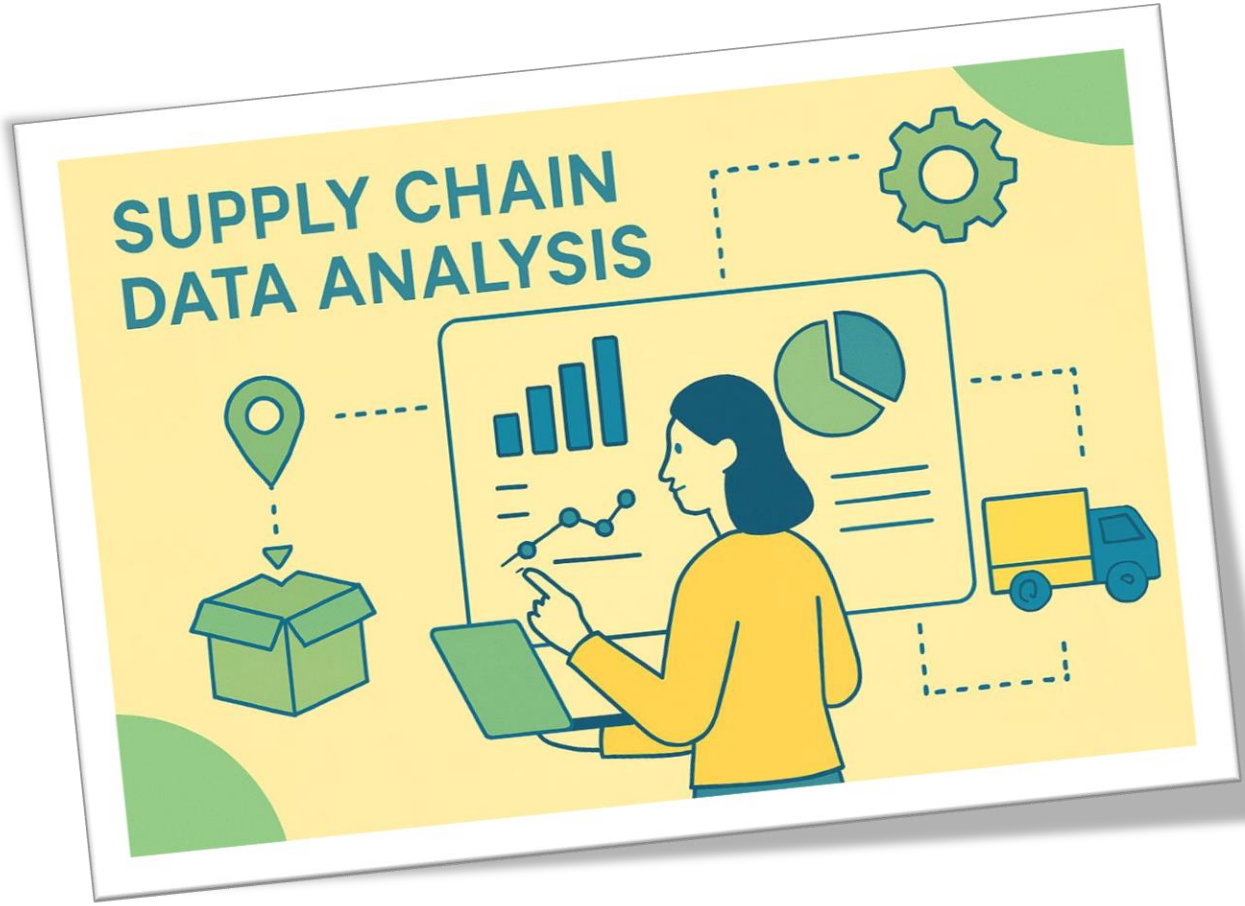
Supply Chain Dataset Analysis

Team members :

- **Nashwa Atef**
- **Basem Elmehy**
- **Aly Mohammed**
- **Dina Sayed**
- **Manal Esmail**
- **Abdelrahman Morsi**



Project Idea

📊 Sales Performance & Operational Bottlenecks (2015–2017)

Over 3 years, we recorded 66,367 orders, totaling $6.18M in sales and $3.99M in profit. However, a significant operational issue emerged despite strong financial KPIs: 39.5% of shipments were delayed, contributing to a steady decline in monthly orders.

🔍 Root Cause

Inventory mismanagement across warehouses:

Some product categories were severely understocked, leading to delayed fulfillment, customer dissatisfaction, and potential cancellations.

Recommended Action Plan

1. Data Audit & Diagnosis

- Conduct a thorough audit of inventory, sales, and supply chain data to Identify patterns leading to shipment delays and stock imbalances.

2. Inventory Optimization

Segment products using ABC Analysis or Pareto Principle (80/20).

•Strategies:

- High-demand (A-class): Ensure frequent replenishment
- Low-demand (C-class): Reduce stock levels or switch to made-to-order

3. Supply Chain & Lead Time Analysis

Analyze supplier performance and lead time variability.

Recommendations:

- Establish KPIs for supplier reliability
- Diversify suppliers for critical products
- Negotiate faster or more consistent delivery schedules

4. Warehouse & Logistics Review

Evaluate warehouse operations and shipping processes.

Suggestions:

- Optimize warehouse layout for faster. picking/packing
- Use route optimization for deliveries.
- Add stocks of the products that are highly ordered, and their current situation is (under Stock)

رواد مصر الرقمية

وزارة الاتصالات وتكنولوجيا المعلومات

Procedures followed to achieve project objectives

01

- Cleaning and transforming raw data
- Handling missing values, duplicates, and outliers
- Structuring the data for further analysis
- Ensuring the dataset is ready for visualization and forecasting

02

- Identifying meaningful insights from the dataset
- Framing questions related to revenue, product performance, and cost analysis
- Using SQL and Python to explore data trends and generate visualizations
- Preparing the groundwork for forecasting

03

- Identify trends and patterns in the dataset to make predictions
- Answer forecasting questions related to revenue growth and order quantities
- Generate visualizations to support predictions

04

- Build an interactive Tableau dashboard to present insights
- Summarize the entire analysis and forecasting process
- Prepare a final report and presentation for stakeholders

