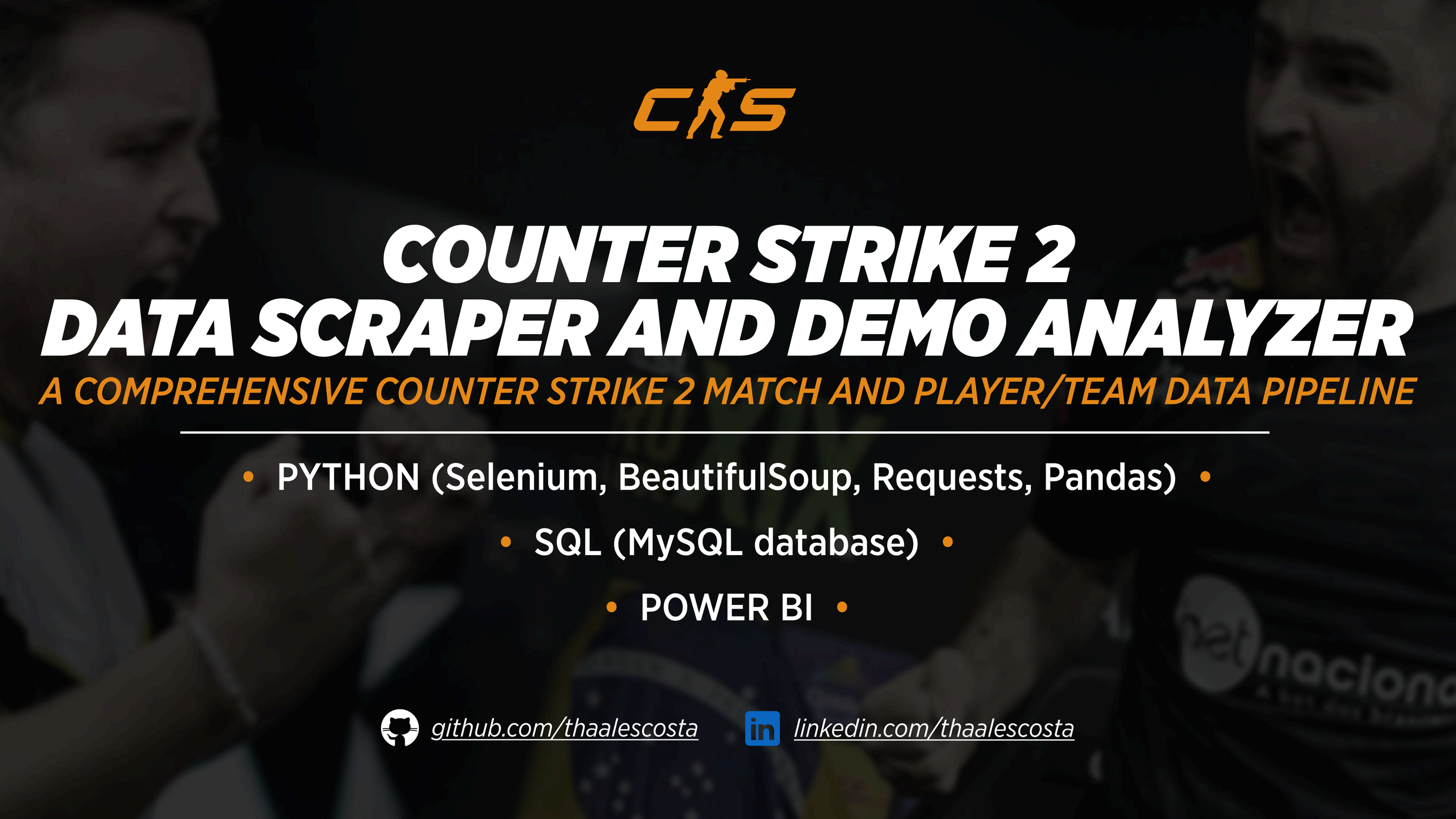


COUNTER STRIKE 2 DATA SCRAPER AND DEMO ANALYZER

A COMPREHENSIVE COUNTER STRIKE 2 MATCH AND PLAYER/TEAM DATA PIPELINE

- PYTHON (Selenium, BeautifulSoup, Requests, Pandas)
- SQL (MySQL database)
- POWER BI

