

esportsLABgg Counter-Strike Data Challenge

technical description of the task

1 We are proud to announce the Counter-Strike Data Challenge!

EsportsLAB is in the stage of dynamic growth, and we are in search of motivated people that will help us develop a **next-generation AI powered** tool dedicated to better development of skills in playing computer games thanks to a data-driven approach to training methodology.

Now, we like to announce **our data exploration challenge**, we propose a challenging data analysis task related to **Counter Strike: Global Offensive** FPS computer game (CS:GO). Our challenge is not limited to CS:GO players, just knowledge of Python programming and basic data analysis are sufficient. We state it as an open problem, we do not enforce any particular approach, and leave it to your creativity. Authors of the best solutions will be awarded **Razer pro e-gaming equipment**, and will be invited to work with us on the next-gen computer vision technology during **generously paid internships** at our startup. The internship will be remote, and the duration of three months, with possible extension.

We will award the three best solutions with

1. DeathAdder V2 Mouse + Kraken X headphones + Huntsman keyboard,
2. DeathAdder V2 Mouse + Huntsman keyboard,
3. Huntsman keyboard.

To be eligible for the reward you must be available and have an interest in participating in paid internships at esportsLABgg.

2 Task Details

Your main task is to write a script, which takes as input a CSV file with **CS:GO matches grenade throws data**. The provided datafiles contain features of selected grenades thrown by either team – terrorists (T) or counter-terrorists (CT) during matches on two major competitive maps (*de_inferno* and *de_mirage*). The features are described in Sec. 2.3 below.

Your task is to create a python binary classifier for labeling grenade throws as correct/incorrect. Your solution should be a single python script **which reads from the command-line input the name of a file with grenade features (can use e.g. 'argparse' library)**. Your script name should be *'classify.ph'*, if the test grenades filename is *'test.csv'*, we will execute your script as follows

```
python classify.py test.csv
```

The role of your script is to perform classification of each of the grenade (described using a single row of features) in the CSV input file, and **append the classification result (boolean value TRUE = correct throw/FALSE = incorrect throw) to the provided input file as an additional column 'RESULT' and modify it in-place**. See Section 2.3 for the detailed technical specification.

2.1 Technology

The task should be implemented in **Python programming language**. We are not restricting usage of any specialized python libraries, you are free to use **any available external library** you wish. If you train some parametrized machine learning model, please provide also the file with serialized parameters, and your solution

script should be loading the file. If the file is too large to send it over e-mail, you can provide it through a cloud service like Google Drive.

Your solution must be submitted before the deadline (Feb 1 2021, 11:59 PM CET) by email sent to counterchallenge@esportslab.gg.

2.2 Restrictions

You must work individually. It is forbidden to copy and paste any part of other participant solutions. Any violation of this rule and significant overlap of several solution scripts will result in the DQ of all submissions that are significantly similar or overlapping.

2.3 Problem Statement

GRENADE FEATURES Each grenade throw recorded in the input CSV file (single row) is described using the following set of features (the CSV contains also set of id values for our internal use: `demo_id`, `demo_round_id`, `round_start_tick`, `weapon_fire_id`)

- *team*: *T* – terrorists, *CT* – counter-terrorists;
- (*detonation_raw_x*, *detonation_raw_y*, *detonation_raw_z*): grenade detonation raw coordinates;
- (*throw_from_raw_x*, *throw_from_raw_y*, *throw_from_raw_z*): raw coordinates of the player when the grenade is being thrown;
- *throw_tick*: the exact tick (unit of game time, 128 ticks per second, counted from the beginning of the game), when the grenade is being thrown;
- *detonation_tick*: the exact tick, when the grenade is being detonated;
- *TYPE*: type of the grenade (*smoke*, *flashbang*, *molotov*);
- *map_name*: map on which the match was played (*de_inferno*, *de_mirage*);

LABELLED TRAINING DATA To design a successful classifier you may use training datasets. We provide two training datasets consisting of grenade throw data from two major competitive maps: *de_inferno* and *de_mirage*, each grenade throw is labeled using a boolean value. The meaning of the values in column 'LABEL' is either correct throw (*TRUE*) or incorrect throw (*FALSE*). We provide two sets of labeled training data:

- *train-grenades-de_inferno.csv*: features of **354** grenade throws on *de_inferno* map;
- *train-grenades-de_mirage.csv*: features of **370** grenade throws on *de_mirage* map;

2.4 Solution Script

Your solution python code should be provided as a python script *classify.py*. It will be executed as follows

```
python classify.py test.csv
```

where *test.csv* is a file with test grenade features data.

INPUT a CSV file containing test grenade features data.

OUTPUT Each grenade throw (single row per grenade throw) from the provided input CSV file should be classified as correct or incorrect by appending a boolean value (TRUE = correct throw / FALSE = incorrect throw) to each row, your classifications should be saved in the column labeled 'RESULT'. See an output example in Fig. 2.

