DSAIT 4015 2025-2026

Project assignment 1

Citizen welfare fraud risk modeling of the municipality of Rotterdam

Instructors: Annibale Panichella and Cynthia Liem

Background

For the assignment, you will work with (synthesized) data on risk modeling of citizens of Rotterdam. For several years, the Rotterdam municipality used a machine learning method (colloquially referred to as 'an algorithm') to predict the risk of a citizen committing welfare fraud. This method was deemed controversial, as it was suspected that made predictions were discriminatory. After several years of inquiry, investigative journalists managed uncovering background information about the method, receiving documentation about used variables, training code, and even (accidentally leaked) data. When retraining a model based on the leaked data, the journalists concluded that indeed, discriminatory predictions had been made.

- Further background coverage on this story is available on Brightspace under `Project > Assignment 1 > Case background`
- The original training code for the model is available at https://github.com/Lighthouse-Reports/suspicion_machine
- While the original data could not be publicly shared, the journalists made a synthetic data generator that generates data with similar statistical properties to the original data. Based on this, we have generated a large dataset under `Project > Assignment 1 > Data` that you will use for this assignment.

Overall purpose of the assignment

In this assignment, you will:

1. Critically inspect the available data
2. Think of possible test cases for this data / prediction problem, and automate the generation of such test cases
3. Train a purposefully bad and good model
4. Test these models both in a classical machine learning way, as well as with your tests

A "good" model in this assignment refers to a model that behaves responsibly and avoids undesirable biased patterns (for example, discrimination by gender, age, ethnicity, or other sensitive attributes), **while still achieving good classical machine-learning performance**.

A "bad" model, in contrast, is one that behaves problematically (for instance, showing discriminatory or unfair behavior) **yet still appears to perform well according to traditional ML performance metrics**. In other words, both models may seem acceptable when judged only by accuracy or AUC, but they should differ meaningfully when examined from the perspective of undesirable discrimination and bias.

1. Data inspection

Please read the background articles of this case and inspect the dataset and data documentation. There are many issues with this data (e.g. in what kind of variables are incorporated, how they translate case information into data, and how they are used for prediction).

Describe **at least 3 problems** with the dataset, where **at least 1** of these touches upon questions of validity.

Validity concerns whether a variable or label genuinely represents the real-world concept it claims to measure. For example, if a feature is only a proxy for a protected attribute, or if the label does not truly reflect welfare fraud risk, this is a validity issue.

For the following part 2 and 3 (and the start of part 4), please split your group in 2 subgroups (which in most cases should yield 2 groups of 2 students). Within your subgroup, first work independently on part 2 and 3, so only let one another see your outcomes later, as part of part 4.

2. Possible test cases

Considering this case and dataset, think of possible test cases that would help you assessing whether a predictive model does not display undesired behavior.

Do this **at least**:

- From a perspective of **partitioning** (where when partitioning the data in a certain justifiable way, you would explicitly expect test results on these partitions to (not) be equivalent)
- And from a **metamorphic** perspective (where when applying a specific transformation you define to a datapoint, you assert the system output should not change)

Write software that automates the creation and testing of your chosen partitions, your chosen metamorphic transformation, and possible further tests you deem relevant.

For a higher grade and more interesting testing experiences in part 4, try to do this for a partitioning and a metamorphic relationship that the other subgroup may not think of!

3. A purposefully ‘good’ and ‘bad’ model

Now, train two machine learning models for the Rotterdam case:

- One that is purposefully bad, although it would perform ok in terms of classical machine learning performance metrics;
- One that is purposefully good.

As clarified earlier in this assignment, “good” and “bad” refer specifically to the undesirable biases of the model (i.e., avoidance vs. presence of undesirable biased patterns), **not** to classical predictive performance. Within this framework, **you are free to decide which kind of undesirable bias** your bad model exhibits (e.g., gender, age, proxy variables, geographic origin), as long as it remains plausible and connected to the case context. Be welcome to use different data subsets or additional synthesized data for the training of these two models, that may differ from the original dataset.

The only requirement is that both resulting models are released as binaries are in the onnx format, and that the input to your binaries has the same interface as the original data released to you (that is, if you choose to e.g. drop certain features, do this within your binary, but first read in all features, as an independent tester will only know what the original data looked like).

Under `Project > Assignment 1 > Assignment`, we have included a folder with sample code for how to get your models converted to onnx.

Release your two models as `model_1.onnx` and `model_2.onnx`, randomly picking which of these models holds the good and bad model.

Report on the performance of your good and bad models:

- In the way you would normally do when performing a machine learning project;
- And in terms of outcomes to the tests you established under part 2.

4. Independently testing the ‘good’ and ‘bad’ model

Share your `model_1.onnx` and `model_2.onnx` with the other subgroup within your team. Then, use your tests from part 2 to test the other subgroup’s `model_1.onnx` and `model_2.onnx`.

Based on your tests, can you tell which of the models is the better and worse one? Write a short summary of your main findings.

Cross-compare and discuss your outcomes with the other subgroup (also learning their testing outcomes of your models).

What to deliver

As final deliverable for this assignment, deliver a zip file containing:

1. A project report, with:

- A description and justification of the issues you see with the dataset from part 1 of this assignment (max 0.5 page);
- For each of your sub-groups:
  - A description and justification of chosen tests from part 2 of this assignment (max 1 page);
  - A description and justification of the good and bad model from part 3 of this assignment, with the two ways of performance evaluation you applied (classical machine learning and according to your tests) (max 3 pages);
  - The test results on your model obtained by the other sub-group (max 0.5 page);
- A joint reflection by your whole group on what your respective tests did and did not uncover (max 1 page).

2. A zip file with:

- For each of your sub-groups:
  - A `README.pd` file that explains (1) how to run your tests, (2) any special steps needed to reproduce your results;
  - A `requirements.txt` file listing (1) the python version used and (2) all python dependencies (libraries) with versions;
  - The code for your automated tests;
  - The training code used for your good and bad model;
  - Your `model_1.onnx` and `model_2.onnx`.