2025 Datathon Submission – Challenge #2

Team Name/Tag: MagenCode
Team Members: Giuilio Zuckerman, Asher Rosenfeld, Oriel Atias, Eliana Woolf

1. Project Overview

The goal of Challenge #2 was to train and evaluate a transformer-based system for detecting antisemitic content in social media posts, using the provided Gold Standard annotated datasets. Our approach focused on fine-tuning a BERT-family transformer model optimized for offensive language detection on Twitter data.

2. Dataset & Preprocessing

Dataset Used: GoldStandard2024.csv - Source: Provided by Datathon organizers - Format: Two columns → Text (tweet content) and Biased (binary label: 1 = antisemitic, 0 = not antisemitic) - Cleaning: Dropped rows with missing values, converted labels to integers - Train/test split: 80% train, 20% test, stratified by label Tokenization: Used AutoTokenizer from CardiffNLP's twitter-roberta-base-offensive with padding, truncation, and max_length=256

3. Model Choice & Rationale

We selected cardiffnlp/twitter-roberta-base-offensive as our base model because it is pretrained on large-scale Twitter/X data, optimized for offensive/hate speech detection, and provides strong language understanding for informal, slang-heavy text common in antisemitic tweets. We adapted it for binary classification: 0 = Not antisemitic, 1 = Antisemitic.

4. Training Setup

Parameter	Value
Epochs	3
Train Batch Size	8
Eval Batch Size	8
Learning Rate	2e-5 (linear decay)
Weight Decay	0.01
Optimizer	AdamW
Seed	42
Evaluation Strategy	Epoch
Save Strategy	Epoch
Max Sequence Length	256 tokens
Hardware	Tesla T4 GPU (Colab)

5. Evaluation Metrics

Metric	Value
Accuracy	90.15%
F1 Score	0.899
Eval Loss	0.397
Train Loss	0.183 (final)

6. Error Analysis

False Positives: Tweets containing offensive keywords but not antisemitic in context. False Negatives: Subtle antisemitism lacking explicit slurs, often using sarcasm or coded language.

7. Obstacles & Resolutions

- Dependency conflicts in Colab resolved by reinstalling compatible versions of transformers and accelerate. - Tokenizer OverflowError fixed by setting max_length=256. - Adjusted batch size to fit GPU memory constraints.

8. Reproducibility

Code and dependencies are provided in requirements.txt and the accompanying notebook. Model artifacts are saved in Google Drive. Training is reproducible in ~6 minutes on a Tesla T4 GPU.

9. Reflections

This task underscored the value of domain-specific pretrained models, the difficulty of detecting subtle hate speech, and the impact of preprocessing on performance. Even a small transformer model, carefully fine-tuned, achieved strong results.