TML Assignment 1

The core idea of this attack is to extract as much information as possible about the target (or victim) model. All our solutions were designed around this goal. The improvements were made progressively, rather than in a single step.

As of the latest update, we achieved a **TPR@FPR=0.05 of 0.144** and an **AUC of 0.655** on the scoreboard.

Initial Approach

We began by training **5 shadow models**, extracting only the **confidence scores** (i.e., the maximum softmax output) to train the attack model. Next, we **trained the attack model directly on the full 44-dimensional softmax output vector** from the target model evaluated on public data, which provided a modest improvement in TPR@FPR=0.05.

We then experimented with **adding richer features**—using the full class probability vector instead of just the confidence—and tested several classifiers. This led to further gains, with a TPR@FPR of **0.073** and an AUC of **0.557**.

Shadow Model Scaling and Feature Engineering

We scaled up to **20 shadow models** and used the **top-5 softmax probabilities** as features for the attack model. This resulted in a TPR@FPR of **0.064**. A slightly better performance (TPR@FPR = **0.09**) was obtained by **combining features from the shadow models and the target model** (evaluated on public data).

The most significant improvement came when we engineered and included additional features such as:

- **Loss**
- **Margin**
- **Entropy**
- **True label**
- **Full softmax probability vector**

This boosted our metrics to a **TPR@FPR=0.1433** and an **AUC=0.6603**.

Data Splitting Strategies

Our earlier data splitting strategy used a **fixed training size** of 1000 samples(i.e members from the classifier's perspective) for each shadow model (with 500 and 1000 for eval and test sets/non-members respectively), randomly drawn without replacement from a 20,000-sample public dataset. Each split was independent. The train, eval and test are disjoint from each other.

In the **revised strategy**, we increased to **30 shadow models** and adopted a **linearly increasing training size** ranging from **300 to 3000 samples per model**, to better capture variation in model behavior. The eval and test sizes were fixed at 500 and 1000 respectively. This change allowed us to simulate a wider diversity in model training conditions. Using this improved configuration, we achieved a final score of **TPR@FPR=0.144** and **AUC=0.655**.

Additional Experiments

We also explored data augmentation in an attempt to mimic the target model's training distribution, but this did not yield meaningful improvements.

While increasing the number of shadow models contributed to improved performance, **further improvements could be achieved through other strategies**—for instance, **training multiple attack models specialized per class rather than a single global one**. This class-wise modeling might help capture finer-grained membership signals, especially in high-class-count settings like ours (44 classes).

We used the yonsei-sslab/MIA repository as a **framework to streamline experimentation**. This helped manage dataset preparation, model training, and evaluation with modularity.

- `make_data.py:` Handles the **generation and preprocessing of datasets** for shadow and target models.

- `trainer.py:` Implements the **training and evaluation logic** for shadow, target, and attack models.