**How we are optimizing space in godown?**

**1. Zoning and Layout Optimization**

* **Dedicated Hazardous Cargo Zone**: Allocate a specific section for hazardous materials, ensuring compliance with safety regulations. Use fire-resistant partitions if needed.
* **Stackable vs. Non-Stackable Areas**: Identify stackable goods and assign them to high-storage zones using vertical racking. Non-stackable goods can be placed in easily accessible lower racks.
* **Fast-Moving Cargo Near Entry/Exit**: Since most cargo stays only 2-3 days, high-frequency goods should be placed near loading/unloading areas.
* **FIFO Optimization**: Use a first-in, first-out system to prevent congestion and delays.

**2. Dynamic Space Allocation Using ML**

* **Predictive Slotting**: Train an ML model using past cargo movement data to optimize slot assignments.
* **Heat Maps for Congestion Avoidance**: designed heat maps and optimize pathways for forklifts and personnel.
* **Automated Scheduling**: Implemented a simple algorithm to pre-assign storage slots based on cargo type, and departure urgency.

**Zoning Strategy**

Since the godown is a 3D space (900 sq.m × 15 ft height), we should divide it into logical **zones** and **slots** based on cargo type, handling requirements, and storage constraints.

**1. Zoning Based on Cargo Type**

* **Hazardous Zone** (Separate, well-ventilated, near emergency exits)
* **Small Goods Zone** (Cargos that are transported by manual labor)
* **Large Goods Zone** (Cargos that are transported by forklifts and other machinery)
* **Fast-Moving Zone** (Near exit to reduce handling time)

**2. Slotting Strategy**

Each **zone** is further divided into **slots**, with each slot having a unique identifier These slots are dynamically assigned based on cargo priority, type, and volume.

**Defining Layout**

**Step 1: Determine Slot Size**

* **Standard Pallet Size:** 1.2m × 1m
* **Slot Size:**
  + **Width:** 2.5m (to fit 2 pallets side by side with spacing)
  + **Depth:** 2.5m (to fit 2-3 pallets front to back)
  + **Stacking:** 1-2 levels (depending on cargo type)

**Step 2: Compute Slot Count**

* **Usable Space:** 675 sq.m (after reserving 25% for movement)
* **Slot Size:** ~6.25 sq.m (2.5m x 2.5m)
* **Total Slots:** **~110-120 slots**
  + Each slot holds **5 pallets** → **~550-600 pallets total**
  + Some zones (hazardous, fast-moving) may have smaller slot sizes

**Step 3: Adjusted Zone Allocation**

1. **Fast-Moving Cargo (20% of available space)**
   * **20 slots** (Easy access, minimal stacking)
   * **Slots A1 to B10**
2. **Small Cargo Zone (30% of available space)**
   * **30 slots** (Stacking allowed, 5-10 pallets per slot)
   * **Slots C1 to E10**
3. **Large Cargo Zone (35% of available space)**
   * **35 slots** (Stacking up to 2 levels, 5-10 pallets per slot)
   * **Slots F1 to I5**
4. **Hazardous Zone (15% of availabl space)**
   * **15 slots** (No stacking, 5 pallets per slot)
   * **Slots I6 to J10**
5. **Pathways (~25% of total space)**
   * **3m-wide machinery pathways & 1.5m manual aisles**

**Cargo Data Features for ML Model**

To optimize placement, the model should consider these features:

1. **Cargo ID** (Unique identifier)
2. **Estimated Size**: (Small, Medium, Large, Oversized)
3. **Weight** (For stacking feasibility)
4. **Stackability** (Boolean: Can this be stacked?)
5. **Hazardous Material** (Boolean: Needs separate storage?)
6. **Expected Departure** (For prioritization)
7. **Handling Requirement** (Forklift/manual, fragile, etc.)

**Cargo Size Categories**

We define cargo sizes as:

| **Category** | **Approx. Dimensions (m)** | **Example Cargo** |
| --- | --- | --- |
| **Small** | Up to 1.5m (L/W/H) | Boxes, cartons |
| **Medium** | 1.5m - 2.5m (L/W/H) | Pallets, crates |
| **Large** | 2.5m - 3.5m (L/W/H) | Heavy machinery, big crates |

|  |
| --- |
| **Model Details:**   * **For storage optimization , we are using DNNs, (Deep Neural Networks).** * **It consists of multiple layers of neurons, with neuron size casually decreasing as more layers are added.** * **Due to multiple layers they are prone to overfitting, to avoid this we added several Dropout Layers.** * **Dropout Layer basically shuts off some neurons during training, through which model is not able to memorizethe data and is forced to identify patterns.** * **We have also used SELU (Scaled Exponential Linear Unit) activation function in some hidden layers , which is found to perform better.** * **We have used NADAM (Nesterov-accelerated Adaptive Moment Estimation) optimizer, which identifies the trends quickly in case of classification problems.** |

Tech Stack:

* Sklearn – for data preprocessing pipeline
* Tensorflow & keras – for training DNN.
* Google Gemini – for giving insights and tips about unloading.
* Fast API – for model deployment