RETAILER CASE STUDY

Note for every step documented –
1 – Given data in RED.
2 – Technique in BLUE.
3 – Output of technique in PURPLE.

PAST DATA PER ITEM –

→ Item ID
→ Category
→ Assigned Shelf Space
→ Array of Neighboring Items
→ Item Sales
→ Array of Neighboring Item Sales

GLOBAL DATA ACROSS THE SYSTEM –

→ Minimum Shelf Space per Item
→ Maximum Shelf Space per Item
→ Total Shelf Space

DERIVED DATA FOR EACH ITEM –

→ Item ID
→ Category
→ Best Sales Duration
→ Array of Closest Neighbors and Scores – Data-Driven
→ Array of Closest Neighbors and Scores – Theory-Driven
→ Individual Probable Sales
→ Combination Probable Sales

PROBLEM GOAL –

To find the shelf space and respective neighbor items for each item such that sales are maximized by considering the following –

1. ***Affinity to buying neighbor upon purchase of an item***
2. ***Inventory forecasts (shelf space) based on past demand***
3. ***Season item purchase patterns***

SOLUTION OVERVIEW –

This is the brief of how the solution would proceed –

1. Identify seasons and trends in forecasts for each item from past data and add a season variable to the present data called *Best Sales Duration*.

2. Run two models for neighbor decisions for each item –

a) Exploitation Model – Uses past data and neighbors that qualify above a relevance cutoff score are stored in array in descending order of their scores and the array is named - *Array of Closest Neighbors and Scores – Data-Driven*.

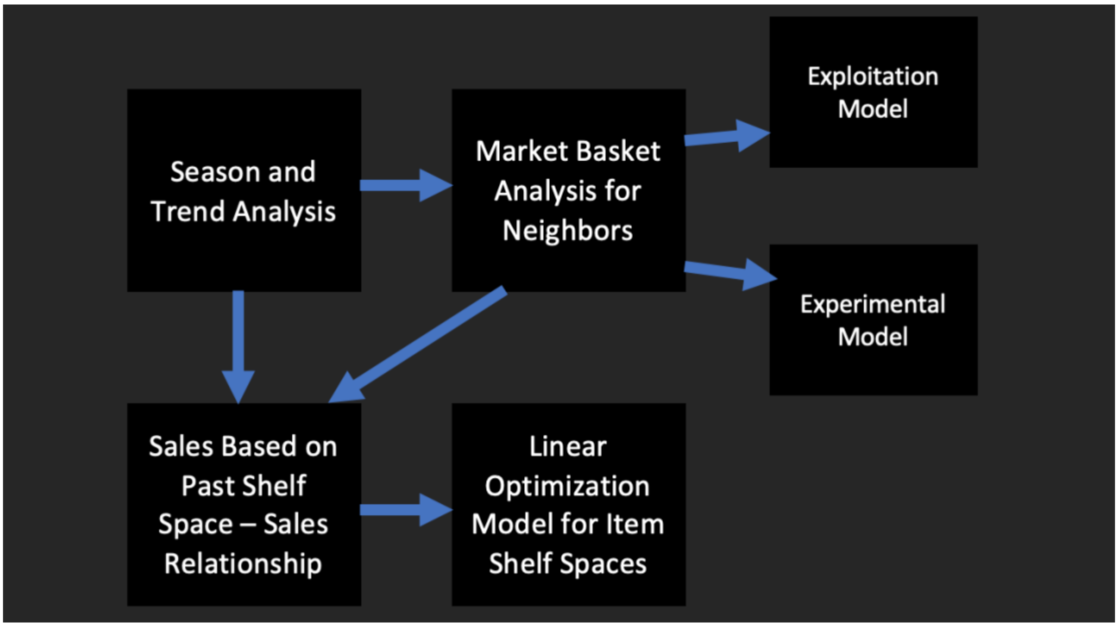
b) Experimentation Model – For the item combinations that have not ever occurred in the past, the probable relevance of one item to another will be simulated using knowledge of the qualitative and quantitative factors associated with relevance of one product to another. The estimated neighbors along with their scores are arranged in descending order in the array names - *Array of Closest Neighbors and Scores – Theory-Driven*.

