Investigating a robot

Coding Exercise

Isaac Brocklesby

9 November 2021

Introduction

The aim of this report is to analyse a dataset containing information on a robot and provide information from it with potential business value. The structure of the report will revolve around exploring the dataset and describing it, providing some statistics, and speculating on potential future uses of the data.

The analysis will mainly be conducted in Python, and the folder containing this report is version controlled in the repository linked below:

https://github.com/DUCPS/robot_exercise

Initial exploration

The data provided came in an Excel file called “TEST_Report_09-27.xlsx”, which has been renamed to “data_v1.xlsx” for convenience. This file contains three sheets (four if you include the export sheet) which contain information on the processes of a particular robot. The file has been converted to csv, resulting in three relevant files: Dispenser_fails.csv, Dispenser_logs.csv, and ticket_items_manipulated.csv. Using pandas, these files have all been loaded into Python.

Of these three, Dispenser_logs.csv and ticket_items_manipulated.csv are the only files which can be investigated more deeply, as Dispenser_fails.csv only contains one row and five columns which don’t appear to provide much meaningful information. It is important to note that it does mention no fails, suggesting this robot may not have failed within the data set provided. This needs to be investigated further however.

Dispenser_logs.csv (d_logs) has 2536 rows and 24 columns. Of these columns, 17 are simply called “Unnamed [n]” and are entirely NA (also frequently called missing values or null). The seven remaining columns are named and described in Appendix A.

ticket_items_manipulated.csv (t_items) has 1462 rows and 22 columns. Of these, “location” and “ingredient_list” are entirely NA. All columns are named and described in Appendix B.

Statistics

How many meals were made per day? How many overall?

The table below shows the number of meals served each day along with the total number of meals served over the duration.

Date	Meals
2021-09-20	35
2021-09-21	37
2021-09-22	35
2021-09-23	30
2021-09-24	16
2021-09-27	43
Total	**196**

What was the most popular ingredient?

The most popular ingredient had an ingredient_id of “haredd565e5e-13a7-4379-9fc8-b9221fae01cf”. Without an additional data set mapping the ingredients to each id, the name of the ingredient cannot be known.

What was the most popular meal combination?

Note popularity has been assumed to mean the number of times a given set of ingredient_id’s or meal was ordered in the data set, with the idea that a more popular meal will be ordered more times in the time frame.

The following is a list of the ingredient_id’s for the most popular meal.

('2d7f72ae-7a99-4538-b178-eb280b16ec25',
'85e5083f-1736-4922-9bf7-1cf7da802d64',

'87941a51-fd64-4171-95b9-db76bd4465f0',

'bfe9f09a-4bfd-4504-a024-8b7cad14407d',

'dd565e5e-13a7-4379-9fc8-b9221fae01cf',

'e1d855b2-f0b5-45d8-a1f8-a10ba21eed20',

'e31c5ea0-00ba-4fb3-9bd6-185237ca1ceb')

How long did each meal take to make on average?

Assuming the meal starts being "made" when the ticket is first created (created_at) and the meal is finished when the final ingredient is dispensed (maximum value of finished_dispense) then the average time to make a meal is 01:16:05.447685244, or roughly 1 hour, 16 minutes, and 5 seconds.

Is there any relationship between ingredient and meal prep time? What about meal popularity and prep time?

Figure 1 illustrates the relationship between ingredient_id and preparation time in seconds corresponding to each ingredient. Many of the ingredients have a very similar plot shape. However there are 3 ingredients with significantly lower medians than the others, suggesting a possible