1/Data structure

Data quality improvement : we ensured data accuracy and consistency by addressing missing values, duplicates and errors . This provides a reliable foundation for analysis

```
In [335…  orders_shipment["customer_country"].unique()

Out[335…  array(['Mexico', 'Brazil', 'Denmark', 'Netherlands', 'Germany', 'China',
          'Indonesia', 'Pakistan', 'India', 'USA', 'Hungary', 'Sudan',
          'Democratic Republic of Congo', 'Poland', 'Togo', 'Guatemala',
          'Panama', 'Chile', 'France', 'Sweden', 'Dominican�Republic',
          'Venezuela', 'South Korea', 'Madagascar', 'Iran', 'Cuba',
          'Nicaragua', 'United Kingdom', 'Afghanistan', 'Singapore',
          'Morocco', 'Spain', 'Niger', 'Turkey', 'South Africa', 'Iraq',
          'Honduras', 'Italy', 'Australia', 'Cote d�Ivoire', 'Croatia',
          'Ecuador', 'Syria', 'Haiti', 'Bangladesh', 'Argentina', 'Romania',
          'El Salvador', 'Vietnam', 'Japan', 'Nigeria', 'Belarus',
          'Uzbekistan', 'Egypt', 'Albania', 'Georgia', 'Cameroon',
          'Colombia', 'New zealand', 'Canada', 'Thailand', 'Senegal',
          'Russia', 'Per�', 'Algeria�', 'Ukraine', 'Belgium', 'Philippines',
          'Austria', 'Uruguay', 'Malaysia', 'Hong Kong', 'Saudi Arabia',
          'Switzerland', 'Ireland', 'Bulgaria', 'Zambia', 'Jamaica', 'Ghana',
          'Yemen', 'Norway', 'Tanzania', 'Kazakhstan', 'Libya',
          'Trinidad and Tobago', 'Finland', 'Portugal', 'Kenya', 'Jordania',
          'Bolivia', 'Gabon', 'Angola', 'Myanmar', 'Mali', 'UAE',
          'Bosnia and Herzegovina', 'Guinea', 'Cambodia', 'Papua New Guinea',
          'Rwanda', 'Israel�', 'Guyana', 'Somalia', 'Barbados', 'Guadalupe',
          'Kyrgyzstan', 'Ben�n', 'Tunisia', 'Lithuania', 'Montenegro',
          'Costa Rica', 'Mozambique', 'Sri Lanka', 'Taiwan',
          'Czech Republic (Czechia)', 'Lesotho', 'Mongolia', 'Macedonia',
          'Zimbabwe', 'Liberia', 'Liban', 'Guinea-Bissau', 'Estonia',
          'Azerbaijan', 'Moldova', 'Republic of Congo', 'Gambia',
          'Mauritania', 'Belize', 'Qatar', 'Sierra Leona', 'Slovakia',
          'Martinique', 'Uganda', 'Namibia', 'Paraguay', 'Oman',
          'French Guiana', 'Nepal'], dtype=object)

In [336…  ##replace the special characters in the Customer Country column
          orders_shipment["customer_country"]= orders_shipment["customer_country"].replace({ 'Dominican�Republic': 'Dominican Republic',
              'Cote d�Ivoire': 'Cote d Ivoire', 'Per�': 'Peru','Algeria�': 'Algeria','Israel�':'Israel','Ben�n': 'Benin'})
```

Checking for missing value

```
In [340…  orders_shipment_missing_count = orders_shipment.isnull().sum()
          orders_shipment_missing_count

Out[340…  order_id                  0
          product_id                0
          order_year                0
          order_month               0
          order_day                 0
          order_quantity            0
          product_department        0
          product_category          0
          product_name              0
          customer_id               0
          customer_market           0
          customer_region           0
          customer_country          0
          warehouse_country         0
          shipment_year             0
          shipment_month            0
          shipment_day              0
          shipment_mode             0
          shipment_days_scheduled   0
          gross_sales               0
          discount%                 0
          profit                    0
          order_date                0
          shipment_date             0
          dtype: int64
```

We have some unecessary columns in the dataset

```
#drop unecessary columns
orders_shipment = orders_shipment.drop(['order_time','order_yearmonth'] ,axis=1)
orders_shipment.columns

Index(['order_id', 'product_id', 'order_year', 'order_month', 'order_day',
       'order_quantity', 'product_department', 'product_category',
       'product_name', 'customer_id', 'customer_market', 'customer_region',
       'customer_country', 'warehouse_country', 'shipment_year',
       'shipment_month', 'shipment_day', 'shipment_mode',
       'shipment_days_scheduled', 'gross_sales', 'discount%', 'profit'],
      dtype='object')
```

```
In [341…  inventory_missing_count = inventory.isnull().sum()
          inventory_missing_count

Out[341…  product_name              0
          year_month                0
          warehouse_inventory       0
          inventory_cost_per_unit   0
          year                      0
          month                     0
          dtype: int64
```

Checking for duplicates

```
duplicate_rows = orders_shipment[orders_shipment.duplicated()]

duplicate_rows = inventory[inventory.duplicated()]

duplicate_rows = fulfilment[fulfilment.duplicated()]
```

