Retail Proof of Concept

useAIble

# Running the App

1. For you to be able to run the application, you will need the following:
   1. SQL Server
   2. Python 3.5
   3. pypiwin32
   4. pythonnet
   5. Tensor Flow
2. The last 3 components are installed by accessing the Python script folder through command prompt and entering the command pip install <component>

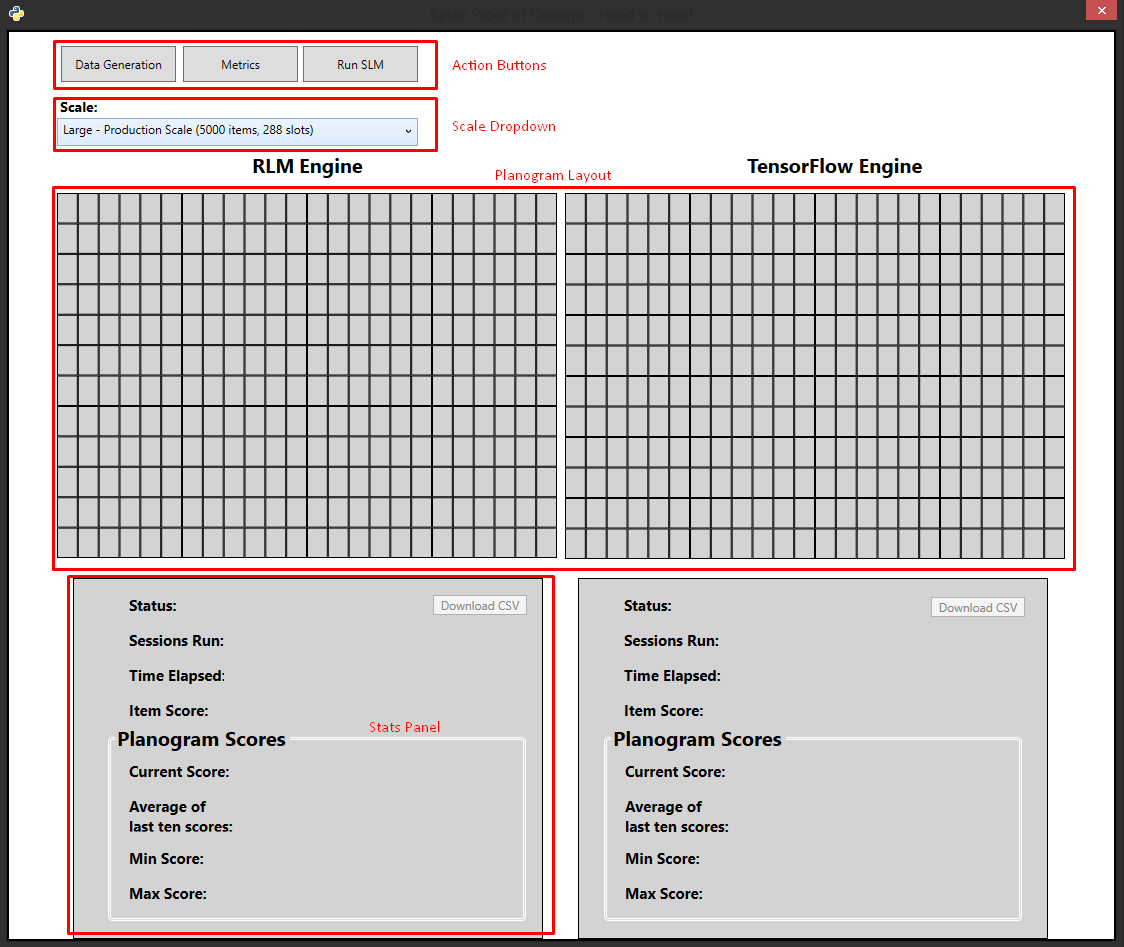


1. By default, the SQL Connection strings are configured to your local SQL Server and uses integrated security. In the event that you are not, you can setup the connection strings through the ConfigFileManager.py File.
2. This application relies on your registry to find python. If it’s not installed or you’re using a portable version, we will also need to copy the app config manually, located at the project directory, shown below, to the Python directory, usually at C:\Users\youUserName\AppData\Local\Programs\Python\Python35



1. Once the above steps are done, rebuild the code.
2. You can find 2 components for the App
   1. The C# One for the RLM (This is needed by the python Tensor Flow App)
   2. The Python one for Tensor Flow
      * The goal is to be able to compare both engines seamlessly
3. The RetailPOC App to test the RLM only can be found under ExampleApps
4. The RetailPOC\_TensorFlow can be found under CompetitorComparableApps Folder
5. You will need to set any of the 2 as startup project to run them.