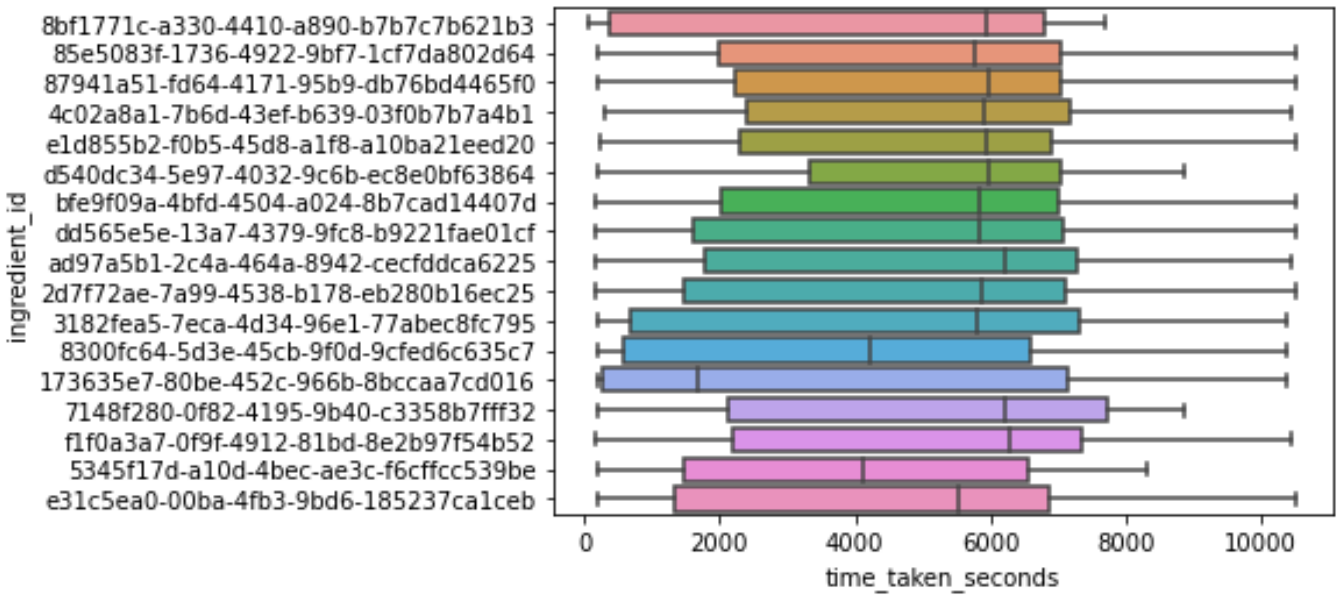

Figure 1: a box plot of the time taken in seconds for a meal to be made with respect to ingredient_id.