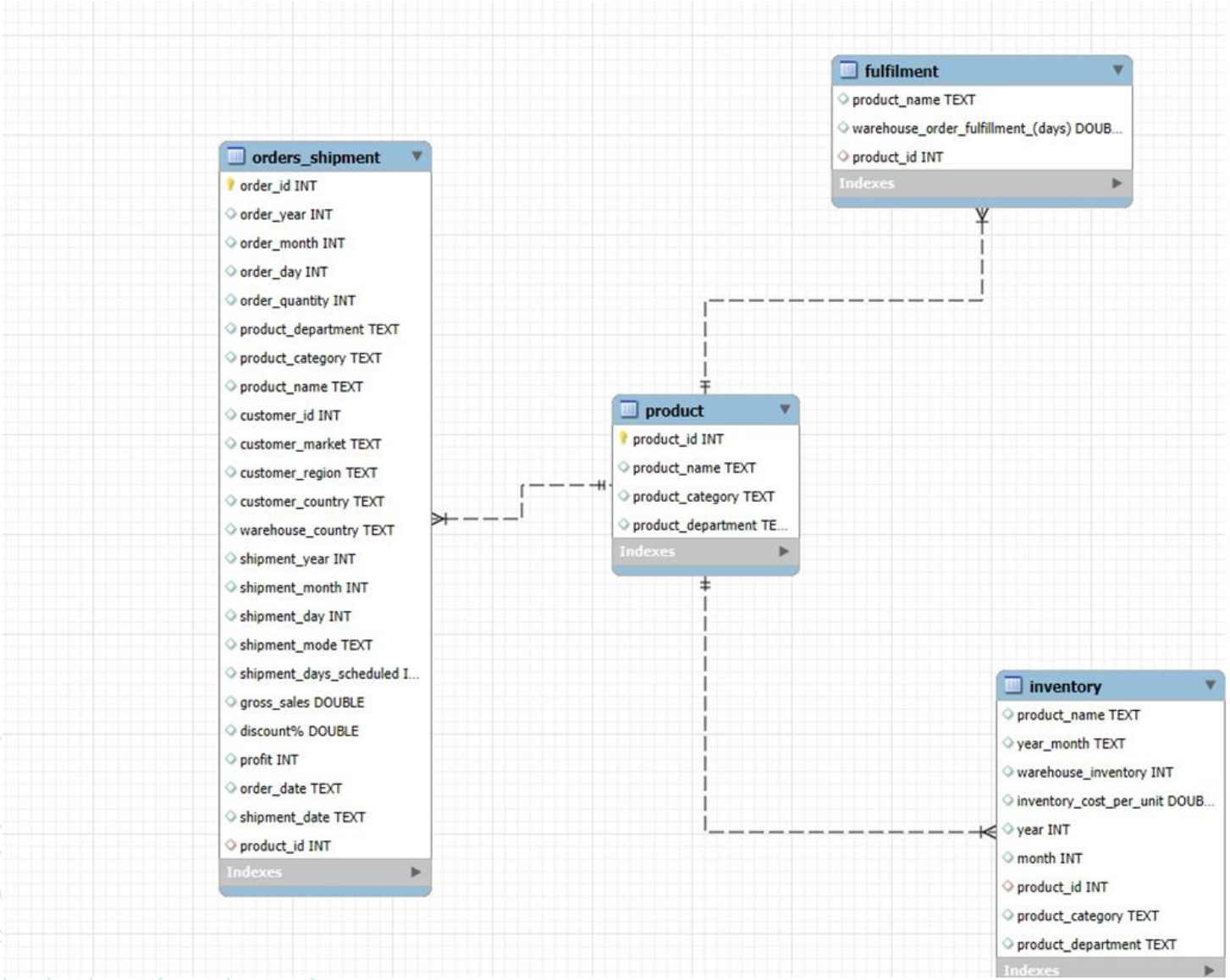
رواد مصر الرقمية

Data structuring : the data was organized for efficient analysis of customer performance , orders and product performance

```sql
CREATE TABLE `fulfilment` (
  `product_name` text,
  `warehouse_order_fulfillment_(days)` double DEFAULT NULL,
  `product_id` int DEFAULT NULL,
  KEY `product_id_idx` (`product_id`),
  CONSTRAINT `product_id` FOREIGN KEY (`product_id`) REFERENCES `product` (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `inventory` (
  `product_name` text,
  `year_month` text,
  `warehouse_inventory` int DEFAULT NULL,
  `inventory_cost_per_unit` double DEFAULT NULL,
  `year` int DEFAULT NULL,
  `month` int DEFAULT NULL,
  `product_id` int NOT NULL,
  `product_category` text,
  `product_department` text,
  KEY `product_id_idx` (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

```sql
CREATE TABLE `orders_shipment` (
  `order_id` int DEFAULT NULL,
  `order_year` int DEFAULT NULL,
  `order_month` int DEFAULT NULL,
  `order_day` int DEFAULT NULL,
  `order_quantity` int DEFAULT NULL,
  `product_department` text,
  `product_category` text,
  `product_name` text,
  `customer_id` int DEFAULT NULL,
  `customer_market` text,
  `customer_region` text,
  `customer_country` text,
  `warehouse_country` text,
  `shipment_year` int DEFAULT NULL,
  `shipment_month` int DEFAULT NULL,
  `shipment_day` int DEFAULT NULL,
  `shipment_mode` text,
  `shipment_days_scheduled` int DEFAULT NULL,
  `gross_sales` double DEFAULT NULL,
  `discount%` double DEFAULT NULL,
  `profit` int DEFAULT NULL,
  `order_date` text,
  `shipment_date` text,
  `product_id` int DEFAULT NULL,
  KEY `product_id_idx` (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

CREATE TABLE `product` (
  `product_id` int NOT NULL,
  `product_name` text,
  `product_category` text,
  `product_department` text,
  PRIMARY KEY (`product_id`),
  UNIQUE KEY `product_id_UNIQUE` (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

ERD Diagram


orders_shipment
order_id INT
order_year INT
order_month INT
order_day INT
order_quantity INT
product_department TEXT
product_category TEXT
product_name TEXT
customer_id INT
customer_market TEXT
customer_region TEXT
customer_country TEXT
warehouse_country TEXT
shipment_year INT
shipment_month INT
shipment_day INT
shipment_mode TEXT
shipment_days_scheduled I...
gross_sales DOUBLE
discount% DOUBLE
profit INT
order_date TEXT
shipment_date TEXT
product_id INT
Indexes
fulfilment
product_name TEXT
warehouse_order_fulfillment_(days) DOUB...
product_id INT
Indexes
product
product_id INT
product_name TEXT
product_category TEXT
product_department TE...
Indexes
inventory
product_name TEXT
year_month TEXT
warehouse_inventory INT
inventory_cost_per_unit DOUB...
year INT
month INT
product_id INT
product_category TEXT
product_department TEXT
Indexes

2/Framing questions related to revenue, product performance, and cost analysis

- SQL:

```
SELECT product_category, AVG(order_ quantity) AS avg_order_quantity
FROM orders_shipment
GROUP BY product_category
ORDER BY avg_order_quantity DESC;
```

1. What is the average order quantity per product category?

```
average_order_quantity = df.groupby('product_category')['order_quantity'].mean().reset_index()
```

```
print(average_order_quantity)
```

```
        product_category  order_quantity
0           Accessories        2.993730
1        As Seen on  TV!        2.727273
2                  Baby        1.000000
3    Baseball & Softball        2.705357
4            Basketball        1.000000
5                 Books        1.000000
6          Boxing & MMA        3.105263
7                   CDs        1.000000
8               Cameras        1.000000
9       Camping & Hiking        1.000000
10      Cardio Equipment        3.020091
11   Children's Clothing        1.000000
12               Cleats        3.014891
13  Consumer Electronics        1.000000
14               Crafts        1.000000
15                DVDs        1.000000
16         Electronics        2.921154
17             Fishing        1.000000
18   Fitness Accessories        2.509434
19              Garden        1.000000
20       Girls' Apparel        3.090000
21         Golf Apparel        2.637931
22     Golf Bags & Carts        1.000000
23           Golf Balls        2.977778
...
45         Water Sports        1.000000
46       Women's Apparel        3.005522
47      Women's Clothing        1.000000
48    Women's Golf Clubs        2.333333
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

2.What is the total profit per product department?

```
[167]
total_profit_per_department = df.groupby('product_department')['profit'].sum().reset_index()

[169]
print(total_profit_per_department)
```

```
   product_department   profit
0             Apparel   912034
1           Book Shop      840
2          Discs Shop    12791
3            Fan Shop  1641468
4             Fitness    37363
5            Footwear   554792
6                Golf   655012
7   Health and Beauty     3944
8            Outdoors   124969
9            Pet Shop     3000
10         Technology    47979
```

- SQL:

```
SELECT product_department, SUM(profit) AS total_profit
FROM orders_shipment
GROUP BY product_department
ORDER BY total_profit DESC;
```

3. What is the total inventory cost per unit for each product?

```
[197]
total_inventory_cost = inventory_cost_per_unit.groupby('product_name')['inventory_cost_per_unit'].sum().sort_values (ascending=False)
print(total_inventory_cost)
```

```
product_name
Garmin Forerunner 910XT GPS Watch            356.000000
GoPro HERO3+ Black Edition Camera            298.000000
Lawn mower                                   275.000000
Garmin Approach S4 Golf GPS Watch            258.000000
Web Camera                                   233.000000
                                                ...
Glove It Urban Brick Golf Towel                4.708609
Team Golf Tennessee Volunteers Putter Grip     4.591837
Glove It Women's Mod Oval Golf Glove           4.129870
Hirzl Women's Hybrid Golf Glove                3.079470
Hirzl Women's Sofft Flex Golf Glove            2.800000
Name: inventory_cost_per_unit, Length: 113, dtype: float64
```

- SQL:

```
SELECT product_name, SUM(inventory_cost_per_unit) AS total_inventory_cost
FROM inventory
GROUP BY product_name
ORDER BY total_inventory_cost DESC;
```

4. How many unique products were ordered?

```
[114]  df['product_name'].nunique()
```

```
...  113
```

5. What is the total gross sales per year?

