Data Analysis and Programming for Operations Management

2022-2023 S1

Technology & Operations Management

Faculty of Economics & Business

University of Groningen

Individual Assignment

Summary of requirements

- This is an individual assignment.
- The assessment will be done based on a written report and an oral defense.
- The report should not exceed 3 pages. This excludes graphs, tables, and your source code).
- The deadline for delivering your report is October 28, 2022, 17.00
- The oral defense will be scheduled in the exam weeks.
- To take the defense, you will need to enroll for a time slot. Further details will be announced on Brightspace.

Overview

The assignment is based on a case inspired from real-life. The deliverable is a concise report that explains all the choices you made and the activities you carried over in your assignment. The report should not be just a collection of notes; it needs to have a proper flow (introduction, conclusions, graphs, etc.). The final source code needs to be properly formatted and included as an appendix. The report will be submitted on Brightspace. The assessment criteria for the assignment are the following:

- Solid reasoning in explaining modeling choices
- Good coding practice with proper use of methods and libraries
- Quality of the report
- Quality of the oral defense

The assignment will assess whether you master various programming skills that were practiced throughout the course (e.g., input and output, data structures, Elasticsearch, Gurobi, visualization). The methods and libraries used in the practicals provide guidelines to answer various questions in this assignment, yet you are free to explore other methods and libraries. Many roads lead to Rome!

Note that helping each other is allowed, yet the assignment is individual. That is, plagiarism is strictly forbidden! It is essential that you make your own decisions when coding, modeling, and visualizing, as you will defend your decisions in your report and in the oral defense.

In the oral defense, we will test your understanding of your own report and code, as well as your understanding of what is covered in Dapom in general. The questions you will face in the oral defense largely depend on your report and code. For instance, you must be able to explain the objectives and functionality of the libraries you used. However, you are not expected to memorize all parameters of Gurobi or Elasticsearch libraries. Typical questions may include:

- Which part of your code determines the average daily sales?
- Could you explain the code on the top part of the page 8 of your report?
- How would your optimization model change if there were a lower bound on the number of products that can be stored in the current warehouse?
- "Teacher points at a part of the code of another student." Could you explain what this piece of code is doing?

The oral defense will take half an hour. The defense will be done with two students simultaneously to entice discussion. ***It is not allowed do take the defense at the same time slot with the teammate you worked with at the practicals.***

Case description

In this assignment, you will consult Belsimpel—an online store for mobile phones and phone contracts with headquarters in the city of Groningen. Note that the case text and data you receive are inspired by the challenges that Belsimpel faces, but, for reasons of confidentiality, do not reflect reality accurately.

The product assortment of Belsimpel is rapidly expanding. As a result, their warehouse has become a bottleneck for their operations. Belsimpel does not intend to relocate the warehouse in the short-term. Therefore, it appears that the only way forward is to store a limited collection of products at the current warehouse and move the rest to a rental warehouse elsewhere. The advantage of the current warehouse is its efficiency. Thanks to the well-established organization of warehouse operations developed over the years and the availability of personnel with the right skills and experience, Belsimpel is able to offer next-day deliveries to its customers for any product that is stored at their current warehouse. It might not be possible to provide the same service quality for those products stored at a rental warehouse—where deliveries may take several days, and, it is expected that this will have a significant impact on the sales.

Belsimpel's problem is to decide which products to keep in the current warehouse so that profit losses are kept at minimum. This allocation problem has ties with potential changes in the customer behavior in response to delivery times as well as the organization of the operational processes of managing inventories at the warehouse. These are briefly discussed in the following.

Product classes

Belsimpel wants to use product classes to effectively manage its large assortment. The classification will be based on the importance of products which is measured by the profit they generate (sales x profit margin). There will be three classes of products. These will correspond to the top 20%, following 30%, and the last 50% of products, respectively.

Managing inventories

Belsimpel will be using a periodic-review base-stock policy for managing inventories. The replenishment interval will be one week for all products. To avoid stock-outs, base-stock levels will be set on a product by product basis. This will be done following the $\mu+z\sigma$ rule, where z is a multiplier that is specific to each product class, and, μ and σ respectively stand for the mean and the standard deviation of the demand over the replenishment interval. The multipliers for the three product classes will be the z-scores[1] that correspond to probabilities 0.99, 0.95, and 0.90, respectively. Thus, the safety stocks will be relatively larger for more important products. The base-stock levels are critical as the storage space allocated to a product should be large enough to cover its base-stock level.

