TML Assignment # 1 - MIA Report

Name	ID
Rithvika Pervala	7072443
Bushra Ashfaque	7070555

We implemented a Membership Inference Attack (MIA) against a ResNet18 model using a combination of two ideas

- Feature extraction of entropy, loss, confidence scores from Public and Private Data
- Computing LiRA scores on Public and Private Data

These two ideas are then combined into a comprehensive predictors on which our ensemble attack with three different classifiers - Random Forests, LightGBM, and MLP Classifier are trained, validated and tested. In the end MLP Classifier turned out to give our best results as follows.

```
{'TPR@FPR=0.05': 0.145, 'AUC': 0.6548955000000001}
```

Data Accessible to the Adversary

The attacker was provided with:

- A ResNet18 model trained on an undisclosed dataset
- Public dataset (20,000 samples) with membership labels (1=member, 0=non-member)
- Private dataset (20,000 samples) with unknown membership status
- Normalization parameters: `mean=[0.2980, 0.2962, 0.2987], std=[0.2886, 0.2875, 0.2889]`
  - Using these parameters both the Public Dataset - `public_data` and the Private dataset - `private_data` have been normalized

Feature Extraction

Using the hybrid function `extract_mia_lira_features(dataset, is_public)` extract the below 10 discriminative features - both from `public_data` and `private_data` . While some may have high correlation, they still explained different parts of the data. These Features will become part of the Predictors used to train the Ensemble models. The Features are as follows.

Softmax Probabilities (`soft_max_probs`)

- The predicted probabilities for each class by the ResNet18 Model.
- **Purpose -** These probabilities give a full picture of the model's belief for each class and are often overconfident for training member samples - hence making it useful for MIA.

Cross-Entropy Loss (`sample_loss`)

- The per-sample loss between predicted logits and true labels.
- **Purpose -** Training samples typically have lower loss values due to overfitting - test (non-member) samples tend to have higher loss.

Class Entropy (`class_entropy`)

- Entropy of the softmax probabilities for each class.
- **Purpose -** Lower entropy often implies more confident predictions, which is more typical for training members.

Modified / Adjusted Entropy (`adjusted_entropy`)

- Modified entropy is a classic MIA feature used based on how well the model predicts the true label. High for uncertain, incorrect predictions; useful for distinguishing members/non-members.

Top-2 Confidence Margin (`top2_conf_margin`)

- Difference between the top-1 and top-2 softmax probabilities.
- **Purpose -** Higher margin = model is more confident in its top class. Members often have larger margins due to better separation.

True Class Probability (`true_class_prob`)

- Softmax probability assigned to the correct class.
- **Purpose -** One of the most effective features in MIAs—training members typically have higher confidence for the correct class.

Correctness of Prediction (`correctness`)

- 1 if predicted label matches true label, else 0.
- **Purpose -** Models are more accurate on their training data (members), so this is a useful binary feature.

Max Logit (`max_logit`)

- Highest value among the raw logits.
- **Purpose -** High max logit often correlates with high confidence; members may have stronger activations due to overfitting.

True Class Logit (`true_logit`)

- Logit value for the correct class.
- **Purpose -** Higher values suggest the model strongly favors the true label—likely for member samples.

Gradient Proxy (`gradient_proxy`)

- **Purpose -** Approximate sensitivity of the loss w.r.t. the true logit; attempts to proxy gradient information without backpropagation. Potentially helps detect overfitting signals.

LiRA Scores

- LiRA Scores are calculated using `compute_lira_scores(dataset, conf_member, conf_non_member)` which takes in the the true class probabilities of members and non-members - `conf_member` and `conf_non_member` respectively.
- The values of `conf_member` and `conf_non_member` are calculated from the Public Dataset in the above function `extract_mia_lira_features()` while extracted features if Public Dataset.
- The function `compute_lira_scores(dataset, conf_member, conf_non_member)` now can compute LiRA scores for both Public and Private datasets which will then be used as one of the Predictors for Ensemble Models

---

Ensemble Attack with - Features Extracted + LiRA Scores

- Once both the Features and LiRA Scores are computed, they are combined to make 11 (10 Features + 1 LiRA Score) `combined_features` predictors using which the Ensemble models are trained, validated and tested.
- The configuration for the 3 Attack Models are defined in the function `ensemble_attack(X, y)` as follows

```
models = {
      "RF": RandomForestClassifier(
         n_estimators=100, max_depth=15, min_samples_split=5,
         random_state=42, n_jobs=-1
      ),
      "LightGBM": LGBMClassifier(
         n_estimators=200, max_depth=15, learning_rate=0.05,
         random_state=42, n_jobs=-1
      ),
      "MLP": MLPClassifier(
         hidden_layer_sizes=(64, 32), max_iter=100, random_state=42
      )
   }
```

- The models are then trained in the function `train_mia_lira_model(X, y, model)` where the Public data predictors and labels are split into training and validation set.
- The model is fitted and calibrated using `CalibratedClassifierCV` using which we estimate the AUC and TPR@FPR scores. These are the estimated scores as follows.

```
🔹 Training: RF
AUC: 0.6715
TPR@FPR=0.05: 0.1454

🔹 Training: LightGBM
AUC: 0.6660
TPR@FPR=0.05: 0.1639

🔹 Training: MLP
AUC: 0.6591
TPR@FPR=0.05: 0.1514
```

- Once the models are fitted, they are then tested on the Private Data Predictors and the statistics of the membership scores generated are as follows

```
Attack Results : RF
  Generated scores for 20000 samples
  Score range: [0.0073, 0.6209]
  Score mean: 0.4450
  Score std: 0.1227

Attack Results : LightGBM
  Generated scores for 20000 samples
  Score range: [0.0007, 0.8410]
  Score mean: 0.3095
  Score std: 0.1406

Attack Results : MLP
  Generated scores for 20000 samples
  Score range: [0.0000, 0.9997]
  Score mean: 0.5090
  Score std: 0.1849
```

- After submitted the scores from all the 3 models, MLP Classifier has given out the best results.

```
Submission Results : MLP
{'TPR@FPR=0.05': 0.145, 'AUC': 0.6548955000000001}

Submission Results : RF
{'TPR@FPR=0.05': 0.09166666666666666, 'AUC': 0.6343475555555556}

Submission Results : LightGBM
{'TPR@FPR=0.05': 0.089, 'AUC': 0.6221964444444444}
```

Why use isotonic calibration instead of sigmoid?

- Isotonic calibration preserved relative score ordering while optimizing low-FPR regions.

Files and their descriptions

Files

- `A1_12_Ensemble.ipynb` - Jupyter Notebook containing the MIA Attack code
- `submission_MLP.csv` - Membership Scores predicted by the MLP Classifier

Core Functions

- `extract_mia_lira_features()` - Function using which Features (Public, Private Datasets) and the `conf_member` , `conf_non_member` (Public Dataset) are extracted.
- `compute_lira_scores(dataset, conf_member, conf_non_member)` - Function to compute the LiRA scores of both Public and Private Datasets using `conf_member` , `conf_non_member` computed from Public Dataset.
- `train_mia_lira_model(X, y, model)` - Function to train and calibrate each model.
- `ensemble_attack(X, y)` - Function that defines the ensemble models and preprocesses (Standard Scaling, Changing Data Type) the data.