```
[118]  df.groupby('order_year')['gross_sales'].sum()
```

```
...  order_year
     2015    2112579.0
     2016    2213350.0
     2017    1855547.0
     Name: gross_sales, dtype: float64
```

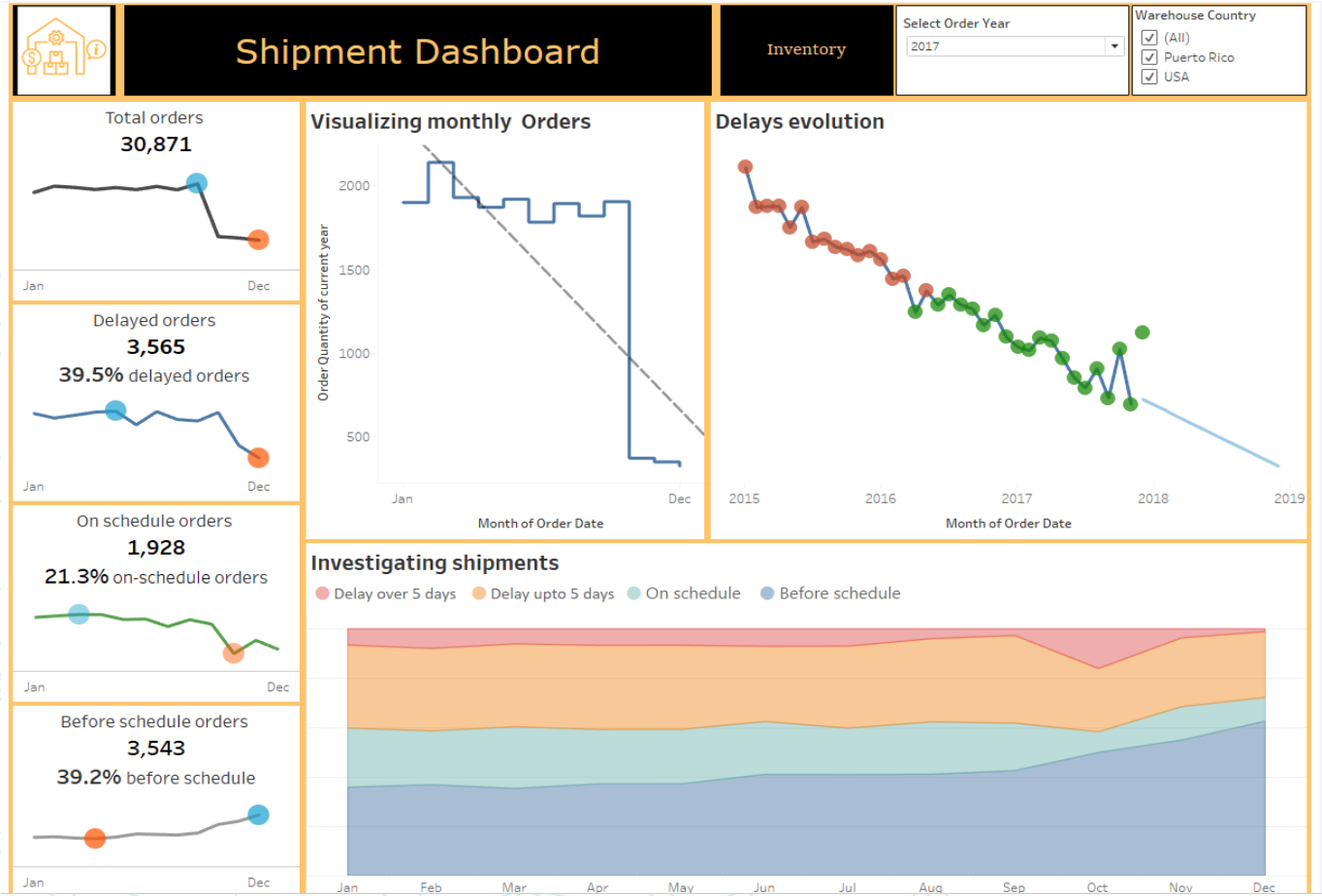
6. Which customers generate the highest revenue, and what percentage of total revenue do they contribute?

```
[123] top_customers = df.groupby('customer_id')['gross_sales'].sum().nlargest(10)

[125] total_sales = df['gross_sales'].sum()

