The Planogram Challenge

Welcome to the useAIble planogram challenge (developer’s version, see contact below for “production version”). In this challenge, we have 288 shelf slots representing empty spaces on a 12ft x 6ft shelf. There are 5,000 items generated by the database. There are 10 metrics on which the items can be scored (for example, in production one metric might be GMROI per square inch). Each item is randomly assigned a score for each of the 10 metrics. Each item is randomly assigned a color so that you may observe the virtual planograms being constructed.

The current settings reflect that there may only be 10 facings of any single item. Note, there is a “small” version that has 24 slots and 500 products to practice various training settings before running at production scale.

Before challenge training can began, sample data must be generated. This will populate the database with randomly generated items, associated attributes and metrics scores for each item. There are two modes of operation; large scale and small scale. The large scale is more representative of a production system but may take significantly longer to train. The small scale mode allows quicker results to be generated and evaluated. A different dataset must be generated for each of the two modes.

The machine learning engine is fed a single input called Slot, representing the space on the shelf. The challenge accepts a single output from the machine learning engine called Item. This represents one of the 5,000 items available for placement on the shelf.

When the spaces have all been filled, the challenge produces a score that represents the sum of each metric for each item in the planogram. This will be the machine learning engines final score for the simulated planogram (this is called a session score or an epoch score).

This process will take place thousands of times as the system “learns”. The system is not directly told the value of any given item. This helps simulate real retail spaces where items are judged as part of an entire planogram and not as individuals. For example, Cheerios might be a best seller; however, an item on a 12ft shelf filled with only Cheerios would be useless in the real world. To simulate this need for a “balanced” planogram the planogram is scored as whole and the machine learning engine is not told the value of any parts. The machine learning engine is told on a per item basis if it has selected the item over the duplicate amount of times (which defaults to 10 facings).

The challenge may be run one of three ways:

1. **Sessions**: This option gives all engines an equal number of tries and shows the resulting high score for each engine.
2. **Time**: This option gives all engines an equal amount of elapsed time to work on the challenge and displays the high score for each engine.
3. **Score:** This lets each engine run until it has hit a predetermined score. The elapsed time and number of sessions are using to evaluate the engine’s performance when the score is the target.

Note that it is recommend that the challenge be run directly from the executable in release mode to avoid allowing the debugger to taint results.

Challenge constraints are:

1. No pre or post “coded” calculations are allowed to benefit any engine.
2. Scoring or network reinforcement should happen at the session or epoch level and not at the cycle level.
3. When the challenge is presented with more than the allowing number of duplicates, it returns an error and the engine must select and alternate item. The default number of duplicate facings is set to 10.

## Reference:

* 288 Slots
* 10 Metrics
* 5,000 Items
* One input called Slot (for slot #)
* One output called Item (for which item to place on the shelf)
* One final score representing the sum of each items scoring for each of 10 metrics

**For more information contact Jeff.Horton@useAIble.com.**