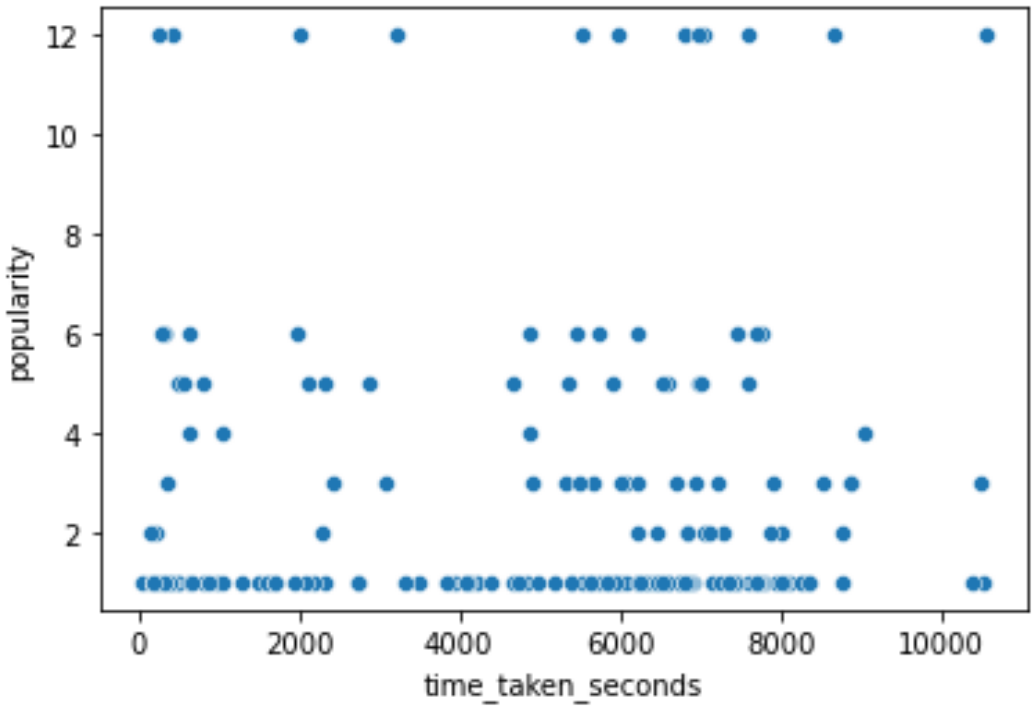

Figure 2: a scatter plot showing the relationship between "popularity" and taken taken in seconds for a meal to be made.

correlation between these ingredients and lower meal times. There are also 4 ingredients with lower maximum values. However as the upper quartile for these ingredients doesn't appear to also be significantly lower this is probably insignificant.

Figure 2 shows the relationship between popularity and time taken in seconds for a meal to be made. This plot does not show any significant trend or correlation, however a more thorough investigation could be done, perhaps with a more complete definition of what a popular meal is.

Recommendations for future investigation

Spending more time investigating the relationship between meal popularity and time taken to create the meal to see if more popular meals do take less time to create could provide valuable insight into what customers want from the robot.

Following on from this, if they a significant relationship between popularity and time is found, then the d_logs data set could be used to identify which actions performed by the robot take the longest time and as such which areas can be improved upon to reduce time and improve customer experience. It is also important to take into consideration here the frequency which an action is performed as a long action which is performed rarely may not be as impactful as a medium action performed frequently.

It would also be interesting to investigate the relationships between cost and ingredient popularity to identify whether customers prefer cheaper food.

As mentioned earlier, having a mapping between ingredient_id and ingredient name could help increase clarity of further investigations into the data set.

Finally, the difference between actual mass and expected mass could be calculated, and then averaged by ingredient to see if any ingredient in particular is worryingly far from expected mass. This could also be tied into the dispenser logs to see if a particular part of the robot is responsible.

Appendices

Appendix A:

Variable Name	Description
Unnamed: 0	An object (integer) between 0 and 638.
id	An unique id string for each (robot) job log.
created_at	Date and time of creation of job log.
updated_at	Date and time job log is updated. In this case it is the same as created_at.
log_type	The type of log. In this case it is always “MACHINE_LOG_ACTION”.
description	A description of the job performed.
meta_data	A JSON containing start time and finish time of the job, ticket id, and where the robot has moved from and too (broken down description of the job).

Appendix B:

Variable Name	Description
id	An id string (job). Note not unique.
state	The state of the job. In this case it is always “TICKETITEM_DONE”.
priority	An integer (intervals of 10) between 10 and 150. Assume higher priority has a lower number.
created_at	Date and time of job creation.
updated_at	Date and time job is updated. (In this case it is NOT the same as created_at).
started_at	Date and time the job is begun.
started_move	Date and time the move for the job begins.
finished_move	Date and time the move for the job finishes.
started_dispense	Date and time the ingredient begins dispensing.
finished_dispense	Date and time the ingredient is finished dispensing.
collected_at	Presumably when the bowl is retrieved after an ingredient is dispensed (all None).
ticket_item_failed	Boolean whether the ticket failed (all False).
ingredient_id	An id string (ingredient). Note not unique.

Variable Name	Description
ticket_id	An id string (ticket). Note not unique.
mass	The expected mass of the ingredient. Given as multiples of 1000, as such assumed to be mg.
actual_mass	The actual mass of the ingredient. As above, assumed to be mg.
item_cost	(Assumed to be) the cost of each ingredient (all 0.0).
location	Location the ticket was created (?) (all null).
ticket_id_counts	(Could not find a reasonable idea for what this column could be).
move_no_in_sequence	The position of the move/job in the sequence of the ticket (assumed).
priority_list	A list of the priorities in order for the machine for a ticket.
ingredient_list	A list of the ingredients used for a ticket (all null).