# AI Assurance

## Inspiration

The following program is inspired by the new European Act on the regulation of AI systems. This regulation encompasses AI systems ranging from rule-based and machine learning to generative AI.

The purpose of my application is to ensure that the data used to train machine learning or rule-based models adheres to the Trustworthy AI framework, particularly emphasizing fairness and impartiality.

One of the major objectives of the changes in the European Act is to prevent AI systems from exploiting individuals based on age, disability, social, or economic status. Additionally, these systems are prohibited from making decisions based on biometric data related to political, religious, philosophical beliefs, sexual orientation, and race. For the context of this work, the aforementioned characteristics will be considered as our sensitive features.

The goal of my application is to examine the data used to train and test models, identifying any bias or discrimination toward the sensitive features that the Act aims to protect.

Currently, the approach is designed for binary classification models, with plans to extend this framework to multiple models in the future.

## Other Work

The application I've developed builds upon previous work and approaches in this area, such as:

* Aif360, created by IBM
* Fairlearn, used for mitigation
* InterpretML, used to interpret black-box ML models
* Google What-If
* HELM, analysis of bias and fairness on LLM created by Stanford

Additionally, my work was inspired by research conducted at Deloitte, notably:

* “Ensuring Reliable AI in Real World Situations” by Deloitte Germany
* “Artificial Intelligence Act” by Deloitte Germany

My application, follows two main steps:

* Python scrip that identifies differences across training and testing splits for the sensitive features.
* Tableau dashboard to allow a visualisation across the splits and outcomes of the model.

## Dataset

For the following work, I used as a data example the COMPAS dataset:

The COMPAS dataset is a collection of data used in the development of the COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) algorithm, which is a tool utilized in the criminal justice system for risk assessment and sentencing decisions. It includes various demographic and criminal history information about individuals, such as age, gender, race, prior convictions, and recidivism outcomes. This dataset has been a subject of scrutiny and analysis due to concerns about potential biases in the algorithm's predictions, particularly regarding racial disparities in sentencing outcomes.

## Python Script

This script takes three different datasets as input:

* Training: This dataset is used to train the model.
* Testing: This dataset is used to test the model.
* Outcome: This dataset contains the model's predictions based on the testing data.

The datasets must adhere to a specific format in naming the features, using underscores ("\_") as separators:

* sen: This prefix identifies the sensitive categories we aim to investigate for fairness.
* cat: This prefix denotes that the feature is categorical.
* count: This prefix indicates that the feature is continuous.
* pred\_feat: This is the feature we are trying to predict.
* prediction: This is the model's outcome prediction.

The script is divided into two parts. First, it tests whether the splits between training and testing data are even. The second part investigates whether the model's outcomes are fair and impartial among members of a sensitive group.

**Testing Splits**

The purpose of testing for even splits is to identify any potential inherent biases in the model and during its creation. The testing process will be as follows:

* Test for an even distribution between groups.
* For each sensitive group, create subgroups for each member across the other sensitive groups.
* Using the real values of the feature to be predicted, subcategorize the sensitive groups to determine if each split has the same predicted distribution.

Types of tests included on scripts:

* Chi-squared
* Kolmogorov-Smirnov
* Proportion Differences Z-Test

**Evaluate the Outcomes**

After evaluating the splits between both datasets, we will assess the outcomes of the predictions using the following metrics:

* **Precision**: The ratio of correctly predicted positive observations to the total predicted positives. It shows how accurate the positive predictions are.
* **Recall**: The ratio of correctly predicted positive observations to all actual positives. It measures how well the model identifies actual positives.
* **F1 Score**: The harmonic mean of precision and recall, providing a balance between them. It combines both to give a single measure of the model's accuracy.
* **True Negatives Ratio**: The proportion of actual negatives correctly identified. It reflects the model's accuracy in identifying negative instances.
* **False Positives Ratio**: The proportion of actual negatives incorrectly identified as positive. It shows how often the model is wrong on negatives.
* **False Negatives Ratio**: The proportion of actual positives incorrectly identified as negative. It indicates the rate at which positives are missed.
* **True Positives Ratio**: Similar to recall, it measures the proportion of actual positives correctly identified. It focuses on the model's effectiveness in predicting positives.

After evaluating the model's output, we will segment the data based on the sensitive features and apply the same metrics to detect any significant disparities across specific groups. This analysis will help us identify if any group associated with a sensitive feature experiences significantly better or worse outcomes compared to the general population (entire dataset).

## Tableau Dashboard

In the dashboard is divided on two pages as the python scrip:

* Splits, here we can partition and subdivide each split across each of the sensitive groups and the predictable feature.
* Outcome, this dashboard will display percentage of positive outcomes across each sensitive group and the evaluation metrics across each member of the group.

# References

## Dataset

GitHub Repository:  
ProPublica. (n.d.). COMPAS Analysis. GitHub. https://github.com/propublica/compas-analysis

Article:  
Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016, May 23). Machine Bias: Risk Assessments in Criminal Sentencing. ProPublica. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>

## Work

Aif360:   
IBM. (n.d.). AI Fairness 360 (AIF360). GitHub. Retrieved from https://github.com/IBM/AIF360

Fairlearn:   
Microsoft. (n.d.). Fairlearn. GitHub. Retrieved from https://github.com/fairlearn/fairlearn

InterpretML:   
Microsoft. (n.d.). InterpretML. GitHub. Retrieved from https://github.com/interpretml/interpret

Google What-If:   
Google. (n.d.). What-If Tool. GitHub. Retrieved from <https://github.com/tensorflow/tensorboard/tree/master/tensorboard/plugins/interactive_inference>

HELM:   
Chen, J. (2018). HELM: A toolkit for understanding high-level embeddings in large scale networks. arXiv preprint arXiv:1805.04092. Retrieved from <https://arxiv.org/abs/1805.04092>

Thogmartin, D., Spitzer, P., & Körner, J. (Year of publication). *Trustworthy AI: Robust and Reliable* [White paper]. Deloitte Germany. Retrieved from <https://www2.deloitte.com/content/dam/Deloitte/de/Documents/Innovation/DELO-8505%20Trustworthy%20AI%20Robust%20and%20Reliable_KS6.pdf>

Thogmartin, D., & Berge, T. (2022, May). *Artificial Intelligence Act* [White paper]. Deloitte Germany. Retrieved from <https://www2.deloitte.com/content/dam/Deloitte/de/Documents/Innovation/Deloitte%20TAI%20DE%20-%20Artificial%20Intelligence%20Act%20update.pdf>