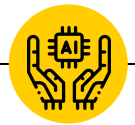
Quantifying the Trustworthiness Level of Artificial Intelligence Models and Decisions

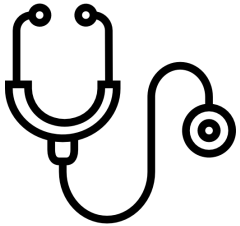


Agenda

- Motivation
- Taxonomy
- Trusted AI Algorithm
- Validation Scenarios
- Demo

Motivation

Credit Scoring



Medical Diagnostics

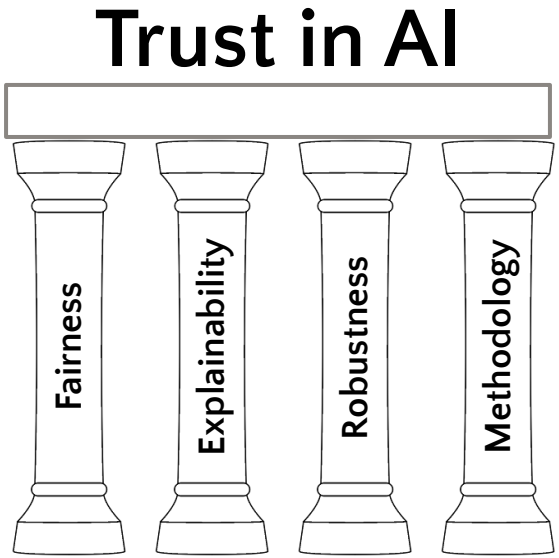
Applicant Screening

Autonomous Driving

Thesis Goals

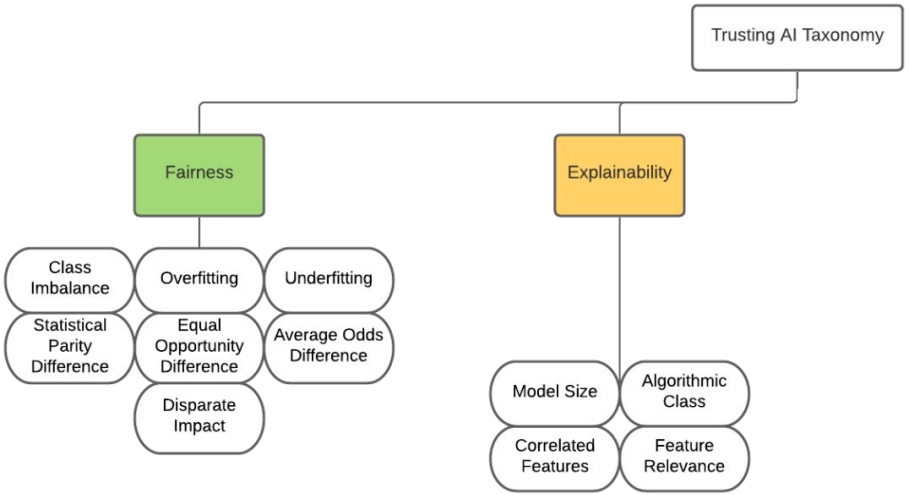
- 1. Identify metrics and key dimensions of trust
- 2. Compile a general taxonomy
- 3. Develop an algorithm to automatically compute the trust score of machine learning models
- 4. Prototypical implementation
- 5. Validation of the proposed solution