Storage space

The efficiency of warehouse operations hinges on swift pick-ups around the warehouse. This is achieved by organizing the warehouse in racks of standard pick-up boxes. This is viable as product sizes are relatively small. To facilitate pick-ups, each box is assigned to a single product yet multiple boxes can be assigned to the same product. This suggests that the total warehouse space as well as the storage space allocated to each product can be measured in terms of boxes. Belsimpel has place for 960 pick-up boxes in the current warehouse. The dimensions of a standard pick-up box are given in cm's. It is known by experience that approximately 90% of the box volume can be used effectively, as the dimensions of individual products often do not perfectly match that of the pickup box.

Impact of delivery times

The data analytics team of Belsimpel has recently run an analysis to get a better understanding of demand dynamics. The results of this analysis reveal the following. First, if next-day deliveries cannot be offered for a product then its demand may drop significantly. Fortunately, the implications will be less severe for products that generate higher profits. In particular, it is anticipated that the drop in sales for the three product classes will be 20%, 30%, and 50%, respectively. Second, offering different delivery times for product couples with highly correlated demands may have adverse effects on sales. For instance, a customer may refrain from ordering a high-value mobile device just because a complementary accessory has a longer delivery time. Besides, even if the customer places an order, the products have to be shipped separately, thereby increasing the delivery costs. Because it is hard to anticipate the actual impact of such incidents, it has been decided that the allocation decisions should obey the following rule. Consider a product couple such that the correlation coefficient between their demands is larger than or equal to 0.6 and the first product belongs to a higher class than the second product. The rule suggests that, if we store the first product at the current warehouse, then we should also store the second product at the current warehouse.

Tackling the allocation problem

The experts in Belsimpel consider two options to tackle the allocation problem. First, the problem can be addressed by the following simple heuristic idea. We prioritize products based on some measure. Then, we place products (along with those whose demands are highly correlated) in the current warehouse in decreasing order of priority, until we have no storage space left. Second, it can be possible to model the allocation problem as an integer programming problem. This will be a variant of the well-known 0-1 Knapsack problem[2], with additional constraints to handle correlated product couples. For all methods, the aim will be to minimize the daily profit loss. This is the sum of daily profit losses of products that are not stored in the current warehouse.

Task

As a consultant, your task is to help Belsimpel in solving their allocation problem. This will involve data gathering and analysis, devising solution methods and implementing them, and running a sensitivity analysis to explore how allocation decisions and corresponding sales losses are affected by a variety of parameters.

Available data

Belsimpel provides you with the following data files.

- *sales.csv*: Historical sales data over a number of days. Each line contains a date and a product id. In your analysis you will assume that these correspond to customer demands (hence you will ignore stock-outs that Belsimpel might have experienced).
- *margins.csv:* Profit margins of each product.
- *dimensions.csv*: Dimensions of each product.

Before proceeding with your analysis, ***examine the structure of the data files and make sense of what each column and row stand for***. The files margins*.csv* and *dimensions.csv* can be directly read by a csv reader. For *sales.csv,* you will use Elasticsearch. This is critical as Belsimpel intends to repeat the same analysis periodically, and they wish to be able to do this with much bigger order files. Keep in mind that the file may take some time to import. If it takes longer than an hour and you do not trust whether the import is actually running, then go to

https://localhost:9200/insert-your-index-name-here/_count

and after a minute or so refresh the count by hitting F5. The count should increase.

If security is enabled in Elasticsearch, you must also provide username (typically 'elastic') and password. If you disabled encrypted communication in Elasticsearch, you need to use http instead of https.

Note that the sales data as well as all parameter values are randomized for reasons of privacy. They do yet closely resemble their real counterparts.

