Datathon and Machine Learning Competition on Antisemitism

Agenda & Objectives

Agenda
1. Prerequisites & Registration
2. Challenge Description #1
3. Working with X's Advanced Search
4. Scraping with Bright Data
5. Pre-Processing the Data
6. Annotations Portal Walkthrough

Objectives: What You'll Learn Today:
- How to register and get started
- How the Annotation Portal works
- How to approach the ML challenge
- Where to find tools and datasets
- How to succeed as a team

Download this Presentation: Click the link

Prerequisites & Setup

Before You Start:

- A computer with internet access
- A Gmail account
- Twitter A X account
- Access to Google Colab (https://colab.research.google.com/)

Stop & Do Now:

- Register on the Annotation Portal: https://annotate.osome.iu.edu
- Check Github: https://github.com/AnnotationPortal/DatathonandHackathon.github.io/blob/main/README.md
- Read Challenge Description: https://github.com/damieh1/datathon_2025/blob/main/Datathon_Challenge.pdf

What is the Datathon?

Today's presentation will focus only on Part I of the Datathon competition.

🧠 **A Two-Part Challenge on Hate Speech Detection**

📍 **Part I: Annotation Competition (Workshop 13th & 20th July)**

- Scrape Social Media Posts from X
- Apply a Framework to Classify Antisemitic Content
- Label and Annotate a Dataset

📍 **Part II: Machine Learning Model (Workshop 27th July)**

- Train a model using annotated data
- Submit predictions & Evaluation Metrics
- Error Analysis

Instructions for Challenge #1

📍 **Subtasks include:**

1. Define your scraping focus (hashtags, user groups, topics) and document your rationale and potential biases.

2. Use the Bright Data interface to scrape at least 100 relevant user-generated posts from X.com.

3. Annotate your data using a structured definition of antisemitism and hate speech.

4. Prepare a X/Twitter dataset, and include a dataset report with label definitions, distribution information, and annotation rationale.

Instructions for Challenge #1

📍 **Deliverables for Challenge #1**

- Adapting and implementing an existing definition of antisemitism
- Reporting how the data was scraped and which guidelines were used to classify and annotate the data in a standardized way

Gain **+10 bonus points** by evaluating the consistency of your team's annotations using an inter-annotator agreement (IAA) metric.

This means:

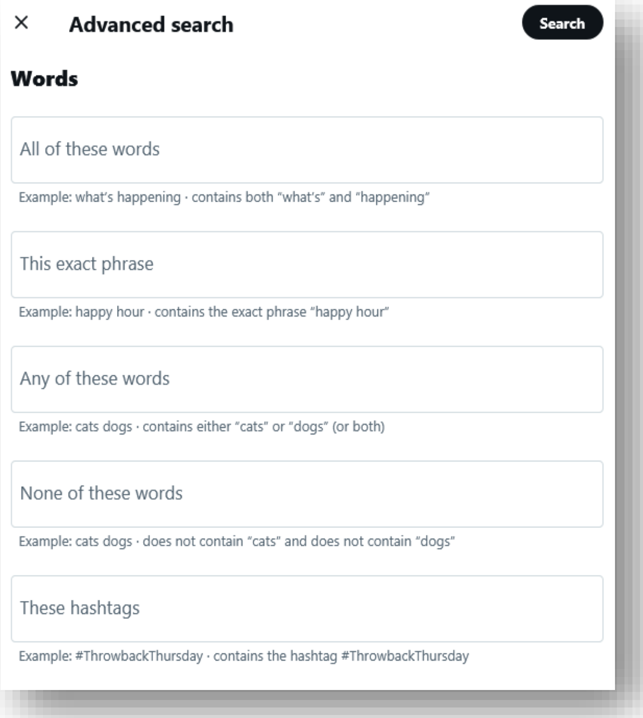
- Having at least two annotators label a shared subset of the data
- Calculating a formal agreement score, such as:
  - **Cohen's Kappa** (for binary or pairwise categorical annotation)
  - **Krippendorff's Alpha** (especially for multi-class or missing data)

Clearly report:

- Which subset was double-annotated
- Your score and a brief interpretation (e.g., "moderate agreement," "high agreement")

Working With X’s Advanced Search Function

Top of the pop-up menu



Words

All of these words

Example: what's happening · contains both "what's" and "happening"

This exact phrase

Example: happy hour · contains the exact phrase "happy hour"

