

Short Report: Inventory Forecasting and Par Level Recommendation System

What is the core business problem and why does it matter?

The main issue in this project is that a hotel chain with many bars is either running out of stock for popular liquor brands or ordering too much of items that don't sell quickly.

When they run out of stock, guests are not satisfied and the hotel loses potential sales. On the other hand, overstocking leads to waste and higher costs. This problem becomes more serious as the business grows. That's why a system that can help predict future demand and guide better inventory decisions is important.

What assumptions did you make? Why?

To keep the project manageable, I made a few assumptions:

1. Each bar and brand combination has at least 12 weeks of past data.
2. The data doesn't include special events or seasonal effects.
3. The lead time for restocking is stable, so I didn't factor in supplier delays.
4. I used weekly data instead of daily data because it's easier to forecast and manage.
5. I assumed consumption patterns don't change suddenly.

These assumptions helped me focus on building a working model without needing too many external data sources.

What model did you use and why did you choose it? Why not others?

I used the ARIMA model, specifically ARIMA(1,1,1). I chose ARIMA because:

- It is good for time series data like weekly consumption.
- It can work with trends in the data.
- It is simple and doesn't need external factors.

I didn't use more complex models like SARIMA (which handles seasonality) or Prophet (by Facebook), because ARIMA was enough for this case and easier to implement. I also avoided machine learning models like XGBoost or LSTM because we don't have large datasets or extra features.

How does your system perform? What would you improve?

The system forecasts the next 4 weeks of consumption for each bar and brand. It also calculates a safety stock and a par level, which is the recommended amount of stock to keep.

The system shows results using bar charts and line graphs. From my tests, it gives good suggestions and avoids extreme overstocking or running out of items.

Things I would improve:

- Use seasonal models like SARIMA if there's enough data.
- Add error tracking like MAPE or RMSE to see how accurate the forecasts are.
- Include external factors like holidays or events.
- Make it easier for bar managers to use by building a web dashboard.
- Make the model update automatically each week.

How would this solution work in a real hotel?

In a real hotel, the system would:

1. Get new sales data every week.
2. Forecast the demand for each item at each bar.
3. Calculate how much stock to order using the forecast and current stock levels.
4. Share this information with the bar manager, either in a report or through a dashboard.

The manager can then place orders based on the recommendations instead of guessing. This helps prevent running out of popular items and avoids wasting money on unused stock.

What would break at scale? What would you track in production?

If this system is used in many hotels at once, a few problems might come up:

- Forecasts could become less accurate as the number of bars and items grows.
- The same ARIMA settings might not work for every item.
- Processing time will increase.
- Unexpected spikes in sales (due to events or errors) could confuse the model.

In production, I would track:

- Forecast accuracy using MAPE or MAE
- How often stockouts or overstock happen
- Inventory turnover (how fast stock is used)
- Time taken to generate forecasts
- Alerts when forecasts don't match actual usage