TAZI Software Engineer (Backend Developer) Interview Project

Project Description

In this project, you'll simulate a continuous data source and do some windowed calculations on the fly in a single application.

Part 1: Continuous Data Source

You'll be given a CSV file, containing probability values of predictions from multiple machine learning models, which are trying solve a classification problem with two labels: A and B.

example:

id	given_label	model1_A	model1_B	model2_A	model2_B	...
1	A	0.3	0.7	0.2	0.8	...
2	B	0.21	0.79	0.1	0.9	...
3	B	0.0	1.0	0.25	0.75	...

You may use any system you prefer to store the data: Relational DBs, Object DBs or any other kind of storage (ElasticSearch etc...) are OK. You're expected to populate the data source gradually (100 instances per second for example) from the given CSV file to simulate data growing in time.

Calculations in Part 2 will be made using this growing data.

Part 2: Calculating Confusion Matrix

In this part, you'll be calculating **confusion matrices** and writing them back to the data storage.

example: (Act: Actual, Prd: Predicted)

Act/Prd	A	B
A	124	12
B	11	321

(Above can be read as, in 124 cases actual label was A and the predicted label was A, whereas in 12 cases actual label vas 12 but predicted label was B, etc.)

To get *predictedLabel*, first you need to combine results from all models, using weighted average.

weights will be provided similar to below:

model	weight
model1	0.5
model2	0.6
model3	0.7

You'll pick the label with the highest probability, that will be the predictedLabel.

You're expected to calculate confusion matrices with **sliding windows** of 1000 instances. (i.e. if data source contains 2000 instances, you'll calculate 1001 confusion matrices).

In other words, whenever there are enough results, you'll calculate a confusion matrix and **write it to the data storage.

- from instance 1 to instance 1000
- from instance 2 to instance 1001
- ...
- from instance 1001 to instance 2000

Notes

- You may use Scala, Java, C# or any other programming language.
- You should use a VCS (preferably git)
- You should write unit tests (for the crucial parts)
- You'll be continuously populating a data source (from CSV) and read from the data source at the same time.
- Please use separate threads for populating data source and doing the calculations, code is expected to be concurrent.

- You should **NOT** store *unnecessary* data in memory, as the original data source may be an **infinite** stream. (You should **NOT** load all the data into the memory at any stage.)
- Please carefully choose suitable data structures.
- You should try to minimize the number of queries, fetches, inserts etc...