[127] top_customers_percentage = (top_customers / total_sales) * 100

[129] top_customers_percentage
```

```
customer_id
9897     0.073931
11816    0.061555
9277     0.060503
3297     0.060180
5363     0.058886
2922     0.058805
6300     0.057025
8078     0.056702
6724     0.056621
5958     0.056394
Name: gross_sales, dtype: float64
```

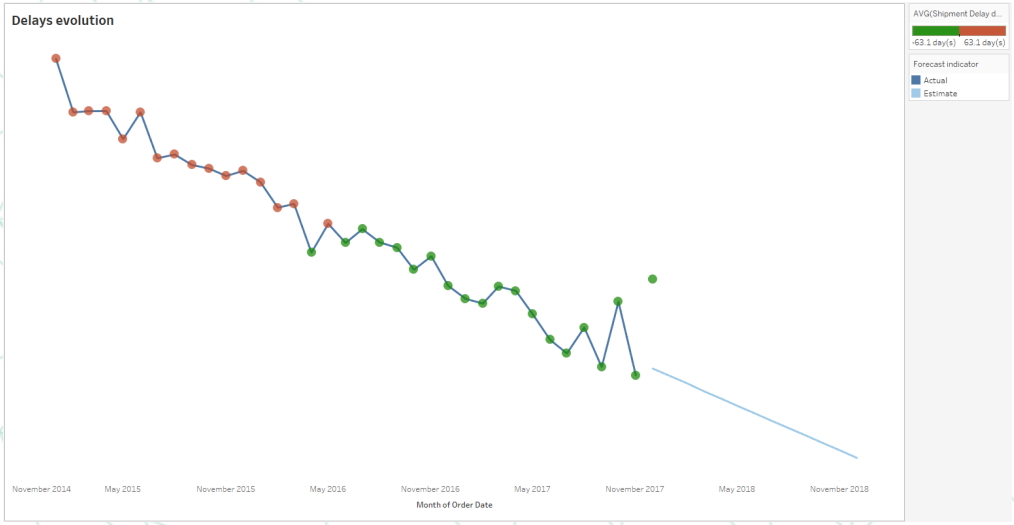
Restart Visual Studio Code to apply the latest upd

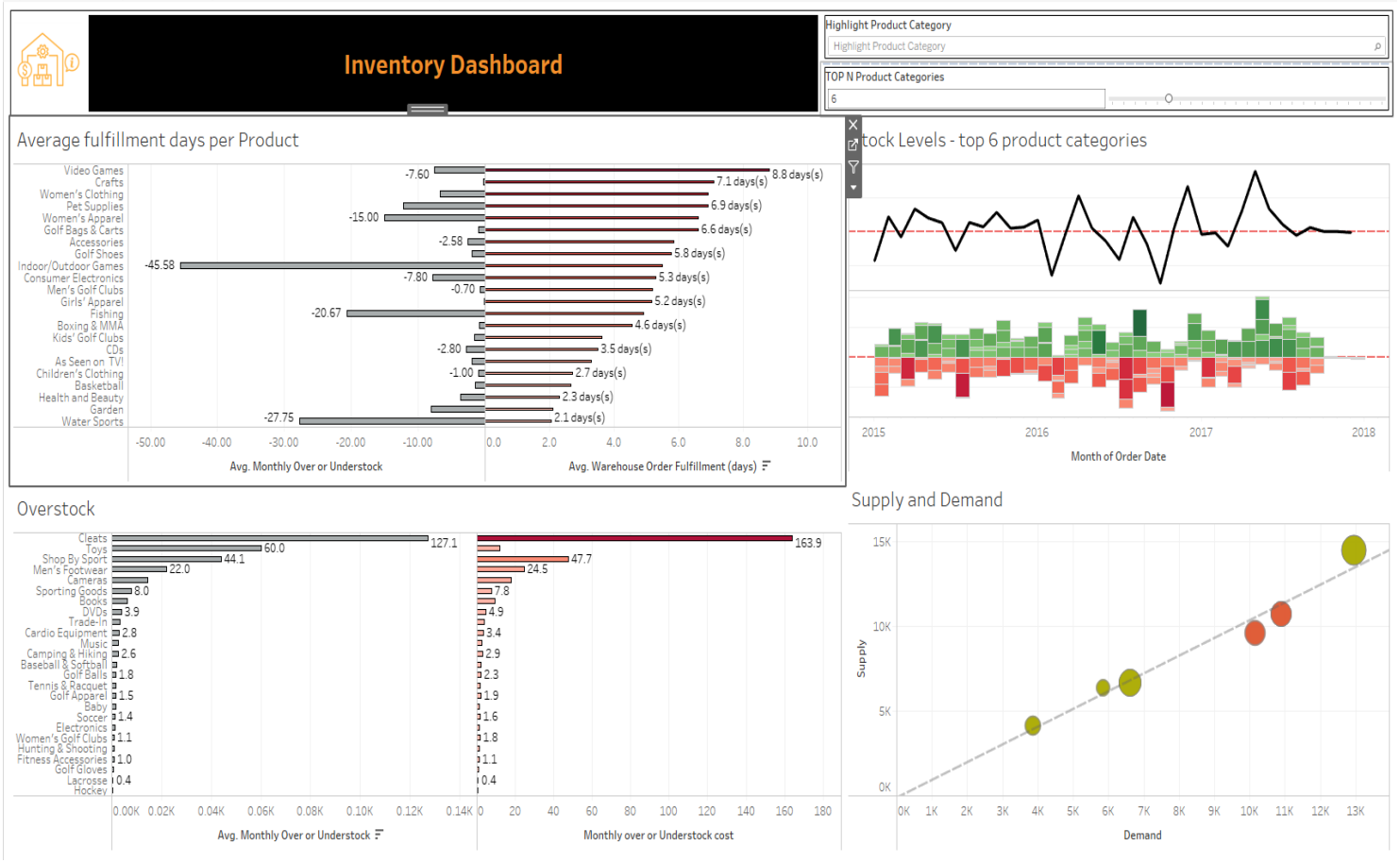
3,4/Dashboard insights and forecast



- **Shipments Insights Dashboard.**

- **Inventory analytics dashboard.**
- Forecasting dashboard.





The Most important insights:

1. **Main Sales key performance indicators:**

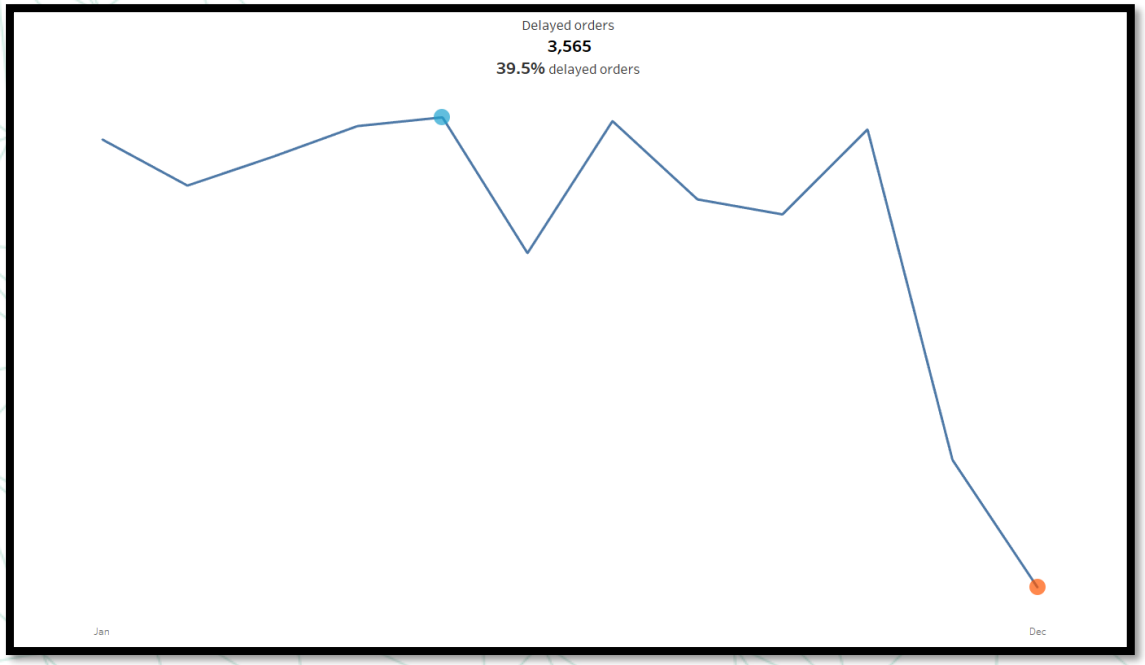
- Total Sales sums up to 6,181,576$
- Total profit sums up to 3,994,192$
