**Superstore Demand Forecasting & Inventory Policy**

**1. Introduction & Goals**

In this project, I wanted to predict how many units a fictional “Superstore” sells each day, then figure out a reorder policy (when to reorder and how much to reorder) so we don’t run out of stock or reorder too often. I don’t have cost data (like holding or ordering costs), so I focused on balancing the frequency of reorders with the risk of stockouts.

**Key Steps**:

1. **Data Cleaning** (getting daily sales data)
2. **Forecasting** (using Excel’s FORECAST.ETS)
3. **Inventory Simulation** (day-by-day logic in Excel)
4. **Scenario Comparison** (testing different reorder points and quantities)
5. **Final Recommendation** (the best approach for fewer orders & minimal stockouts)

**2. Data Cleaning**

**Data Source**: A “Superstore” dataset with Order Dates, Sales, Ship Dates, etc.

1. **PivotTable**:

* I created a PivotTable to sum Sales by Order Date, giving me **one row per day**.
* I then pasted this into a **“CleanData**” sheet so Column A had **Date** and Column B had **Daily Sales**.

1. **Train/Test Split**:

* I had about 1,237 rows total. I used **rows 2–991** as **training** (80%) and **rows 992–1237** as **test** (20%).
* This ensures I can build my forecast on training data, then check accuracy on the test period.

**3. Forecasting**

1. **FORECAST.ETS** in Excel:

* In “**Clean Data**,” I used FORECAST.ETS(A992, $B$2:$B$991, $A$2:$A$991) to predict the daily sales for row 992 onward.
* I dragged it down to row 1237, so I have forecasted sales for each test date.

1. **Checking Accuracy**:
   * I computed the **Mean Absolute Percentage Error (MAPE)** by comparing actual vs. forecast in the test range.

**Chart**: **Figure 1: Actual vs. Forecast**

* I created a **line chart** (Date on the X-axis, Sales on the Y-axis) showing actual (Column B) vs. forecast (Column C) for the test rows (992–1237).

“We can see the forecast generally tracks the actual sales, with some underestimation on peak days.”

**4. Inventory Simulation**

After forecasting, I wanted to see how daily stock levels change. I built columns:

* **Column E**: Beginning Inventory
* **Column F**: Forecast Demand (linked to Column C)
* **Column G**: Ending Inventory = E - F
* **Column H**: Order Placed? = IF(Ending Inventory < ROP, “Yes”, “No”)

**Implementation**:

1. **Row 992**: I manually set a **Beginning Inventory** (say 12,000).
2. **Ending Inventory** (G992) = E992 - F992.
3. **Order Placed?** (H992) = =IF(G992<ROP, "Yes”, “No").
4. **Row 993**:
   * E993 = G992 + IF(H992="Yes", RQ, 0).
   * F993 = C993, etc.

“Figure 2 shows how daily ending inventory fluctuates, and each time it dips below ROP, we place an order the next day.”

**5. Scenario Comparison**

I tested two main variables:

1. **Reorder Point (ROP)**: The stock level at which I reorder.
2. **Reorder Quantity (RQ)**: How many units I order each time.

**Scenarios**:

* **Original**: ROP = 14,215, RQ = 3,424 → about 107 reorders.
* **Lower ROP** (12,000), same RQ (3,424) → 106 reorders (not much improvement).
* **Higher RQ** (5,000), same ROP (14,215) → 73 reorders (big improvement).

I made a small **table** summarizing:

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **ROP** | **RQ** | **# of Reorders** |
| Original (Baseline) | 14,215 | 3,424 | 107 |
| Lower ROP | 12,000 | 3,424 | 106 |
| Increase RQ | 14.215 | 5,000 | 73 |

**Observations**:

* Lowering ROP from 14,215 to 12,000 barely reduced the reorder count.
* Increasing RQ significantly dropped reorders to 73, which meets my goal of fewer orders.

**6. Final Recommendation**

Given my **main goal** is to reduce reorders while avoiding frequent stockouts, I concluded:

* **Keep ROP** at ~14,215 (so we reorder early enough to avoid stockouts).
* **Increase RQ** to ~5,000 (bigger batch size so inventory stays above ROP longer, leading to fewer orders).

**Trade-Off**:

* Higher RQ means holding more inventory on average, which might be expensive in real life if storage costs are high.
* However, I have no cost data, so I prioritize fewer orders.

*(If I had cost data, I’d do an Economic Order Quantity approach, but that’s not possible here.)*

**Conclusion & Takeaways**

1. **Forecast**: Achieved a MAPE of 45716.48%. The forecast is fairly accurate.
2. **Inventory**: The day-by-day simulation lets me see exactly when I reorder and how many times.
3. **Scenarios**:
   * Lowering ROP didn’t help much,
   * Increasing RQ drastically cut reorder frequency from 107 to 70.
4. **Final Policy**: ROP ~14,215, RQ ~5,000. This meets my goal of fewer reorders, with minimal risk of stockouts.

**Looking Forward**:

* If I had cost data (ordering vs. holding costs), I could refine the reorder quantity using an EOQ approach.
* Nevertheless, this project demonstrates the **core** of supply chain analytics: data cleaning, forecasting, and a practical reorder policy.