Any of these words

Example: cats dogs · contains either "cats" or "dogs" (or both)

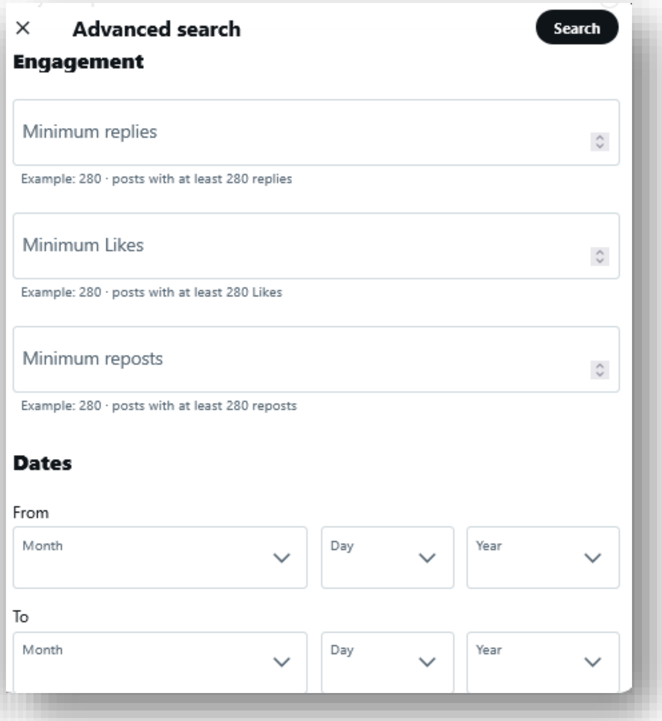
None of these words

Example: cats dogs · does not contain "cats" and does not contain "dogs"

These hashtags

Example: #ThrowbackThursday · contains the hashtag #ThrowbackThursday

Bottom of the pop-up menu



Engagement

Minimum replies

Example: 280 · posts with at least 280 replies

Minimum Likes

Example: 280 · posts with at least 280 Likes

Minimum reposts

Example: 280 · posts with at least 280 reposts

Dates

From

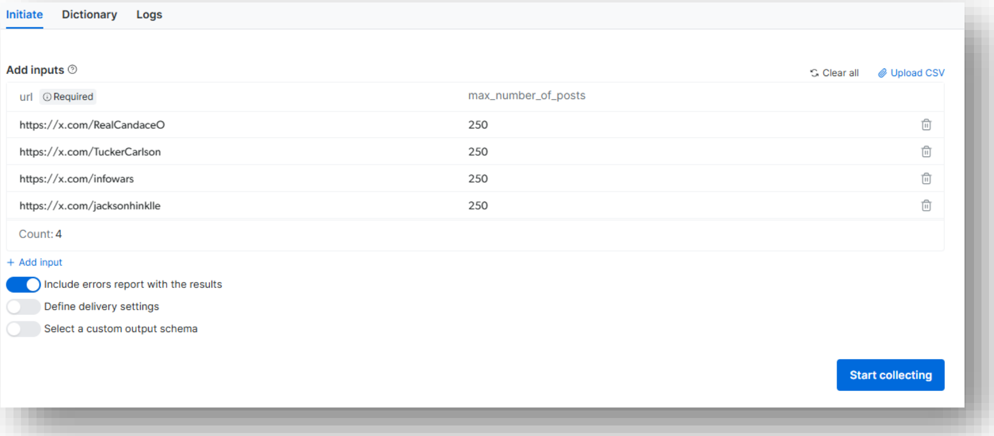
Month Day Year

To

Month Day Year

1. Go to: → X (Twitter) → https://x.com/search-advanced
2. Specify dates, e.g., May 8, 2024.
3. Click "Search.“
4. Select posts with a minimum of 200 views.
5. Go to user profiles and copy URLs to a spreadsheet.
6. Goal: Find a wide range of users who engage in online discourse.