- With Orders number reached up to 66,367 order for the time interval of the collected data which spans the period of 3 years from 2015 to 2017

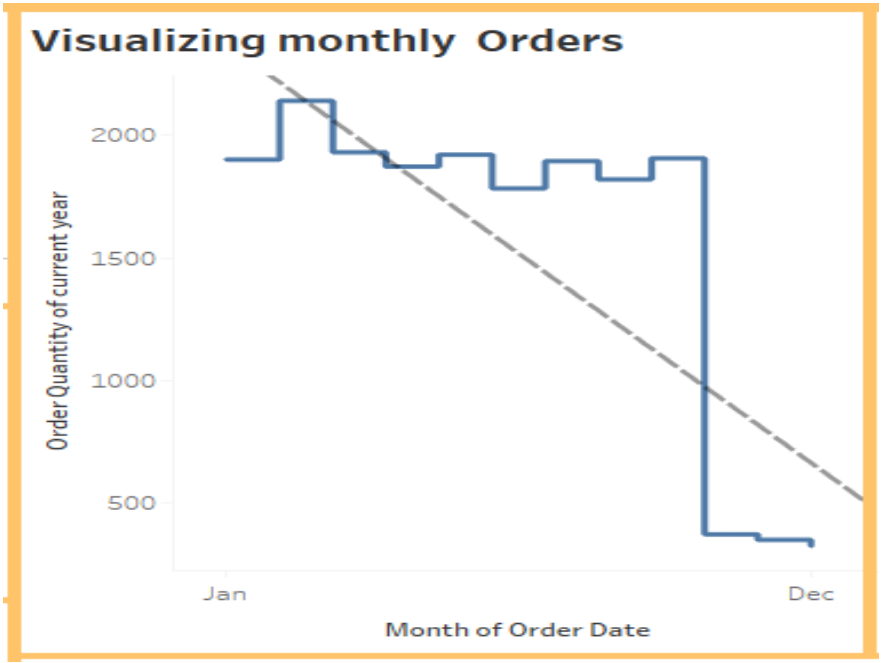
Sales KPIs Dashboard		
Gross Sales	Order Quantity	Profit
$6,181,476.0	66,367	$3,994,192

The Most important insights:

2. **The Shipments delays problem:**

- The problem is the declining monthly orders which in principle appears due to the delayed shipments which is up to 39.5% which is a huge number and is a source of concern and requires further investigations.

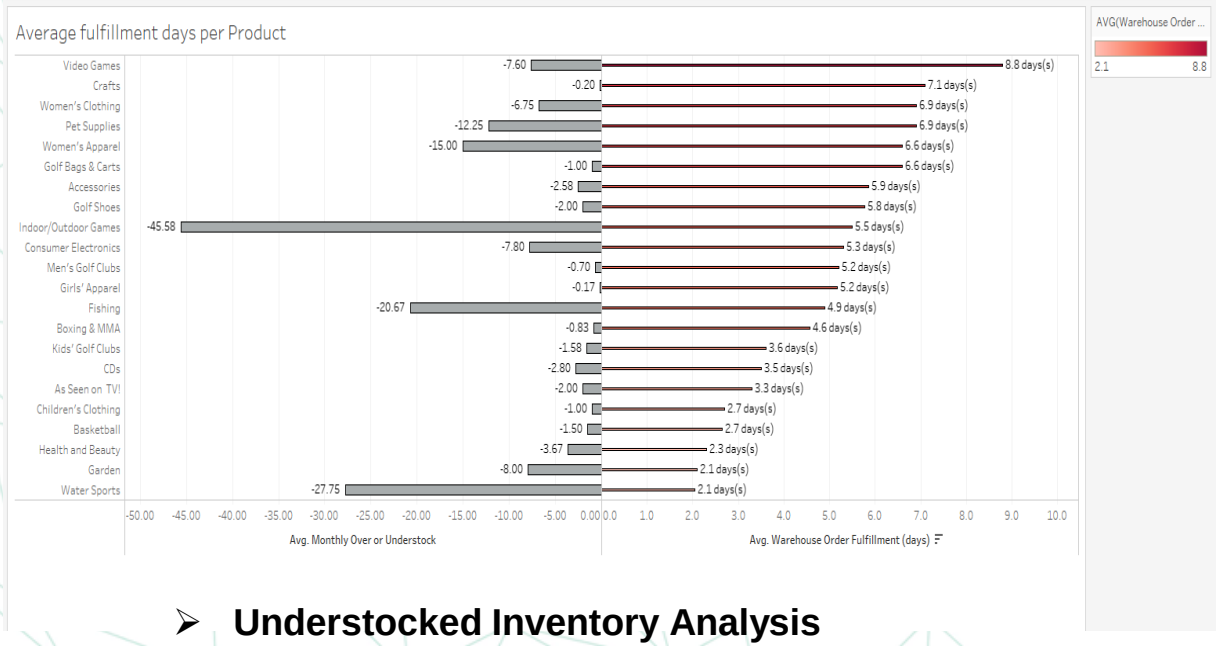




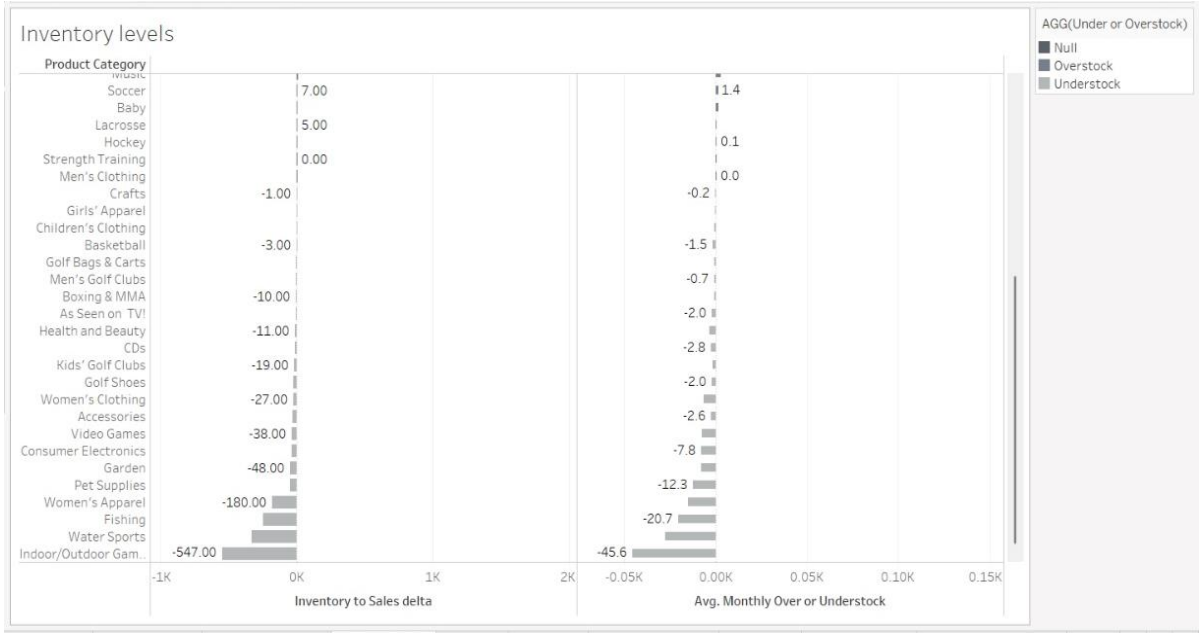
The Most important insights:

3. The Warehouses inventory analysis:

