ML4N - Group Project 7
Analysing Adversarial Attacks on Tabular Data Classifiers

Clarifications for this project can be asked to Gabriele Ciravegna: gabriele.ciravegna@polito.it

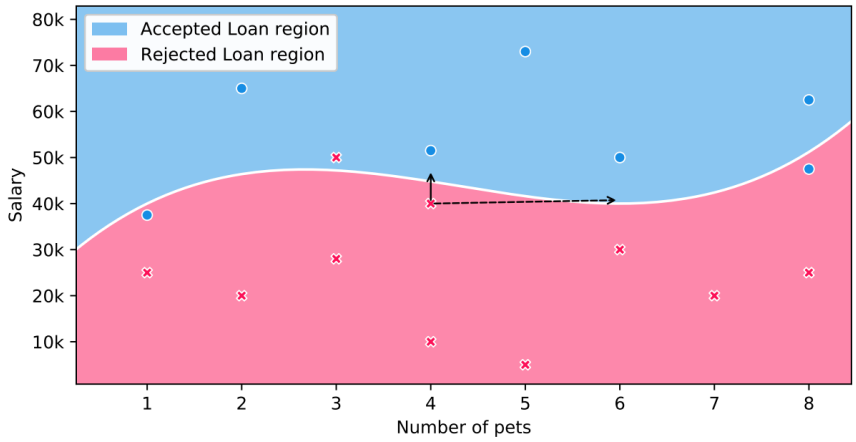

Figure 1: Examples of Adversarial Attack on a classifier trained on the German Credit Dataset.

Guaranteeing the security of Machine Learning (ML) classifiers is a crucial priority of all institutions employing automatic decision-making systems, to protect their businesses against cyberattacks and fraudulent attempts. Adversarial attacks are novel techniques that, other being proven to be effective to fool image classification models, can also be applied to tabular data. Adversarial attacks aim at producing adversarial examples, in other words, slightly modified inputs that induce the Artificial Intelligence (AI) system to return incorrect outputs that are advantageous for the attacker. To illustrate the threat of adversarial attacks in a tabular context, we consider the scenario where a bank customer applies for a loan. A machine learning model is used to make a decision regarding the acceptance of the application based on customer provided information (incomes, age, etc.). The model advises the bank to reject the application of our customer. However, he is determined to get the loan by filling false information to mislead the model. The key for this attack to succeed is its imperceptibility: the application should remain credible and relevant, in coherence with the model's prediction. As shown in Figure 1, adversarial attack must create adversarial samples x_{adv} that are sufficiently close to the real data x, i.e., $|x - x_{adv}|_2 < \epsilon$, where ϵ is that maximum distance to which an attack is considered credible.

This assignment challenges students to explore the impact of adversarial attacks on tabular data classifiers, specifically within the context of loan risk prediction using the German Credit Risk dataset. The task involves a step-by-step investigation structured around essential components.

1 Data exploration and preprocessing

1.1 Dataset Acquisition

Begin by downloading the German Credit Risk dataset[1].

1.2 Data Preprocessing

As a preprocessing step, convert categorical features to numerical features, e.g., by means of a one-hot encoding, and apply a normalization to map all features between $[0, 1]$. Assign labels to the dataset indicating the true loan attribution outcomes for supervised learning. Clearly define categories or classes representing loan approval or denial. Additionally, reserve 20% of the dataset for final testing and assessment of the robustness of the models.

1.3 Exploratory Data Analysis

Delve into the dataset's patterns using various visualization techniques, such as histograms of average values, scatter plots of most interesting features and correlation matrices. Provide insights into underlying patterns and, most importantly, identify features significantly influencing loan risk classification.

2 Unsupervised exploration and clustering

2.1 Dimensionality reduction for data visualization

Apply dimensionality reduction techniques: t-Distributed Stochastic Neighbor Embedding (t-SNE) and Principal Component Analysis (PCA) to reduce and visualize the data.

2.2 Unsupervised Data Analyis

Employ 2 unsupervised clustering algorithms to group instances based on similar characteristics. Find their best hyper-parameters. Assess the quality of clusters and characterize them for the most common features. Also try to assess their potential implications by controlling the purity with respect to loan attribution label.

3 Supervised Data analysis

3.1 Classifier Selection

Experiment with a range of classifiers suitable for binary classification tasks. Select a minimum of three classifiers studied in the course (e.g., Decision Trees, Support Vector Machines, Neural Networks). At least one has to be a Neural Network for the following adversarial evaluation.

3.2 Cross Validation

Implement a rigorous cross-validation process involving training, evaluating and tuning of the hyper-parameters to get the best validation accuracy. Each classifier's performance should be summarized with a brief description of the optimized hyperparameters and a rationale for observed outcomes.

3.3 Classifier Evaluation

Evaluate the best selected classifier (the one obtaining the highest classification accuracy on the validation set) on the testing set. Utilize metrics like accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC) to assess the classifier's effectiveness in loan attribution.

[1] https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data

4 Adversarial Attacks

4.1 Random Noise

Apply random noise to the test data and measure its impact on classifier accuracy. Experiment with varying intensities of noise ϵ (clip the noise to have ϵ Euclidean norm) to measure the degradation in classifier performance. This metric is also called Robust Accuracy, i.e., the accuracy of a classifier when facing adversarial data.

4.2 Feature specific noise

Conduct a detailed study on the effect of random noise on individual features. Explore the classifier's sensitivity to specific features and identify the most vulnerable one. Determine the minimum distance ϵ required to frequently deceive the classifier. Discuss the implications of these adversarial attacks on the models, based on the example provided in the Introduction.

4.3 Adversarial Attack with ART

The Adversarial Robustness Toolbox (ART)[2] is a Python library for Machine Learning Security. ART provides tools for the automatic creation of adversarial attacks, based on advanced optimization techniques. Following the example in https://github.com/Trusted-AI/adversarial-robustness-toolbox/blob/main/examples/get_started_scikit_learn.py (works with scikit-learn, checks the one about Pytorch if you are using that instead), evaluate the adversarial attack against the neural network classifier at different distances (epsilon parameter). Is the distance similar to the one you obtained? Which features are affected more on average? Test at least two attacks implemented in the library (e.g., FGSM and PGD)

4.4 Countermeasure Exploration

Explore potential countermeasures to mitigate the impact of adversarial attacks on the loan attribution model. For instance, you can consider adversarial training: which consists in training a new classifier on both the standard data and the adversarial data. Re-evaluate the classifier equipped with countermeasure on both clean and new adversarial data to check both clean and robust accuracy.

References:

While external references are not strictly necessary for this assignment (except for ART), any utilized resources, such as course materials, should be appropriately cited within the document.

[2] https://github.com/Trusted-AI/adversarial-robustness-toolbox