Trustworthy Machine Learning, SS 2025
Prof. Dr. Adam Dziedzic and Prof. Dr. Franziska Boenisch
Assignment Sheet #1: Membership Inference Attack

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY
UNIVERSITÄT DES SAARLANDES

Teaching Assistant: Nupur Kulkarni

Deadline: Wednesday, May 28, 2025

Task: Implement a membership inference attack and achieve the highest attack success (TPR@FPR=0.05) and area under the curve (AUC).

The goal of membership inference attacks is to disclose whether a data point was part of the dataset the model was trained on. In this assignment, you are in the *attacker's shoes,* and your goal is to launch a successful membership inference attack. Before launching this attack, it would be great to revise what a membership inference attack is from the lecture - Privacy I.

As an attacker, you get access to:

- Resnet18 model trained on an *undisclosed* dataset
- Public dataset that contains the ids, images, class labels, and membership information (1 == is member, 0 == not a member). Members are data points that were used in the training dataset to train the model, and non-members were not.
- Private dataset that contains the ids, images, and class labels, with membership set to None, which means *the membership is unknown*. Note that some of the data points in this dataset were used to train the model (members) and some were not (non-members).

Additional information that you should know

The model has been trained using **normalized** data. You also have access to the normalized parameters. These normalized parameters are *per channel*.

```
mean = [0.2980, 0.2962, 0.2987]
std = [0.2886, 0.2875, 0.2889]
```

What you don't have access to:

- The underlying training dataset used to train Resnet18 model
- Membership information of the private dataset

What's your task in this assignment

- Your goal is to classify each sample from the private dataset as a member or non-member, with a membership confidence score. You are *not* supposed to submit membership scores as 0 (non-member) or 1 (member). We expect you to submit ***continuous* membership confidence scores**.

Note - Membership confidence score reflects how likely it is that a given sample was part of the training set used to train the model.

Your starting point

To start off, we give you a coding template that will help you load the trained model and the public and private datasets. The coding template also includes an example submission code that we will discuss next.

How to submit your results?

- We will provide a unique *team number* and *token* to every team in the course. This unique token for every team can be accessed on the Personal Status tab of the course on CISPA CMS. You are supposed to use only this token to submit your scores to our server. If you did not receive this team-specific information, please contact us.

- You are supposed to submit your scores using the code below, also provided in the code template. Remember to replace "score": np.random.randn(len(data.ids)) with the numpy array of the membership confidence scores that you obtained for the data points from the private dataset. Also, replace "TOKEN" with the token provided to you *with the double quotes*.

```
#### EXAMPLE SUBMISSION

df = pd.DataFrame(
  {
     "ids": data.ids,
     "score": np.random.randn(len(data.ids)),
  }
)
df.to_csv("test.csv", index=None)
response = requests.post("http://34.122.51.94:9090/mia", files={"file": open("test.csv", "rb")},
headers={"token": "TOKEN"})
print(response.json())
```

Points to note regarding submission:

- You can only submit a CSV with scores **every hour**. If your submission is malformed or wrong, you should get a comprehensive error message. However, even if you fail, you still get a 1h cooldown.
- The evaluation server can crash catastrophically at any moment. It is a good idea to store your solutions (csv files) somewhere safe so that if that happens, you can resubmit.

Scoreboard

- You can access the scoreboard for this task here http://34.122.51.94:9090/score_board in the MIA TPR tab. This will help you to compare your solutions with other teams and see where you stand.
- The beginning value of your "score" is always worse than that of a random guess submission. Don't get so nervous if you see -1000 TPR@FPR at the beginning. By running the code example above with "score": np.random.randn(len(data.ids)) you should get ~0.05 TPR@FPR=0.05 (see the "debug" team on the score board) and AUC of ~0.50.
- The scoreboard shows the best results only. As an output to your request, you get back the TPR and AUC for your current submission. If it's lower than the one stored in the scoreboard state, the scoreboard is not updated (as well as the score).
- This scoreboard only shows the results on 30% of the samples from your submission, so it is an intermediate scoreboard.
- The evaluation is split into two sets: immediate and final (30:70). The score you will see before the deadline is for the immediate set; the final (now hidden) one will be revealed once the deadline for the assignment passes.

Read the following instructions carefully

- Create a github repository TML25_A1_YourTeamNumber. For example, if you are in Team 13, your repository should be named TML25_A1_13
- Upload your code files to the repository
- Create a README.md in your GitHub repository for this assignment. Describe how you solved the assignment and point us to the most important files and pieces of code. The final grade will depend heavily on how you describe your implementation.
- If it is private, invite Adam, Franziska, and Nupur (adam-dziedzic, fraboeni, nupur412) to the repository. In case the invite has expired, please resend the invites.
- For your final submission, please make a tag (click on tags, "create new release", create it).
- **Submit a zip file with your report and the whole repository.** Additionally, provide us with a link to the tagged version of the online Test in CMS under Assignment X (X is the number of the current assignment) before the deadline.
- You are only supposed to make *one submission* per group.