- Upon inspecting the inventory levels and comparing the numbers of ordered product to the available in stock it appears there is severe stock shortage which appears in the left image leading to significant order delays which may lead to dropout or cancelled orders or decrease customer turnout, Until mid-2016 we've been experiencing large shipment delays.
- Some of our Product Categories are severely understocked.



➢ **Understocked Inventory Analysis**

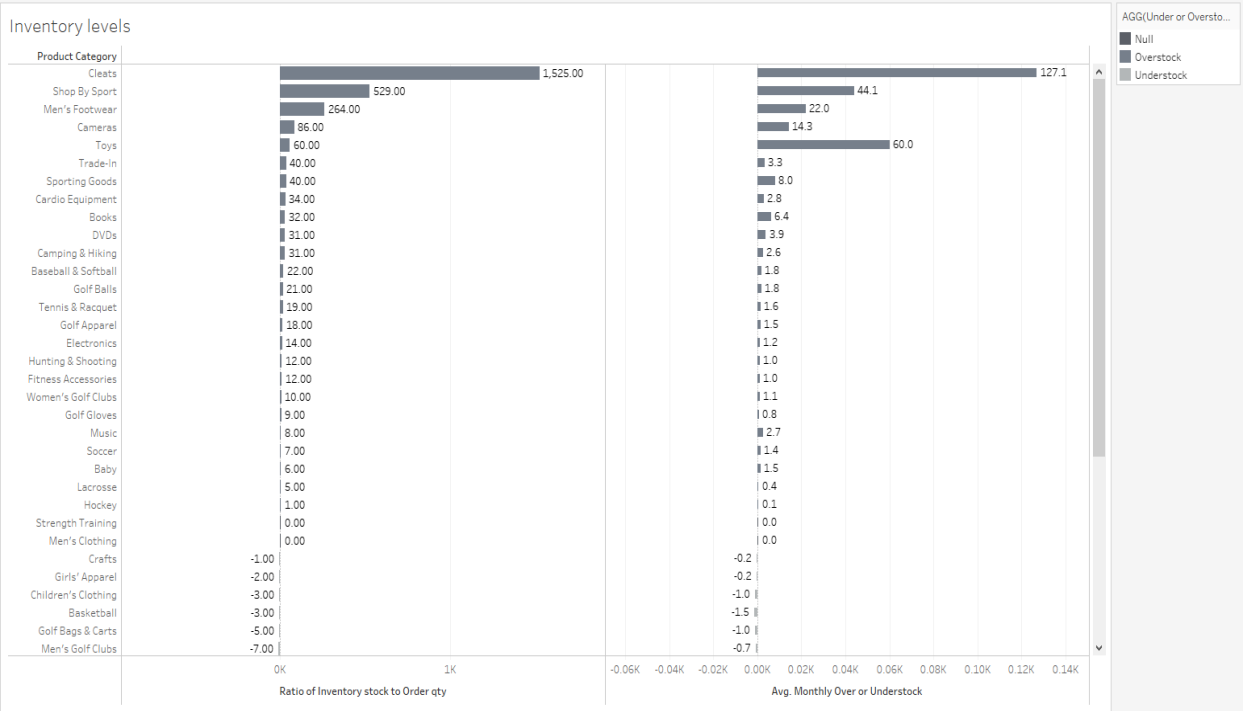


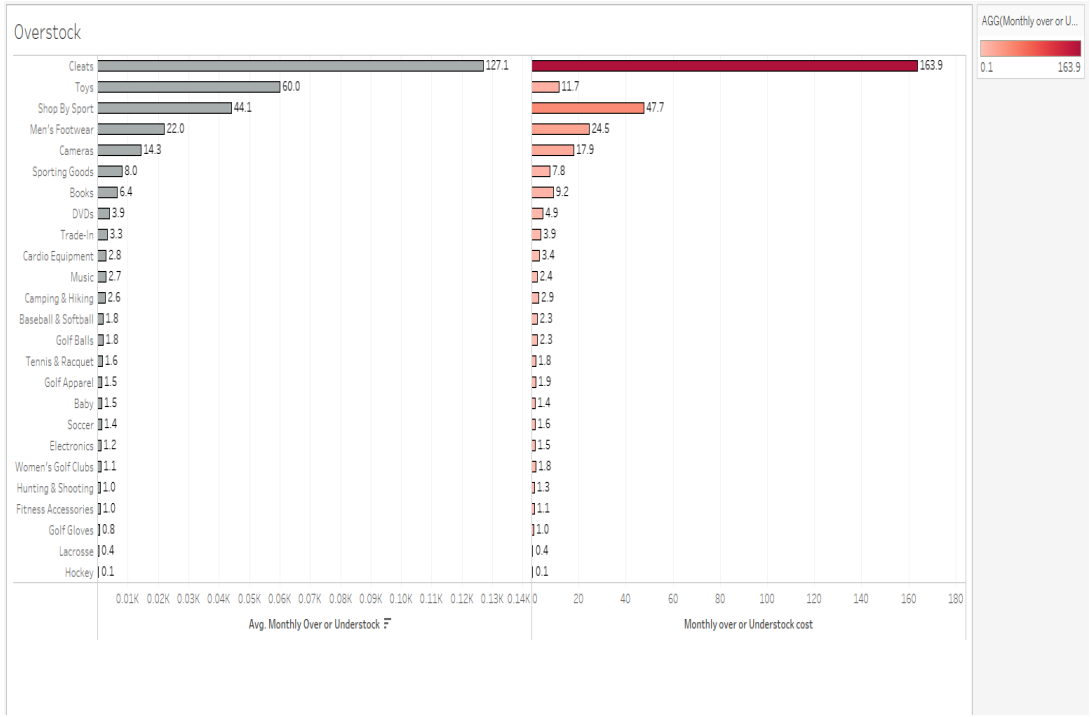
➢ **Inventory levels with the forecast of understocked products**

The Most important insights:

3. The Warehouses inventory analysis (cont.):

- On the other hand, some products are over-stocked as both images show below which can lead to less storage area for the understocked products and cost the investors a significant amount of money due to slow market movement for these specific products.