EXAMPLE INPUT Below, we present an excerpt from a data-frame with features of several grenade throws executed on *de_inferno* map, where each row of the data-frame corresponds to a single grenade throw:

EXAMPLE OUTPUT Below, we present how the output from classifying grenade features in Fig. 1 should look like. The resulting classifications are the boolean values (TRUE = correct throw/ FALSE = incorrect throw) in the column 'RESULT', appended to the input data-frame.

team	detonation_ra	detonation_ra	detonation_ra	throw_from_ra	throw_from_ra	throw_from_ra	throw_ti	detonation	TYPE	map_name
CT	1479.26	1206.73	289.055	1294.665039	1186.002563	160.03125	11920	12144	flashban	de_inferno
CT	2300.1	454.382	140.338	1777.015625	-344.6234131	256.03125	13381	13605	flashban	de_inferno
T	2295.27	902.346	157.744	2210.061035	-404.5783386	88.00440216	6360	6696	smoke	de_inferno
T	978.253	2688.29	129.936	110.8179338	1569.625977	132.0136566	8896	9468	smoke	de_inferno
CT	1143.77	505.709	108.548	1486.039917	885.4845581	143.7997131	2934	3140	smoke	de_inferno
CT	1845.73	-204.199	258.031	2096.311768	-245.3414307	292.03125	8978	9188	smoke	de_inferno
T	1347.85	233.354	127.096	1353.629028	353.8656921	126.9070435	4663	4695	smoke	de_inferno
T	324.585	1860.63	128.528	464.9901428	2661.980225	162.3417511	6964	7484	smoke	de_inferno
CT	767.476	2311.68	137.392	1026.509521	2781.96875	128.0241089	11285	11517	smoke	de_inferno
T	49.8799	2585.69	160.031	875.6395264	2460.85083	145.03125	11083	11357	molotov	de_inferno
T	356.845	2285.72	381.049	1130.003052	2799.801514	128.0228271	3217	3443	flashban	de_inferno
CT	177.308	1029.44	86.0193	830.6674805	2332.754639	139.6670227	3053	3441	smoke	de_inferno
CT	1241.29	515.671	118.672	1401.106445	54.49951172	128.03125	7205	7413	smoke	de_inferno
CT	776.493	2209.55	135.916	-79.44869395	1330.03125	106.7088699	4635	4913	molotov	de_inferno
T	735.913	2837.36	127.064	110.8142014	1569.62793	132.0137177	14388	14988	smoke	de_inferno
T	1288.62	493.576	141.977	797.2390747	240.4300537	90.2346344	9485	9709	flashban	de_inferno
T	1063.53	2685.95	129.948	397.3203125	3077.920166	160.03125	8466	8724	smoke	de_inferno

Figure 1: Example data-frame with several grenade features. See above for description of the data columns.

team	detonation_ra	detonation_ra	detonation_ra	throw_from_ra	throw_from_ra	throw_from_ra	throw_ti	detonation	TYPE	map_name	RESULT
CT	1479.26	1206.73	289.055	1294.665039	1186.002563	160.03125	11920	12144	flashban	de_inferno	FALSE
CT	2300.1	454.382	140.338	1777.015625	-344.6234131	256.03125	13381	13605	flashban	de_inferno	TRUE
T	2295.27	902.346	157.744	2210.061035	-404.5783386	88.00440216	6360	6696	smoke	de_inferno	TRUE
T	978.253	2688.29	129.936	110.8179338	1569.625977	132.0136566	8896	9468	smoke	de_inferno	TRUE
CT	1143.77	505.709	108.548	1486.039917	885.4845581	143.7997131	2934	3140	smoke	de_inferno	TRUE
CT	1845.73	-204.199	258.031	2096.311768	-245.3414307	292.03125	8978	9188	smoke	de_inferno	TRUE
T	1347.85	233.354	127.096	1353.629028	353.8656921	126.9070435	4663	4695	smoke	de_inferno	FALSE
T	324.585	1860.63	128.528	464.9901428	2661.980225	162.3417511	6964	7484	smoke	de_inferno	TRUE
CT	767.476	2311.68	137.392	1026.509521	2781.96875	128.0241089	11285	11517	smoke	de_inferno	TRUE
T	49.8799	2585.69	160.031	875.6395264	2460.85083	145.03125	11083	11357	molotov	de_inferno	TRUE
T	356.845	2285.72	381.049	1130.003052	2799.801514	128.0228271	3217	3443	flashban	de_inferno	TRUE
CT	177.308	1029.44	86.0193	830.6674805	2332.754639	139.6670227	3053	3441	smoke	de_inferno	TRUE
CT	1241.29	515.671	118.672	1401.106445	54.49951172	128.03125	7205	7413	smoke	de_inferno	TRUE
CT	776.493	2209.55	135.916	-79.44869395	1330.03125	106.7088699	4635	4913	molotov	de_inferno	TRUE
T	735.913	2837.36	127.064	110.8142014	1569.62793	132.0137177	14388	14988	smoke	de_inferno	TRUE

Figure 2: Labeled grenade features from the example data-frame Fig. 1. Observe that the classifications are saved in the appended column 'RESULT'

2.5 Evaluation Criteria

We will evaluate your submission against secret set of test grenade throws from the same maps de_inferno and de_mirage that we took out from the training sets. To evaluate your submission we will use the following criteria

1. number of accurate test grenade throw classifications.
2. clarity of your solution.
3. code quality.

2.6 Submission

Your submission consisting of the python script and a pdf document describing your solution shall be submitted before the deadline (Feb 1 2021, 11:59 PM CET) by email sent to counterchallenge@esportslab.gg.

2.7 Further questions

Any questions related to the task you can send to the email address counterchallenge@esportslab.gg, or through our twitter account [@esportsLABgg](https://twitter.com/esportsLABgg).

Good luck!!!