



Dynamic Pricing Competition Test Environment



Purpose

Since the start of the competition we have seen many great and creative algorithm submissions for the dynamic pricing scenarios. But we also noted that there is some uncertainty about what we require from your code in order for it to compete in the competition. All of you should have received feedback by now on your most recent algorithm submission but in order for you to create better algorithms on your own we provide you with a small testing environment today.

*The testing environment will simulate prices of your competitors and give each participant **random** demand back. For this reason the testing environment should mainly be used to check if your algorithm performs reasonably fast and handles input/output correctly. The revenue and demand output you will see in the Excel file is solely for illustrative purposes since we can not reveal the underlying demand function in the testing environment.*

Requirements

In addition to any python packages that your code utilizes, you will also need to have the following packages in your python environment:

- pandas
- numpy

1. Download Zip File and Unpack on your local machine

In the email we sent you this pdf you will have also received a zip file with the testing environment. Download, Unpack and Save on the machine where you want to test your submission on.

2. Place your submission in the "dpc" subfolder

The next step is to place your code submission and all other files you might use in the dpc subfolder. You can just copy the content from your latest upload to the FTP server. By default we have placed the three new example codes into the dpc subfolder, but feel free to replace them with your own submissions.



projects > dynamic_pricing_competition > testing > participant_testing_environment

| Name | Date modified | Type | Size |
|----------------|------------------|-------------|-------|
| dpc | 12/06/2019 17:52 | File folder | |
| test_algorithm | 12/06/2019 17:40 | PY File | 18 KB |

First Level of Testing Environment you have received from us

projects > dynamic_pricing_competition > testing > participant_testing_environment > dpc

| Name | Date modified | Type | Size |
|--------------|------------------|---------|------|
| duopoly | 12/06/2019 17:15 | PY File | 7 KB |
| support_file | 12/06/2019 16:57 | PY File | 1 KB |

Place your submission file in the dpc subfolder (you can add support files)

3. Run Test Environment

Open your terminal or command line and navigate to your "participant_testing_environment" directory. In case you use any virtual environment, activate it and then run the test_algorithm.py file with the scenario you want to evaluate as the only argument.

For Duopoly: python test_algorithm.py duopoly

For Oligopoly: python test_algorithm.py oligopoly

For Dynamic: python test_algorithm.py dynamic

```
Select Command Prompt
C:\Users\Paul\git\extra-projects\dynamic_pricing_competition\testing\participant_testing_environment>
python test_algorithm.py duopoly
```

Running the test environment for the duopoly scenario



```
C:\Users\Paul\git\extra-projects\dynamic_pricing_competition\testing\participant_testing_environment>
python test_algorithm.py duopoly
2019-06-12 18:01:01,607 test_algorithm INFO      Importing the function for the duopoly pricing scena
rio.
2019-06-12 18:01:01,612 test_algorithm INFO      Running evaluation for the duopoly pricing scenario.

Currently in Selling Season: 1
-----
Currently in Selling Period: 50
-----
Currently in Selling Period: 100
-----
Finished with Selling Season: 1
-----
-----
-----
Currently in Selling Season: 2
-----
-----
Currently in Selling Period: 50
```

If your script can be imported you will see a similar screen to this

Depending on your algorithm the evaluation should be relatively fast (our example codes take about 10 seconds each to run). Once the evaluation is done you will find an Excel file in your "participant_testing_environment" folder with the results of each iteration, errors that occurred and the time performance of your code.

4. You can now open the Excel file

Open the Excel file to find 3 sheets:

- user_results
- errors
- runtimes



| | Selling_Seas | Selling_Period | Price | Competitor_has_capacity | Demand | Revenue | Error | Runtime_ms |
|----|--------------|----------------|-------------|-------------------------|--------|-------------|-------|------------|
| 0 | 1 | 1 | 64.1 | TRUE | 4 | 256.4 | FALSE | 0 |
| 1 | 1 | 2 | 30.1 | TRUE | 6 | 180.6 | FALSE | 0 |
| 2 | 1 | 3 | 52.5 | TRUE | 3 | 157.5 | FALSE | 0 |
| 3 | 1 | 4 | 67.7 | TRUE | 9 | 609.3 | FALSE | 0 |
| 4 | 1 | 5 | 70.9 | TRUE | 2 | 141.8 | FALSE | 0 |
| 5 | 1 | 6 | 74.9 | TRUE | 1 | 74.9 | FALSE | 0 |
| 6 | 1 | 7 | 70.6 | TRUE | 7 | 494.2 | FALSE | 0 |
| 7 | 1 | 8 | 42.3 | TRUE | 2 | 84.6 | FALSE | 0 |
| 8 | 1 | 9 | 65.5 | TRUE | 3 | 196.5 | FALSE | 0 |
| 9 | 1 | 10 | 44.6 | TRUE | 3 | 133.8 | FALSE | 0 |
| 10 | 1 | 11 | 47 | TRUE | 4 | 188 | FALSE | 0 |
| 11 | 1 | 12 | 56.33333333 | TRUE | 8 | 450.6666667 | FALSE | 0 |
| 12 | 1 | 13 | 55 | TRUE | 9 | 495 | FALSE | 0 |
| 13 | 1 | 14 | 44.33333333 | TRUE | 9 | 399 | FALSE | 0 |
| 14 | 1 | 15 | 34 | TRUE | 6 | 204 | FALSE | 0 |

If your code did not generate any errors the errors sheet will be empty (good job!). In the runtimes sheet you will find the average, minimum and maximum time (in ms) one pricing request of yours took.

Note that the runtimes are in milliseconds and will be rounded in the Excel file that is why you will see runtimes being equal to 0 when they are actually very small.

5. Change/Update your code and test again

In case your code generated errors, you can now fix the error and test again. Hopefully, it has been solved.

Done!

By now you should be able to test your code on your own. Based on the latest email you will also be able to check if your code runs fast enough to compete in the competition. However, it can be that the system that you are running the test environment on is more performant than our competition machines and therefore the runtime statistics are just to give you an idea of how fast your algorithms perform.

Please let us know in case you find any bugs/issues in the testing environment or have questions related to it.

We are looking forward to your submissions!