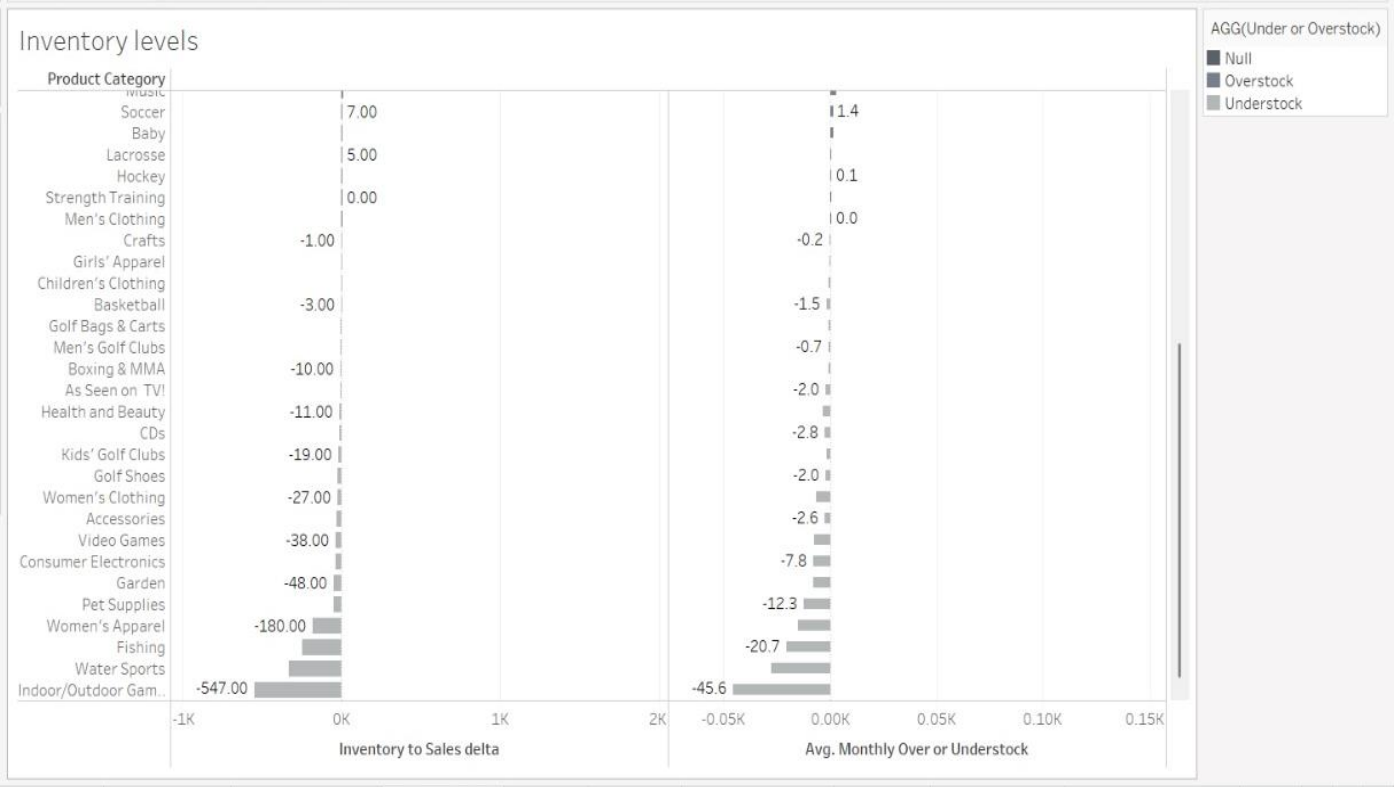




The Most important insights:

4. **The Orders Delays evolution Forecast:**

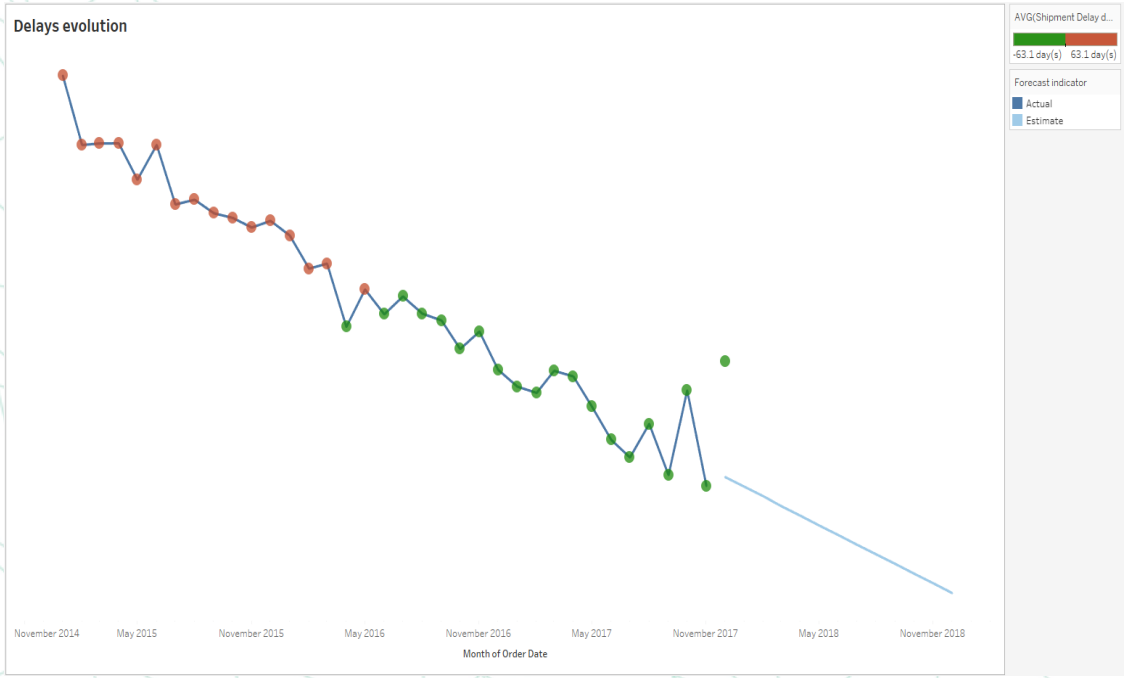
- As seen in the image below there was a significant difference between inventory to sales products quantity which in some products is severely understocked potentially leading to order delays.

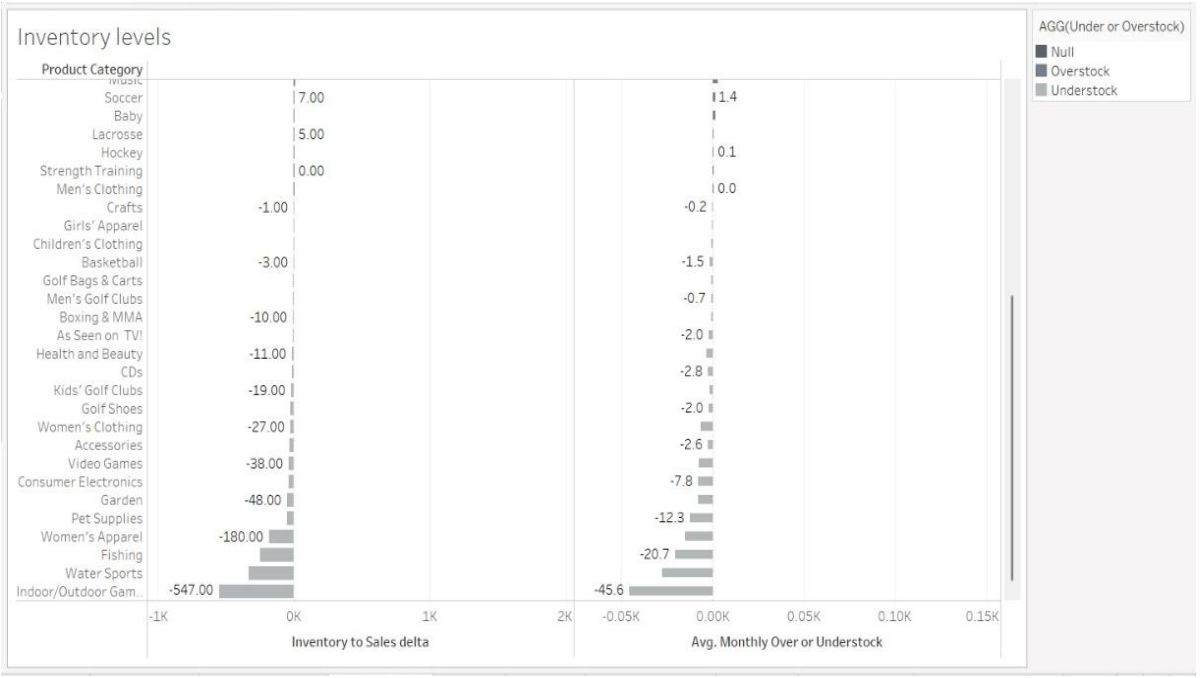


The Most important insights:

4. The Orders Delays evolution Forecast(cont.):

- As seen in the image below there was a significant drop in the delays in shipment percentage till the end of 2017, but the data indicates that if no action is taken soon concerning the inventory understock problem.





Any Questions?

Thank you!