Pillars of Trust

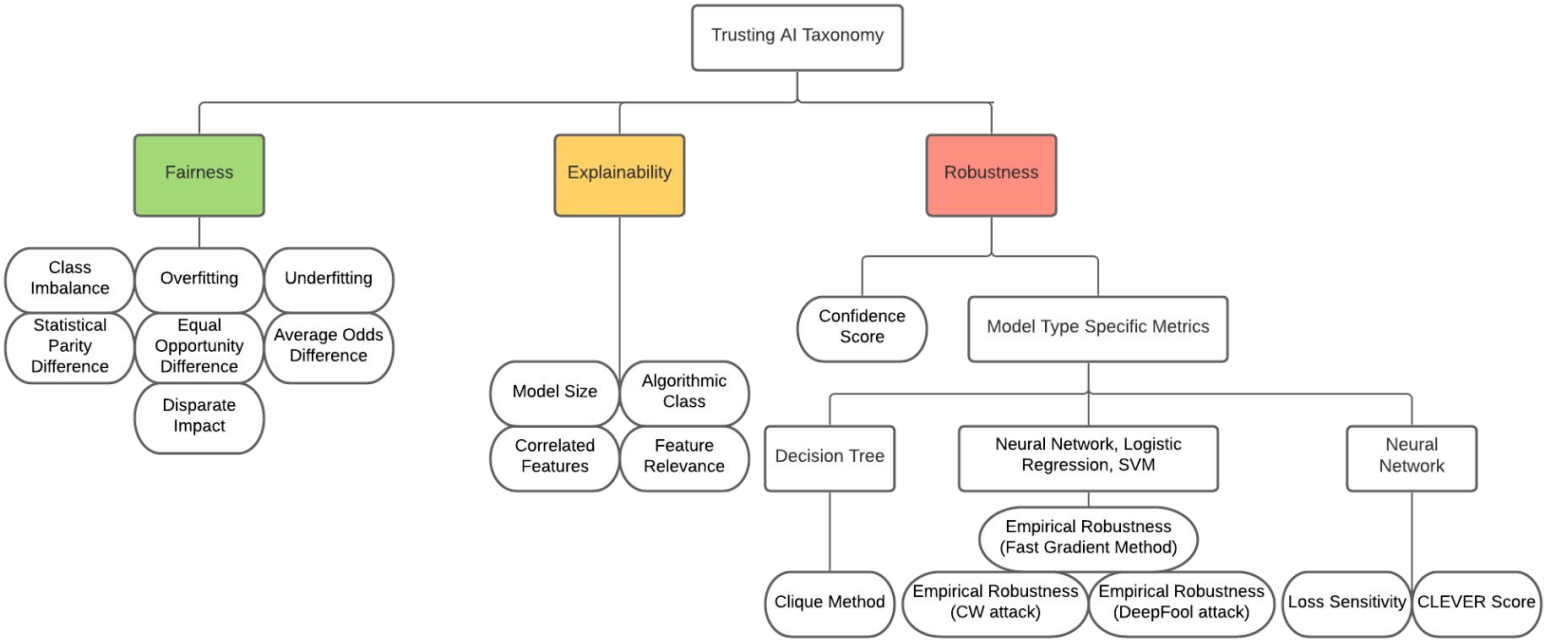


Taxonomy - Trusted AI Metrics

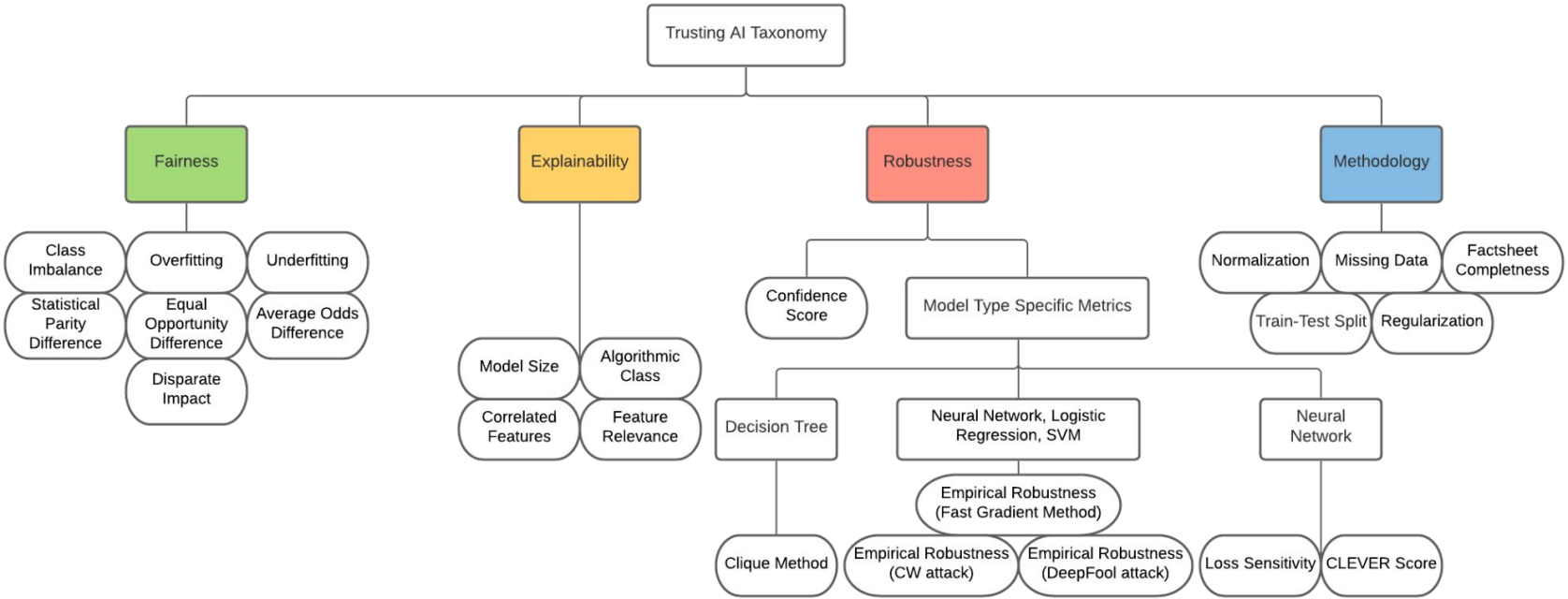
Taxonomy - Trusted AI Metrics



Taxonomy - Trusted AI Metrics



Taxonomy - Trusted AI Metrics



Metric Dependencies

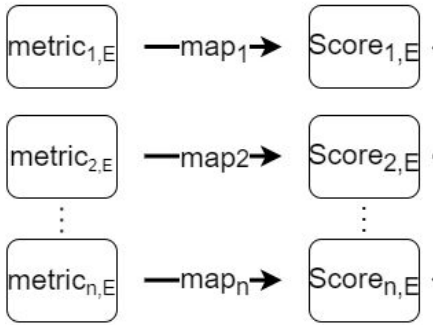
- ML Model
- Training Data
- Test Data
- Factsheet

