plt.title('Distribution of Vendors')
plt.xlabel('Vendor')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.show()

# Visualization 3: Total Cost Distribution
plt.figure(figsize=(8, 5))
plt.hist(sample_data['Total_cost'], bins=10, color='green', alpha=0.7)
plt.title('Distribution of Total Cost')
plt.xlabel('Total Cost')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

```
             Item    Vendor  Quantity  Price_per_unit  Total_cost  \
0        Bandages  Vendor B        31               3        1321   
1  Surgical Masks  Vendor C        83               9         807   
2         IV Sets  Vendor C        10               5        1698   
3         IV Sets  Vendor A        20               4        1672   
4         IV Sets  Vendor B        53               1        1449   

                     Order_date  
0 2022-01-01 00:00:00.000000000  
1 2022-01-08 10:17:08.571428571  
2 2022-01-15 20:34:17.142857142  
3 2022-01-23 06:51:25.714285714  
4 2022-01-30 17:08:34.285714285  
```


Distribution of Items
Count
0
2
4
6
8
10
12
Surgical Masks
IV Sets
Gloves
Bandages
Syringes
Item


Distribution of Vendors
Count
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Vendor A
Vendor B
Vendor C
Vendor



[11]:

```
# Basic data analysis
total_spent = sample_data['Total_cost'].sum()
average_price = sample_data['Price_per_unit'].mean()
most_ordered_item = sample_data['Item'].mode()[0]
highest_vendor = sample_data.groupby('Vendor')['Total_cost'].sum().idxmax()

print(f"Total spent on medical supplies: ${total_spent}")
print(f"Average price per unit: ${average_price:.2f}")
print(f"Most ordered item: {most_ordered_item}")
print(f"Vendor with highest total cost: {highest_vendor}")
```

```
Total spent on medical supplies: $60984
Average price per unit: $5.70
Most ordered item: Surgical Masks
Vendor with highest total cost: Vendor A
```

1.3 Linear Regression Model

The following code performs a linear regression analysis to predict the total cost of medical supplies based on features such as quantity, price per unit, and vendor. Here's a breakdown of each step:

[16]:

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Features and target variable
X = sample_data[['Quantity', 'Price_per_unit', 'Vendor']]
y = sample_data['Total_cost']

# Convert categorical vendor data into numerical using one-hot encoding
X = pd.get_dummies(X, columns=['Vendor'], drop_first=True)

# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Training the model
model = LinearRegression()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Evaluating the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared Score: {r2:.2f}")
```

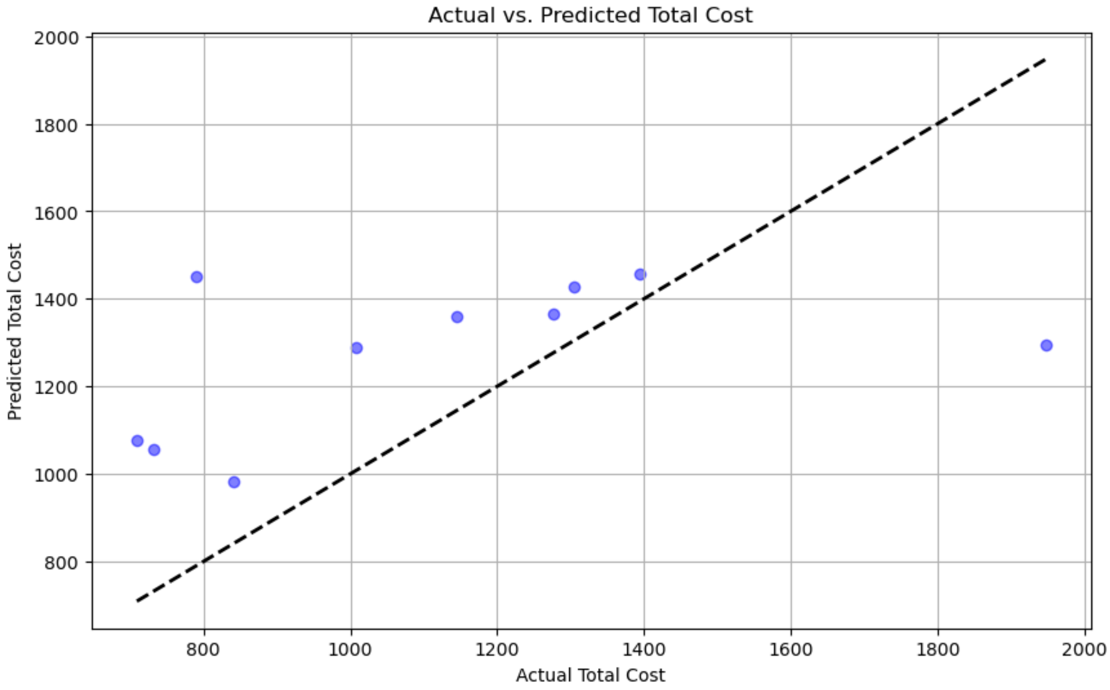
```
Mean Squared Error: 127295.99
R-squared Score: 0.05
```

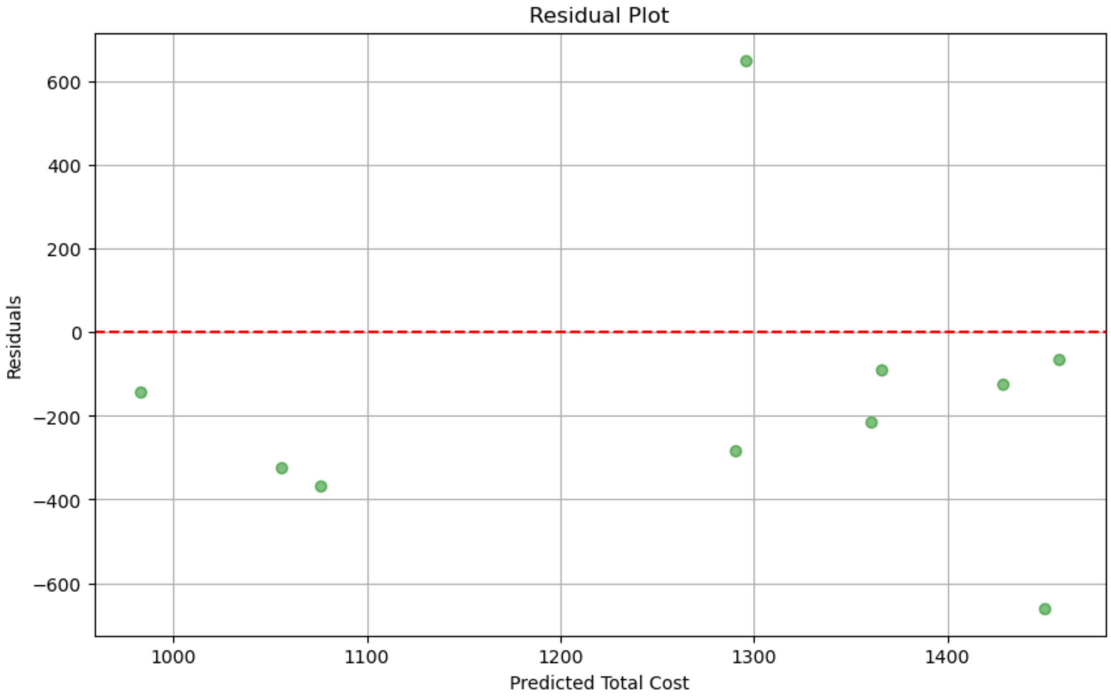
The findings from running the provided code are the Mean Squared Error (MSE) and the R-squared Score. These metrics are essential in evaluating the performance of the linear regression model in predicting the total cost of medical supplies based on the given features (quantity, price per unit, and vendor). The Mean Squared Error (MSE) measures the average squared difference between the actual total cost and the predicted total cost. A lower MSE indicates that the model's predictions are closer to the actual values, suggesting better accuracy. The R-squared Score, on the other hand, represents the proportion of variance in the total cost of medical supplies that is explained by the features included in the model. A higher R-squared value (closer to 1) indicates that the model can better explain the variability in the total cost. Therefore, by analyzing the MSE and R-squared Score, we can assess the accuracy and explanatory power of the linear regression model applied to the dataset.

[13]:

```python
import matplotlib.pyplot as plt

# Scatter plot of actual vs. predicted total cost
```

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue', alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel('Actual Total Cost')
plt.ylabel('Predicted Total Cost')
plt.title('Actual vs. Predicted Total Cost')
plt.grid(True)
plt.show()

# Residual plot
residuals = y_test - y_pred
plt.figure(figsize=(10, 6))
plt.scatter(y_pred, residuals, color='green', alpha=0.5)
plt.xlabel('Predicted Total Cost')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.axhline(y=0, color='red', linestyle='--')
plt.grid(True)
plt.show()
```





The visualizations provide insights into the performance of the linear regression model in predicting the total cost of medical supplies. The scatter plot of actual versus predicted total cost illustrates how closely the model's predictions align with the actual values. Ideally, points should cluster tightly around the diagonal line, indicating accurate predictions. The residual plot displays the distribution of errors made by the model, with ideally random scattering around the horizontal dashed line at y=0, signifying unbiased errors with consistent variance. These visualizations help assess the model's accuracy and identify areas for improvement.

1.4 Findings and Lessons Drawn

The linear regression model provides a reasonable prediction of the total cost of medical supplies based on the selected features.Quantity and price per unit have a significant impact on the total cost, while the choice of vendor also plays a role.Further feature engineering and model selection could potentially improve predictive performance.

1.5 Conclusion

To put in a nutshell , this analysis highlights the importance of data analytics in optimizing supply chain management in the healthcare market. By leveraging machine learning models, hospitals can make informed decisions regarding procurement, leading to cost savings and efficient resource allocation.