# Parts of the Main Screen

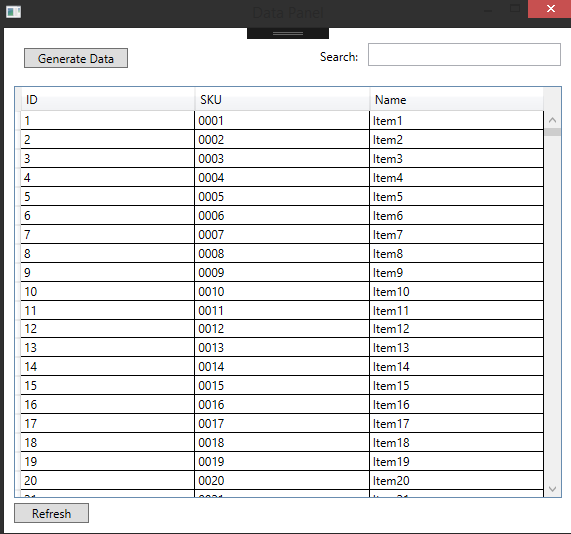


1. Scale dropdown – allows you to select the size of your planogram layout. This is useful to test engine’s learning faster in a smaller dataset. Changing size will prompt you to generate data.
2. The Planogram Layout – shows the current placement of items on the planogram shelves.
3. Action Buttons – buttons that direct the user to specific actions in optimizing a planogram
4. Stats Panel – information panel that displays the statistics of the last training done.
   1. Score – the training score of the last training
   2. Sessions Run – the number of sessions done
   3. Time Elapsed – the total time elapsed of the training
   4. Item Score – shows the score of the hovered on the planogram layout. This shows the raw score and beside it is the percentage of it from the range of scores of all items on the layout.
   5. Planogram Scores
      1. Current Score – the score of the latest session
      2. Average Score – the score average of all the training sessions done with the current data set
      3. Minimum Score – the lowest score of all the session’s combined
      4. Maximum Score – the highest score of all the session’s combined
5. Download Report Button – downloads a csv file of the statistics for each session.

# Generating Mock Data

1. Click on the Data Generation Button.
2. You’ll be directed to the Data Panel as shown in Figure 2

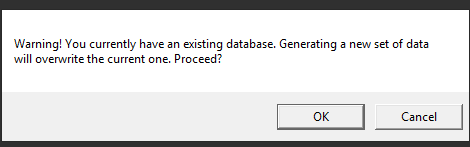
Figure 2



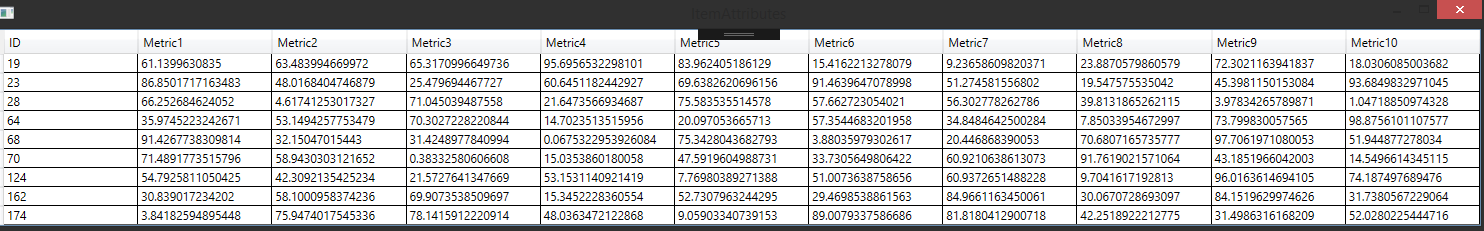
|  |
| --- |
| **The Data Panel**  Using the Data Panel, you can also query an item’s attributes through the available search bar located at the upper right. You can do a query using the SKU or the name. |

1. Click on Generate Data.
2. If you already have an existing data set, there should be a warning to tell you that it will be overwritten if you push through with a new generate data process. See Figure 3.

Figure 3



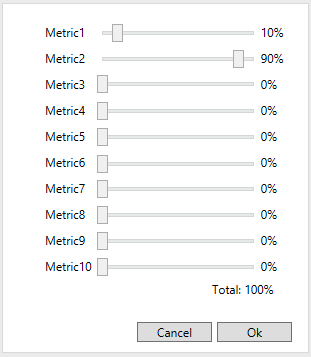
1. If you Double click an Item Row, it should show you the metrics for that item. See image below.



# Setting up the Metrics

1. Click on the Metrics Button
2. You’ll be directed to the Metric Panel where you can set the weights of each Metric through Sliders as shown on Figure 4.

Figure 4

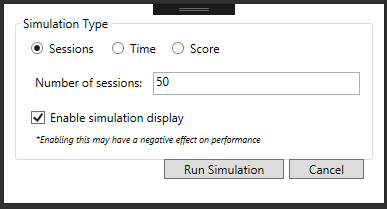


1. The Metric total weight should always be a hundred. In order to move some metrics, you’ll need to lower the others to give more weight to one.
2. Once you click okay, the new metrics will apply to all future simulations unless changed.

# Running a Simulation

1. Click on the Run SLM Button
2. You’ll be directed to the Simulation Panel as shown on Figure 5.

Figure 5



1. Checking the Enable Simulation Display Checkbox will allow the user to see the engine fill up the slots with items as it goes through training, however, this will make the optimization slower.
2. Setting Simulation Type to Sessions, should prompt the user to enter the number of sessions. This will dictate how many sessions the engine will run.
3. Setting Simulation Type to Time allows user to enter the number of hours that the engine will run.
4. Setting Simulation Type to Score allows user to enter the score of which the engine will continue running until it reaches the set score for 10 Consecutive Times.
   1. Clicking on the Max button will set the target score to 80% of the total possible score.
5. After running the simulation, you should get a message to confirm that is done as shown on Figure 6.

Figure 6

