Vending Machine Prediction Program
Utilizing ARIMA

Ben Shawn, bps150030@utdallas.edu; Roland Smith, Roland.Smith@utdallas.edu; Jon Ibarra, jci150030@utdallas.edu; Catherine O’Mary, cmo150030@utdallas.edu

CS 4485 / Fall 2018
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract
The current process of drink deliveries to Keurig Dr Pepper vending machines is not an exact science. The schedule dictating which machines are restocked on which days is not founded in optimal data analytics, but on trade. The quantity of product loaded onto delivery trucks is also based on the supervisor’s personal discretion.

This project aims to improve the efficiency of the machine delivery system by using ARIMA machine learning algorithm and historical delivery data to provide better predictions of when products run low, saving drivers from visiting already stocked machines and the company from missing out on sales where product is fully depleted. In addition, the algorithm can also be used to calculate the appropriate quantity of each product to be loaded onto the trucks.

Impact
While our algorithm is unable to give precise predictions, it provides an upper bound on the number of cases needed during deliveries which saves labor in loading/unloading product from the delivery trucks. Additionally, the code is easily modified if Keurig Dr Pepper chooses to collect delivery data in bottles instead of cases for more precise predictions. Configurable machine-product capacity will further the efficiency of the delivery by more accurately flagging machines that need refills.

Performa  
ARIMA predictions were within -0.5/+1 range of actuals 56% of the time for our test machines. Notably, if we exclude the last day of testing (Nov 29) which would be the most inaccurate due to its length of time from the ARIMA rate calculations (Nov 16), predictions were within range 72% of cases. Notable deviations likely result from unprecise data collection (deliveries are reported in ceiling[# of cases] instead of # of bottles leading to overestimating amount of cases).

Architecture

1. Input .csv files (provided by DP)
2. MachineConfigGenerator (coded in C#) Generates editable default values for each machine –product pair (machine_config.csv)
   - Default product capacity – split machine capacity evenly between all products in machine
   - Default refill threshold – product dips below 10% of product capacity
3. ARIMA (coded in python) finds the predicted rate of depletion of each machine-product pair
4. Takes rates & machine-product info, generates csv for how often per week a machine will need refilling
5. Assumes machine is completely full with each delivery made to it, predicts levels of all machine-product pairs and flags machine-products that need refills.

Results

<table>
<thead>
<tr>
<th>Number of predicted cases minus actual cases</th>
<th>[0-0.5]</th>
<th>[0.5-1]</th>
<th>[1-3]</th>
<th>[3-5]</th>
<th>[5-10]</th>
<th>&gt;10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deliveries November 21&lt;sup&gt;st&lt;/sup&gt;</td>
<td>40</td>
<td>60</td>
<td>20</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deliveries November 28&lt;sup&gt;th&lt;/sup&gt;</td>
<td>100</td>
<td>80</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Average # of predicted cases minus actual cases (for 9/21) = 3.2

Average # of predicted cases minus actual cases (for 9/28) = 3.05

Note that outliers (machine –product pairs with predictions greater than 10 cases off) were all from the same 11 machines, possibly indicating that the model we trained is not universally applicable.

Summary
Although our code is only the first stepping stone in streamlining the delivery process for refilling vending machines, our project has discovered additional avenues our sponsor may choose to explore, as well as provided a tool to be honed to improve the efficiency of product deliveries. After our project completion, our sponsor now knows obstacles standing in the way of precise predictions of vending delivery patterns. Additionally, we have also created two reporting programs to analyze ARIMA output into a more useful format.

Going forward, we expect that the true benefits are yet to be realized through more precise data collection, wide application of the program (we focused on 20 chosen machines), elimination of null values in the data, and adjusting capacity of product in a given machine from the assumed value to the accurate value. None the less, the infrastructure to support these additional improvements is now in place, giving it much future potential in the effort to make deliveries more efficient.