github.com/thaalescosta *linkedin.com/thaalescosta*

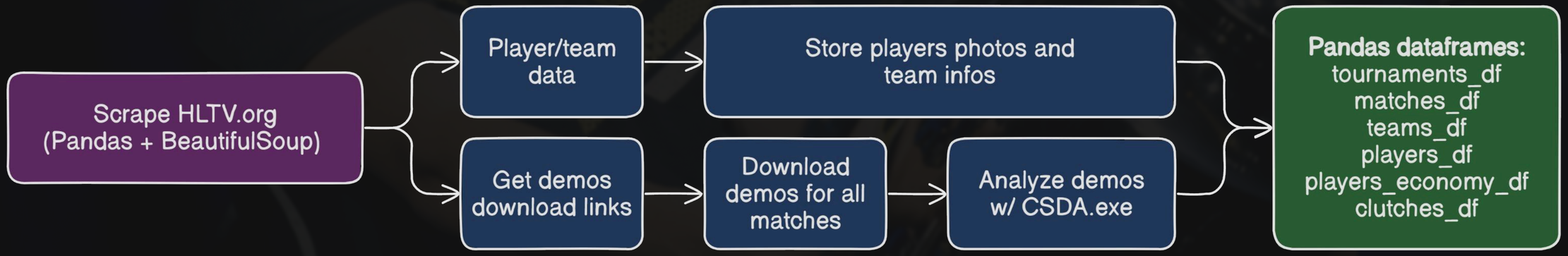
WHAT WAS I THINKING?

I was having a hard time finding good datasets of **Counter-Strike 2** matches on the internet. Every dataset I found on Google/Kaggle was either extremely outdated, way too simple, or both.

After some time searching around I figured **why not make my own** dataset with data I actually want to analyze?

And while I'm at it, why not make it more interesting and maybe create a database in MySQL and a Power BI dashboard?

So here we are.



*A representation of the logic behind **ExtractPlayerData.ipynb** and **ExtractMatchData.ipynb***

(not including the SQL and Power BI part)

BUILDING A PLAYERS AND TEAMS DATASET

You'll find two Jupyter Notebook in the GitHub repository: **ExtractPlayerData.ipynb** and **ExtractMatchData.ipynb**

ExtractPlayerData.ipynb scrapes player and team information from the Top N teams in the HLTV Ranking.

https://www.hltv.org/ranking/teams/2025/march/17

The user will have the option to choose how many of these teams the script will scrape by changing the variable ***topN:***

```
topN = 50 ## GET THE PLAYERS/TEAMS INFO FOR THE TOP X HLTV TEAMS ##

url = "https://www.hltv.org/ranking/teams/2025/march/3"
```

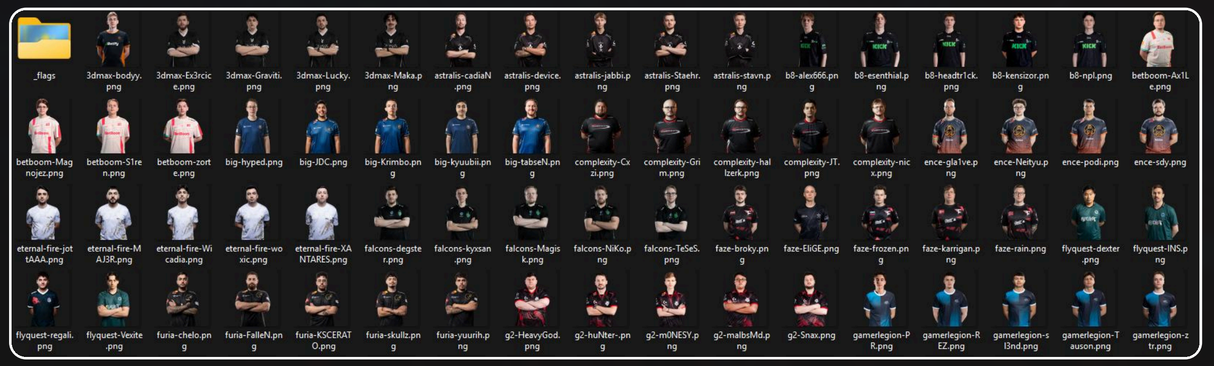
Since the rankings update periodically, the user should also update the variable ***url*** to the current date.