Trusted AI Algorithm Design

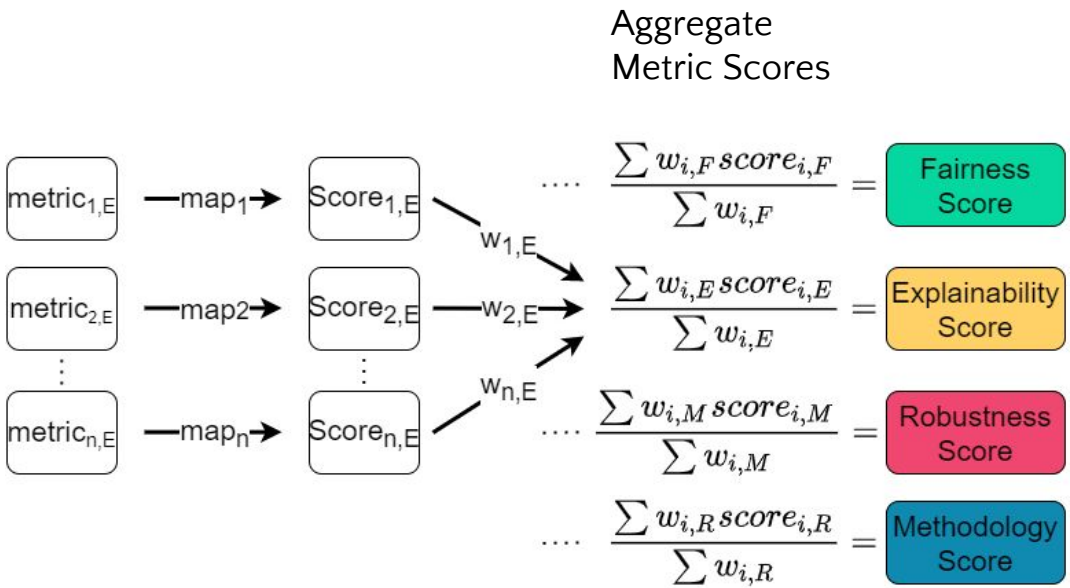
Pillar Specific Metrics

Trusted AI Algorithm Design

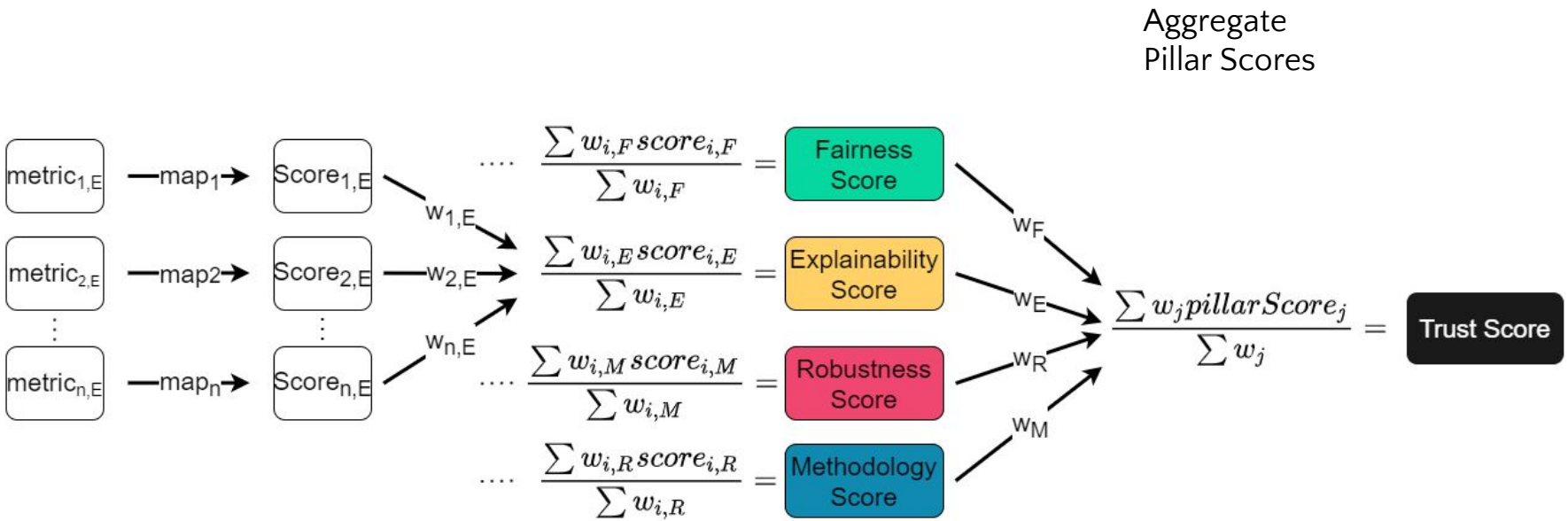
Metric Scores
between 1-5



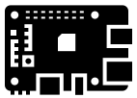
Trusted AI Algorithm Design



Trusted AI Algorithm Design



Validation Scenario 1



IoT Attack Classification:

- Multiple client devices
- Collecting status information
- Classify if device is subject to attack
- Classify the type of attack