Working With Bright Data



Add inputs

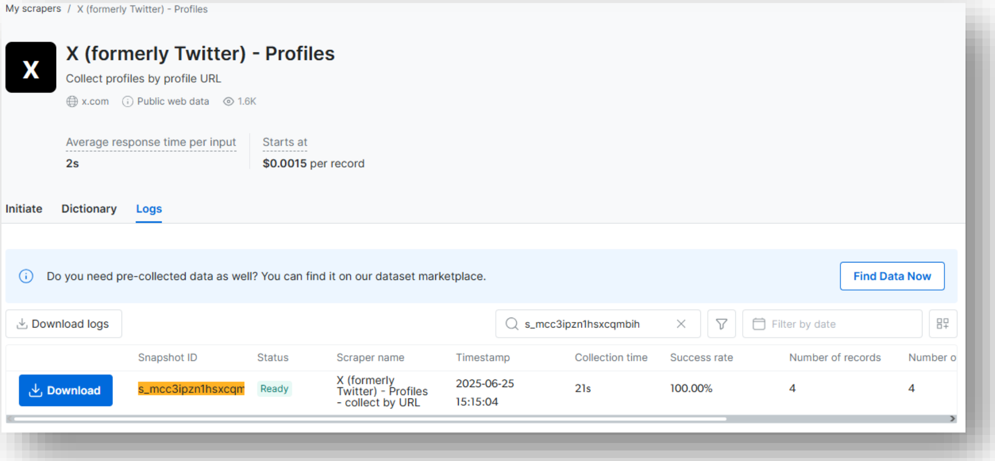
url Required	max_number_of_posts
https://x.com/RealCandaceO	250
https://x.com/TuckerCarlson	250
https://x.com/infowars	250
https://x.com/jacksonhinklle	250

Count: 4

● Include errors report with the results

○ Define delivery settings

○ Select a custom output schema



X (formerly Twitter) - Profiles

Collect profiles by profile URL

Average response time per input: 2s

Starts at: $0.0015 per record

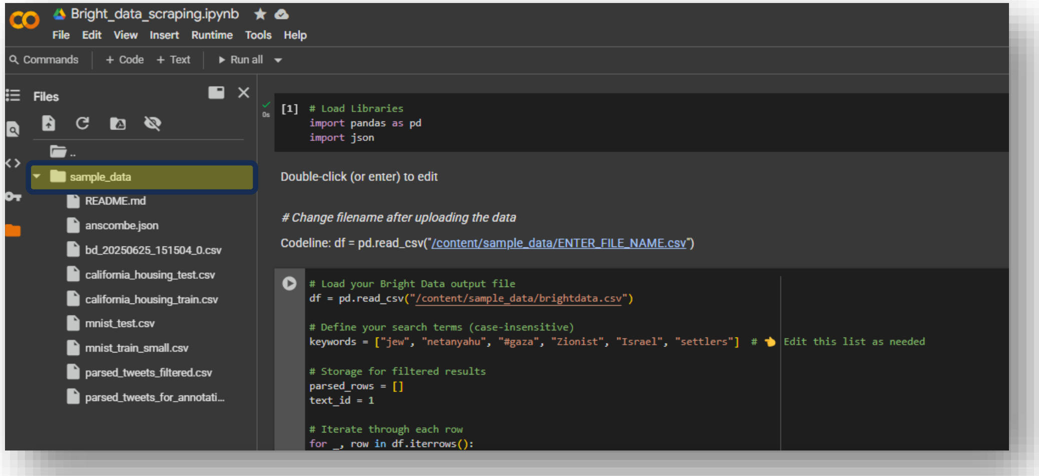
Do you need pre-collected data as well? You can find it on our dataset marketplace.

s_mcc3ipzn1hsxcqmbih

	Snapshot ID	Status	Scraper name	Timestamp	Collection time	Success rate	Number of records	Number o
Download	s_mcc3ipzn1hsxcqm	Ready	X (formerly Twitter) - Profiles - collect by URL	2025-06-25 15:15:04	21s	100.00%	4	4

1. Go to: Bright Data → Web Scrapes → X (Twitter) → Posts → Discover by URL → https://brightdata.com/cp/scrapers/no_code
2. Click: Add Inputs → https://x.com/RandomXUser
3. Specify Number of Posts → max. 250 per User
4. Start Collecting → Runes the Query
5. Download Output as .CSV

Working With Google Colab


Bright_data_scraping.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text ▶ Run all
Files
..
sample_data
README.md
anscombe.json
bd_20250625_151504_0.csv
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
parsed_tweets_filtered.csv
parsed_tweets_for_annotati...

```
[1] # Load Libraries
    import pandas as pd
    import json
```