3. The sales for each individual item is mapped to shelf space and neighbors of the item using past data, in two ways –

a) By mapping sales to individual items – *Individual Probable Sales*
b) By mapping sales to combinations of neighbors – *Combination Probable Sales*

4. Then the derived data from the last three steps is used as input variables to an optimization model that gives the shelf space for each item.

Diagrammatic Systems View -



DETAILED SOLUTION –

Step 1 – Season and Trend Analysis

Given: Item ID, category, and sales from past data.
Use: For each item, apply multiplicative decomposition on the time-series data.
To: Derive (season, trend factor) as an output for the *Best Sales Duration*.

Step 2A – Neighbor Scoring: Exploitation Method

Given: Item ID, Array of Neighboring Items, Item Sales, Array of Neighboring Item Sales from past data.
Use: The Apriori Pruning algorithm to remove associations that do not exceed the required support and confidence threshold.
To: Derive the most suitable neighbors based on their confidence scores and generate the variable - *Closest Neighbors and Scores – Data-Driven*.

Note 1: It is also possible to have a sub-experimentation model that varies the confidence and support cutoffs to see how the sales vary, but that would require extra computation power.

Step 2B – Neighbor Scoring: Experimentation

Given: Item ID, Category, Best Sales Duration from past and derived data.
Use: A case-driven simulation model based on qualitative factors such as category and domain-expertise, as well as quantitative factors such as seasonality, transitive deductions (if A and B are bought together, and B and C are bought together, than how likely are A and C to be bought together), and already known distributions of item set purchases.
To: Derive the most suitable neighbors out of the combinations that have never been tried before but may be a profitable experiment and name the variable - *Array of Closest Neighbors and Scores – Theory-Driven*.

Step 3A – Sales Based on Shelf Space: Individual Items

Given: Item ID, Assigned Shelf Space, Item Sales, Best Sales Duration from past data.
Do: Run a simple Bayesian-based time-series model on sales given item shelf space and Best Sales Duration and run this for items that did not run out. For items that DID run out construct an experimentation system that uses theoretical measures to increase inventory to the right amount and estimate sales as an approximate fit.
To: Determine sales of each item based on shelf space and create the variable - *Individual Probable Sales*.

Step 3B– Sales Based on Shelf Space: Combinations of Items

Given: Item ID, Assigned Shelf Space, Array of Neighboring Items, Item Sales, Array of Neighboring Item Sales from past data.

Do: Get total shelf space for each combination based on Step 2A. Run a simple Bayesian-based time-series model on sales given combination shelf space and run this for item sets that did not run out. For combinations that DID run out construct an experimentation system that uses theoretical measures to determine the new composition of products in the combination and estimate sales as an approximate fit.
To: Determine sales of each combination based on shelf space and create the variable - *Combination Probable Sales*.

Step 4 – Optimization Model

Given – Individual Probable Sales from derived data.
Do:

Variables – Shelf space for each item {s_1 to s_n}
Constraints –
1. All s_x <= Maximum Shelf Space per Item
2. All s_x >= Minimum Shelf Space per Item
3. Summation {s_1 to s_n} = Total Shelf Space

Objective –
Maximize {
    Summation over 1 to n items -
        (Individual Probable Sales Factor * s_x)
    }

To: Get shelf space for each item.

Step 5 – Now that we have the shelf space for each model, we just adjust the items around based on the neighbor scores in this order –

1. *Combination Probable Sales*
2. *Array of Closest Neighbors and Scores – Theory-Driven*
3. *Closest Neighbors and Scores – Data-Driven*

Note 2: The reason I've put the experiment results before the exploitation results is because the combinations result already relies on the strongest of the exploitation results, and hence the experiment can be afforded. The remaining exploits not included in combinations are then arranged as usual pairs of two. The ration in which we weight these can of course be another experiment all in its own.