An overview of the analysis

Step 1—data gathering and processing

In the first step, you will read the data files and compute the basic statistics:

1. Gather the following information for each product: (1) total demand on each day, (2) profit margin, (3) dimensions. A **requirement** is that you load the orders file in Elasticsearch, and that you will never have a complete copy of all individual orders in Python/Pandas. In this way, you will make it possible to scale your solution to bigger order files (more products, more days). This implies that you need to use Elasticsearch to aggregate individual orders to days.
2. Compute the average and standard deviation of the daily demand for each product. Plot these on an errorbar[3] where data locations and error bar sizes stand for the averages and standard deviations, respectively. Make sure that products are sorted in a descending order based on their average daily demands for better visibility.
3. Compute the average daily profit generated by each product and plot them on a histogram using a proper number of bins.
4. Compute the volume of each product and plot these on a histogram using a proper number of bins.
5. Briefly comment on your observations based on the graphs you have produced.

Next, you will look into the product classes.

6. Make a list of products in each product class.
7. Plot the number of products and the total average daily profit generated by the products in each product class on separate bar charts.
8. Plot the average daily profits of each product on a bar chart where products are sorted in a descending order based on the profit they generate. Mark the products that are at the boundaries of the product classes so that they are visible.
9. Briefly comment on your observations based on the graphs you have produced.

Now you will analyze the base-stock levels.

10. Compute the average and standard deviation of the demand over the replenishment interval for each product.
    Hint: If the average and the standard deviation of the daily demand are μ and σ; then those of the demand over T days will be $T\mu$ and $\sqrt{T}\sigma$, respectively.
11. Compute the base-stock level for each product.
12. Compute the number of pick-up boxes required for each product, based on the product volumes and that of the standard pick-up box. Plot these on a histogram using proper number of bins.
13. Briefly comment on your observations based on the graph you have produced.

Finally, you will identify the product couples with highly correlated demands.

14. Compute the correlation matrix[4] between the average daily demands of products. Plot the correlation matrix on a heatmap[5]. Make sure that you use a colorbar and a suitable colormap for better visibility.
15. Make a list of product couples such that (1) the correlation coefficient between their average daily demands are larger than or equal to the threshold value and (2) the first product belongs to a higher class than the second product. Report the number of product couples in your list.
16. Plot your correlation matrix once again but this time highlight the product couples you have determined with a visible mark.
17. Briefly comment on your observations based on the graphs you have produced.

Step 2—optimization

In the second step, you will address the allocation problem. To that end, you will devise simple heuristics as well as an integer program that yields an optimal solution.

1. For all methods, the most critical parameter will be the average daily profit loss per product—which will be realized in case the product cannot be stored in the current warehouse. Compute this value for each product.
2. Implement the heuristic where the ranking is done following the average daily profit loss for each product.
3. Implement the heuristic where the ranking is done following the ratio of the average daily profit loss to the number of pick-up boxes required for each product.
4. Devise an integer program that extends the 0-1 Knapsack problem for the purposes of the allocation problem on hand.
5. Solve the problem using all three approaches you have devised, and report the resulting total average daily profit losses on a small table.
6. Briefly comment on your findings.

Step 3—bonus: sensitivity analyses

This step is not mandatory. The students who successfully complete this step will equally share 6 bonus points.

In this last step, you will examine the impact of possible deviations from the available storage capacity and the anticipated average daily profit losses per product.

1. Conduct a sensitivity analysis considering storage capacities of the main warehouse ranging from 0% up to 200% of the actual capacity with increments of 5%, and iteratively solve the allocation problem using the three methods you have devised. For each of these experiments, record the total average daily profit losses.
2. Plot the daily total profit losses obtained by the integer programming model on a line graph for varying capacity levels and mark the total average daily profit loss for the current capacity level for better visibility.
3. Compute the difference between the total average daily profit losses obtained by the heuristic approaches and that of the integer programming model. Plot these on a single line graph for varying capacity levels.
4. Conduct a sensitivity analysis by varying the average daily profit losses of all products from 0% up to 200% of their currently anticipated values with increments of 5%, and iteratively solve the allocation problem using only the integer programming model. For each of these experiments, record the total average daily profit loss.
5. Plot the total average daily profit losses obtained by the integer programming model on a line graph for varying levels of daily profit losses per product and mark the value for the currently anticipated total average daily profit loss for better visibility.
6. Briefly comment on your observations based on the graphs you have produced.

---

[1] See e.g. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html
[2] See e.g. https://en.wikipedia.org/wiki/Knapsack_problem
[3] See e.g. https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.errorbar.html
[4] See e.g. https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html
[5] See e.g. https://seaborn.pydata.org/generated/seaborn.heatmap.html