Double-click (or enter) to edit

# Change filename after uploading the data

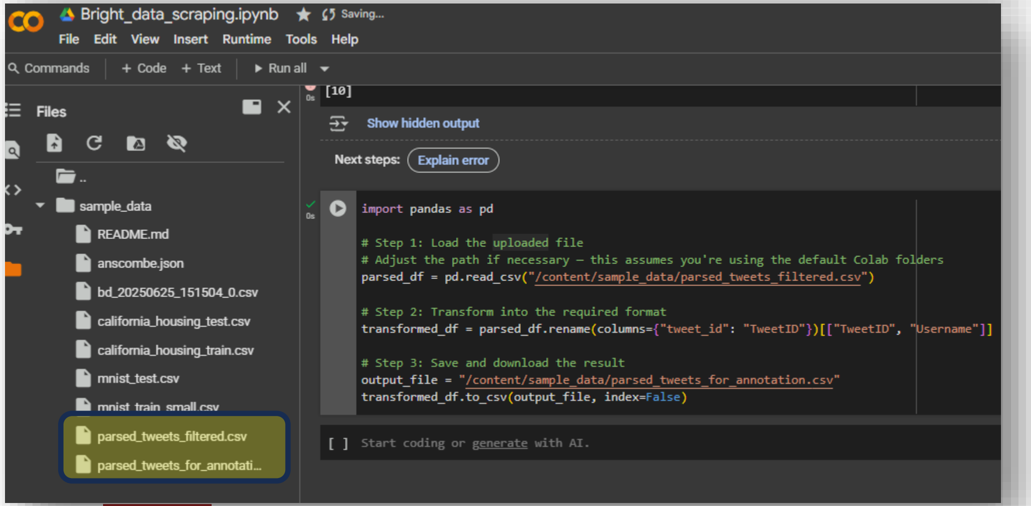
Codeline: df = pd.read_csv("/content/sample_data/ENTER_FILE_NAME.csv")

```
# Load your Bright Data output file
df = pd.read_csv("/content/sample_data/brightdata.csv")

# Define your search terms (case-insensitive)
keywords = ["jew", "netanyahu", "#gaza", "Zionist", "Israel", "settlers"]  # 👈 Edit this list as needed

# Storage for filtered results
parsed_rows = []
text_id = 1

# Iterate through each row
for _, row in df.iterrows():
```


Bright_data_scraping.ipynb
Saving...
File Edit View Insert Runtime Tools Help
Commands + Code + Text ▶ Run all
Files
..
sample_data
README.md
anscombe.json
bd_20250625_151504_0.csv
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
parsed_tweets_filtered.csv
parsed_tweets_for_annotati...
[10]
Show hidden output
Next steps:
Explain error

```
import pandas as pd

# Step 1: Load the uploaded file
# Adjust the path if necessary — this assumes you're using the default Colab folders
parsed_df = pd.read_csv("/content/sample_data/parsed_tweets_filtered.csv")

# Step 2: Transform into the required format
transformed_df = parsed_df.rename(columns={"tweet_id": "TweetID"})[["TweetID", "Username"]]

# Step 3: Save and download the result
output_file = "/content/sample_data/parsed_tweets_for_annotation.csv"
transformed_df.to_csv(output_file, index=False)
```

```
[ ] Start coding or generate with AI.
```

