Predicting Garbage Collector's Invocation

For use by intended recipient only and not for further distribution

Garbage Collector Invocation

Overall objective of this challenge is to use provided data to predict when Garbage Collector gets triggered, and the amount of free memory available.

Background

Goldman Sachs is building a customized search engine for an internal text dataset. The search engine is launched on a container with a maximum memory limit. It takes some initial resources to launch the search engine on the container. Once the search engine is initialized, every time a search query is passed, it requests some additional resources for the search query to be served. The amount of resources requested by the search query would be a function of the frequency of the query keyword in the text corpus.

Sample search functionality:

Frequency of occurrence:

Search query	Frequency of occurrence in corpus
Token1	200
Token2	100
Token3	50
Token4	25

Queries:

Q1 = “Token1”: Resource requested would be proportional to frequency (200)

Q2 = “Token3”: Resource requested would be proportional to frequency (50)

So if the two sets of queries given above are performed, Q1 would require more resource allocation as compared to Q2.

Now, as queries are passed on to the search engine the used memory would keep on increasing after each query is served. Here in lies the importance of the garbage collector which would be invoked when the container is running out of free resources.

The objective of this exercise is two-fold:

- Given a set of queries with the initial configuration of the container, identify the queries at which the garbage collector would be invoked.
- Predict the free memory available after each query is served.

Input Format

Fields present in training set:

Variable	Description
initialUsedMemory(GB)	The amount of used memory for the container
initialFreeMemory(GB)	The amount of free memory before the query
query	The token which is searched in the query
gcRun	Boolean variable to identified if GC was triggered when the query is submitted
gcInitialMemory(GB)	The initial occupied memory of the java heap before the GC was run
gcFinalMemory(GB)	The final occupied memory of the java heap space after GC was run
gcTotalMemory(GB)	Total memory allocated for the java heap
userTimeGC(sec)	This is the amount of CPU time spent in user mode code outside the kernel and within the process
sysTimeGC(sec)	This is the amount of CPU time in the kernel within the process

Variable	Description
sysTimeCG(sec)	This is the amount of CPU time in the kernel within the process
realTimeCG(sec)	This is the real time visible to the user for the process
cpuTimeOverall(sec)	This is the overall CPU time taken for the search query to be served
finalUsedMemory(GB)	The amount of used memory after the query
finalFreeMemory(GB)	The amount of free memory after the query

Link to the training file with the above mentioned variables: Training File

Fields present in the testing set:

initialUsedMemory	initialFreeMemory	query	cpuTimeTaken	gcRun
4.24	2.77	token_11	0.40	?
?	?	token_76	0.12	?
?	?	token_53	0.24	?

Link to the testing file TestingFile

Constraints

The only constraint for the problem is that the sum of the used memory and the free memory would never exceed the total container memory which is limited to 9 GB.

Output Format

CSV File Format to be uploaded for evaluation: For the sequence of queries provided in the testing file the candidates would need to upload their prediction for the free memory and the gcRun in the following format:

serialNum	initialFreeMemory	gcRun
1	2.77	TRUE
2	2.45	FALSE

The candidate would be measured on:

- The accuracy of the gcRun prediction which would be evaluated based on the precision and recall of the model. More weightage would be given to the recall of the model as compared to the precision. Also, if the candidates are able to predict the GC trigger in the vicinity of the actual GC trigger in the test set, that would be counted in the calculation of the true positives in precision and recall calculation
- The RMSE error for the free memory prediction at each step
- Equal weightage would be given to both the factors mentioned above.

Model Doc & Source Code Submission

Addidtional two hours would be given to upload the source code and documentation.Please upload the following in challenge created for the documentation (**ML Documentation - Garbage Collector Invocation**):

- All source code files (quoting any references you may have used)
- A documentation of your solution. This can be in PDF, PPT, PPTX, DOC or DOCX formats. Do include the following aspects in your modeldoc:
  1. Assumptions
  2. Any mathematical simplifications/approximations
  3. Modelling choices and comments on appropriateness of answers
  4. Any plots/graphs which explain the process of your model/variable selection

Note:

Please ensure consistency between output and the methodology you explain in the model document If there are any additional ideas that you wish to include in the documentation which could not be implemented, please clearly mark them so.

Disclaimer: “For use by intended recipient only and not for further distribution”