# ***Candidate information:***

Name: Mohammad Arbaz

Mail: [arbaz7862002@gmail.com](mailto:arbaz7862002@gmail.com)

Resume link: [resume link](https://drive.google.com/file/d/12VS6-uHC_gSKVVFuCIhp2YZRPyJNwxtA/view?usp=sharing)

**Price Optimization Documentation**

**Problem Statement:**

The main task here is to come up with a way to suggest the **best possible MRP (price)** for each product (SKU).  
 The idea is that if we can find the right price, we should be able to **increase sales volume while also keeping profit high**.

In real life, pricing decisions depend on a lot of things – cost of production, logistics, promotions, bulk (B2B) vs. retail customers, and of course what competitors are doing in the market. In our dataset, we don’t have all of that detail, so we are trying to build a data-driven framework from the sales history we do have.

**Solution:**

**Data processing and feature Engineering:**

* The first step was to **clean the sales dataset**.  
  + Removed cancelled orders (they don’t contribute to revenue)
  + Dropped rows where amount, quantity, or price was zero or missing.
  + Standardized date formats (some were DD-MM-YY, some DD-MM-YYYY).
  + Calculated unit price as amount / quantity.
* Then I **aggregated sales per day per SKU** so that we can see how each product was selling at each price point on each day.
* I also created a few **extra features**:
  + Day of week and month (for seasonality).
  + Promo flag (if available).
  + Summary stats like min price, max price, average price, etc.
  + For each SKU, I also looked at which price historically sold the most units (“best historical price”).

**Creating new MRP suggestion for each product:**

**Approach 1: *Simple Conditional Modelling***

* If there are multiple constraints from the client side that the COGS is too high and they say that each promotion is a discount card and if they also say that each customers are buying in buld as a b2b order instead of direct order then we simply use some conditional modelling to derive the mrp.

**Approach 2: using a regression model.**

I split the solution into two main parts:

1. **Price → Demand modeling** For each SKU, I tried to understand how sales quantity responds to price changes.  
    I used a few different models:
   * Log–log regression (gives constant elasticity, so you can interpret % change in demand per % change in price).
   * Log–log quadratic (allows elasticity to change with price levels).
   * Spline regression (more flexible curve fitting).
   * Random Forest (tree model, captures non-linear relationships without assumptions).
2. To check if these models actually make sense, I did a **holdout validation** – basically kept aside the last 20% of days and tested how well the models predict demand there. I tracked metrics like R², MAE, etc. This is not perfect, but it helps show whether the model has any predictive power.
3. **Time-series forecasting with SARIMAX** After estimating how price affects demand, I also wanted to account for seasonality and trends. For this I used SARIMAX models, which allow me to include **price and promotion as external drivers**.  
    This lets me simulate: “If tomorrow we keep the price at X, how many units will we sell over the next 30 days?”

### **How I simulated optimal prices**

For each SKU:

* I built a **grid of possible prices** around the current price (say ±25%, but not below historical min or above max).
* For each candidate price:
  + Use SARIMAX to forecast demand for the next 30 days.
  + Multiply (price – unit cost) × forecasted quantity to get expected profit. (Since actual cost wasn’t given, I used 70% of the latest price as a proxy).
* The price that gave the **highest profit** was recorded as the “optimal price” from that model.

I also kept a **baseline forecast** at the current price so we can compare if the model’s suggested price actually improves things.

### **Outputs generated**

All model recommendations per SKU – debug file

Best recommendation per SKU – debug file

One-line per SKU summary (sku\_one\_line\_summary.csv) – final file to refer

### **Important notes and limitations**

* With the given dataset, results might not always look realistic. That’s because we don’t have true costs, discounts, competitor prices, or ad spend. Those are all important for real-world pricing.
* The “optimal price” we get here is based only on the historical sales and assumptions. In reality, the only way to know for sure is to **test it live** (A/B experiments).
* The point of this exercise is not to give a magic number but to show a **framework**:
  1. Clean data →
  2. Understand price vs. demand →
  3. Forecast demand at future prices →
  4. Pick the one with the highest expected profit.

### **Closing thoughts**

This approach gives the client a structured way to think about pricing decisions. Even with limited data, we can simulate what different prices might do, validate the models a bit, and provide a recommendation along with confidence signals (validation metrics).

In practice, this would be the starting point. With more complete data (true COGS, competitor benchmarks, promotions, stock availability), the model would become much more powerful and closer to real-world pricing optimization.

During the development of this solution, I utilized ChatGPT as a coding assistant. The tool was particularly helpful in quickly translating problem logic into working code, allowing me to focus more on designing the overall approach and deriving the right logic for the business problem. This not only accelerated the coding process but also reduced time spent on resolving syntactical or implementation-level issues.