1. Go to: Google Colab → https://colab.research.google.com

2. Upload: Bright Data Output .CSV

3. Parse Data
   → Run Code on Colab

4. Prepare Data for Annotation Portal
   → Run Code on Colab

5. Download compatible .CSV Output

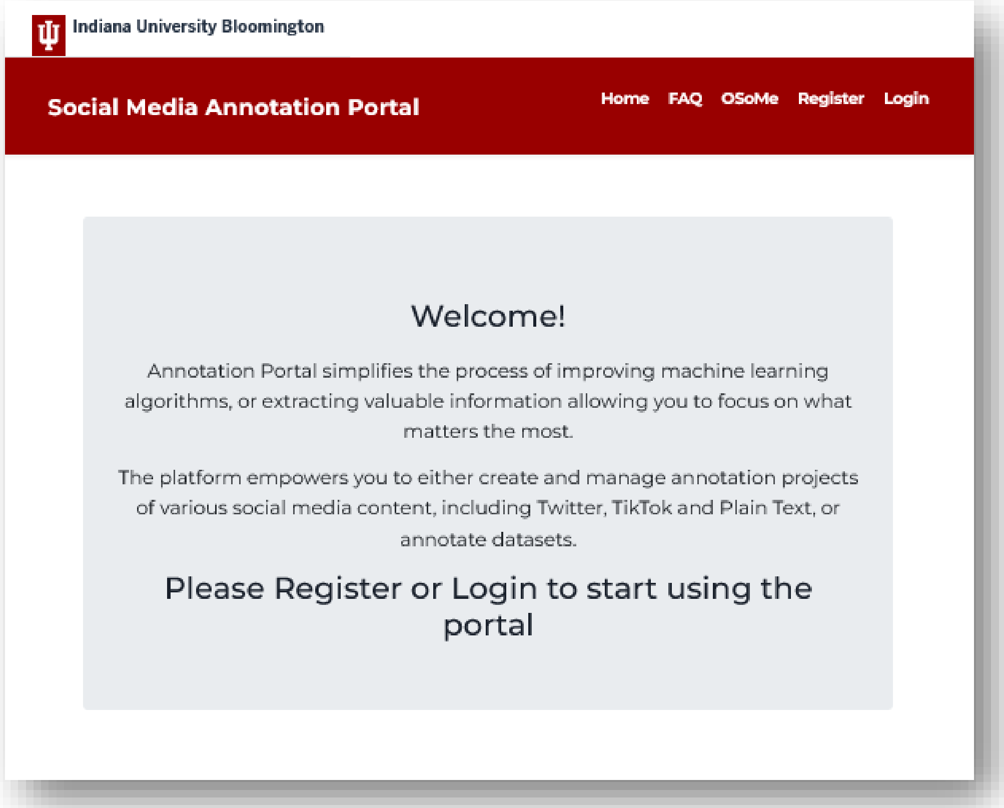
Click this → Open in Colab

Annotation Portal Walkthrough

How to Use the Portal:

1. Register and & Login
2. Appoint a Manager to Your Team
3. Create New Project
4. Create a Sample
5. Create Annotation Scheme and Questions
6. Important! Do not start annotating before the schema has been fully created.
7. Export when done

Demo Pause: Let’s annotate 1 example together!


Indiana University Bloomington
Social Media Annotation Portal
Home
FAQ
OSoMe
Register
Login

Welcome!

Annotation Portal simplifies the process of improving machine learning algorithms, or extracting valuable information allowing you to focus on what matters the most.

The platform empowers you to either create and manage annotation projects of various social media content, including Twitter, TikTok and Plain Text, or annotate datasets.

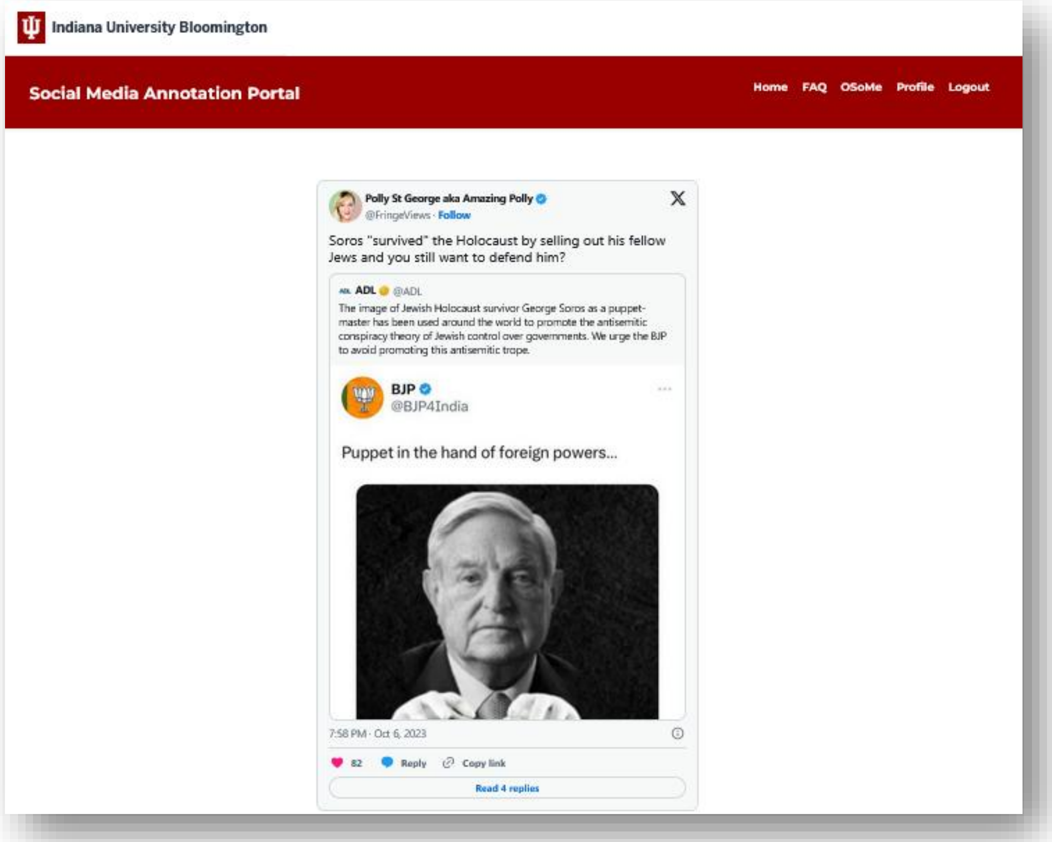
Please Register or Login to start using the portal