BUILDING A PLAYERS AND TEAMS DATASET

After **ExtractPlayerData.ipynb** is done scraping, the resulting Pandas dataframe will look like this:

	team_name	team_page	player_name	player_country	player_nick	player_hltv_url	photo_player_path	player_flag_path
0	Spirit	https://www.hltv.org/team/7020/spirit	Leonid 'chopper' Vishnyakov	Russia	chopper	https://www.hltv.org/player/7716/chopper	c:\Projects\cs2-match-and-player-scraper\Extra...	c:\Projects\cs2-match-and-player-scraper\Extra...
1	Spirit	https://www.hltv.org/team/7020/spirit	Dmitry 'sh1ro' Sokolov	Russia	sh1ro	https://www.hltv.org/player/16920/sh1ro	c:\Projects\cs2-match-and-player-scraper\Extra...	c:\Projects\cs2-match-and-player-scraper\Extra...
2	Spirit	https://www.hltv.org/team/7020/spirit	Boris 'magixx' Vorobiev	Russia	magixx	https://www.hltv.org/player/18317/magixx	c:\Projects\cs2-match-and-player-scraper\Extra...	c:\Projects\cs2-match-and-player-scraper\Extra...
3	Spirit	https://www.hltv.org/team/7020/spirit	Myroslav 'zont1x' Plakhotia	Ukraine	zont1x	https://www.hltv.org/player/20423/zont1x	c:\Projects\cs2-match-and-player-scraper\Extra...	c:\Projects\cs2-match-and-player-scraper\Extra...
4	Spirit	https://www.hltv.org/team/7020/spirit	Danil 'donk' Kryshkovets	Russia	donk	https://www.hltv.org/player/21167/donk	c:\Projects\cs2-match-and-player-scraper\Extra...	c:\Projects\cs2-match-and-player-scraper\Extra...

- **team_name:** Name of the player's team
- **team_page:** The URL to the player's team page on HLTV
- **player_name:** Player's full name and nametag in between
- **player_country:** Country of the respective player
- **player_nick:** Player's nametag
- **player_hltv_url:** Player's HLTV page URL
- **photo_player_path:** Path to the directory where the player PNG was saved
- **photo_flag_path:** Path to the directory where the player's respective flag PNG was saved



The user can find the PNGs inside **"\ExtractPlayerData\photo_players"**, which will be located inside the cloned folder

DOWNLOAD AND ANALYZE DEMOS

ExtractMatchData.ipynb was really tricky and it probably took me longer than it should have.

Before even thinking on analyzing demos, I had to modify the **CSDA.exe** (CS Demo Analyzer - link on repo) so it would generate only relevant data instead of a bunch of trash. With that out of the way, I needed to find a way to download every single demo for the tournaments of my (or your) choosing.

The list of tournaments scraped can be altered by adding/removing the event's ID to the **id_event** list shown below:

```
# ------------------------------------------------------------------
#   The ID of each event can be found in HLTV in either of the links below
#   For example:
#   https://www.hltv.org/events/7909/blast-bounty-2025-season-1-finals
#   https://www.hltv.org/results?event=7909
# ------------------------------------------------------------------

id_event = [ 8034, 7909, 7903, 7524, 7557, 7556, 7441, 7993, 7436, 7554, 7732 ]
```

There were a lot of steps involved until I was able to download every single demo, unpack their .zip files, analyze each .dem file and finally store everything I wanted into Pandas dataframes.

I won't show here how these dataframes look so this doesn't end up too long, but feel free to snoop around the .csv files in **"ExtractMatchData\tournaments_tables"**

DOWNLOAD AND ANALYZE DEMOS

In summary, these are the steps that **ExtractMatchData.ipynb** takes to finally do what I wanted:

1. Scrape all the matchups that happened in a tournament for all tournaments listed in **id_event**
2. Get a URL from the DOWNLOAD DEMO button from every matchup scraped
3. **U**se each URL to make a GET request and fetch the actual direct link to download the .zip file containing the demos
4. Unpack said .zip files
5. Run each .dem file through CSDA.exe
6. For every run, append the like .csv files to their respective dataframes (matches_df, kills_df, clutches_df, etc.)
7. Create (if it doesn't exist) a MySQL database called dbCS2
8. Create (if they don't exist) a table for each dataframe
9. Insert the data from each dataframe to their respective tables

There will be more steps to it, like making SQL queries to transform the tables into new ones that will actually be structured to fit my purpose - which is use them in Power BI to create a ton of visualizations.

This is all on my plans but I haven't had the time to do it... **yet.**