https://annotate.osome.iu.edu/

Annotation Best Practices

✅ **Tips for Quality Annotation:**

- Read carefully – context matters
- Be consistent with labels
- Use the comment field if unsure
- Check your export for errors



Soros "survived" the Holocaust by selling out his fellow Jews and you still want to defend him?

The image of Jewish Holocaust survivor George Soros as a puppet-master has been used around the world to promote the antisemitic conspiracy theory of Jewish control over governments. We urge the BJP to avoid promoting this antisemitic trope.

Puppet in the hand of foreign powers...

7:58 PM · Oct 6, 2023

Summary & Final Tips

Okay, before you jump in, let's make sure you crossed everything on the checklist.

📌 **Recap Checklist:**

[ ] Registered on the Portal, X
[ ] Setup Annotation Scheme
[ ] Registered Accessed GitHub and Colab
[ ] Explored ML challenge resources
[ ] Read Challenge Descriptions 1 and 2 thoroughly

📫 Questions? Reach out to your instructor or organizers.
💡 Remember: Collaboration, Curiosity & Critical Thinking!

Instructions for Challenge #2

📍 **Pre-Annotated Datasets:**

- Use our pre-annotated gold standard dataset to build and evaluate a hate speech detection system.
- Select and combine the following curated datasets:
  - Antisemitism on Twitter: A Dataset for Machine Learning and Text Analytics
  - Antisemitism on X: Trends in Counter-Speech and Israel-Related Discourse Before and After October 7
- Datasets are classified as either biased or non-biased

📍 **Subtasks include:**

1. Download the (Goldstandard/GroundTruth) datasets listed above
2. Use state-of-the-art transformer models to train and fine-tune a system to detect antisemitic content.
3. Evaluate your model:
   - Report precision, recall, F1-score, and display a confusion matrix.
   - List the hyperparameters used for training.
   - Conduct error analysis and provide qualitative examples, especially false positives.

ML Challenge & Resources

🤖 **Phase 2: Machine Learning Challenge**

- Download gold-standard dataset
- Use Colab or local notebook to train

🔧 **GitHub Resources Include:**

- NLP Tools and Code Examples
- Sample Datasets and Jupyter Notebooks

🛑 Participation in the third workshop on July 27 provides further information.

Instructions for Challenge #2

📍 **Deliverables for Challenge #2**

- Build and evaluate a transformer-based system for detecting antisemitic content using our pre-annotated gold standard datasets.
- Fine-tune a transformer model using the provided annotated datasets.

📍 **Code Submission**

- Clearly document the model used and include a summary of your training setup (train/test split, random seed, training strategy).
- Upload or link your training script(s), configuration files, and any preprocessing pipeline.

Earn up to **+10 bonus points** by testing your model on **unseen data**:

- Collect a small new sample of tweets by using Bright Data' scraping services
- Manually annotate 20–30 examples using the same label scheme
- Apply your trained model to this new set
- Report performance and reflect on how well the model generalizes

Thanks for your attention

Dr. Daniel Miehling

Visiting Assistant Professor
Borns Jewish Studies Program

Samerian Foundation Research Fellow
Institute for the Study of
Contemporary Antisemitism (ISCA)

Indiana University Bloomington
damieh@iu.edu