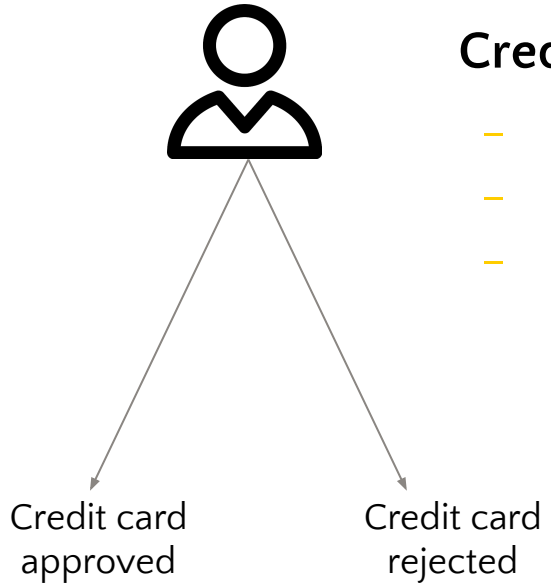
Not under attack

Under attack

Type of attack

1. *Fakepsd attack*
2. *Sendout attack*
3. *Write attack*
4. *Random attack*
5. *Exchange attack*
6. *Hide attack*

Validation Scenario 2

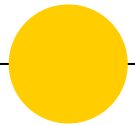


Credit Card Approval:

- Personal user information
- Credit scoring
- Classify if a credit card can be given

Demo Prototype

https://www.csg.uzh.ch/trusted-ai/



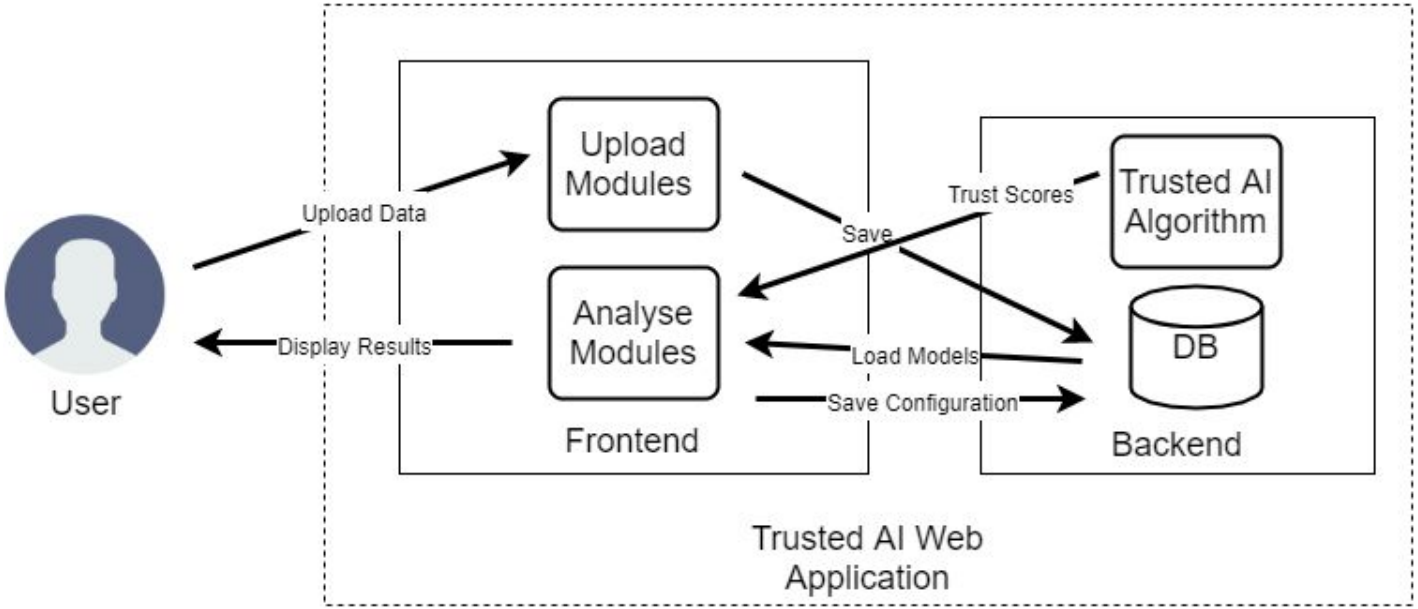
Conclusion and Future Work

- Compile a general taxonomy
- Develop an algorithm to automatically compute the trust score of machine learning models
- Validation of the proposed solution
- Add a new pillar (e.g. Privacy)
- Extend the set of metrics
- Add support for regression model
- Add support for other ML libraries

Questions?

Backup Slides

App Architecture

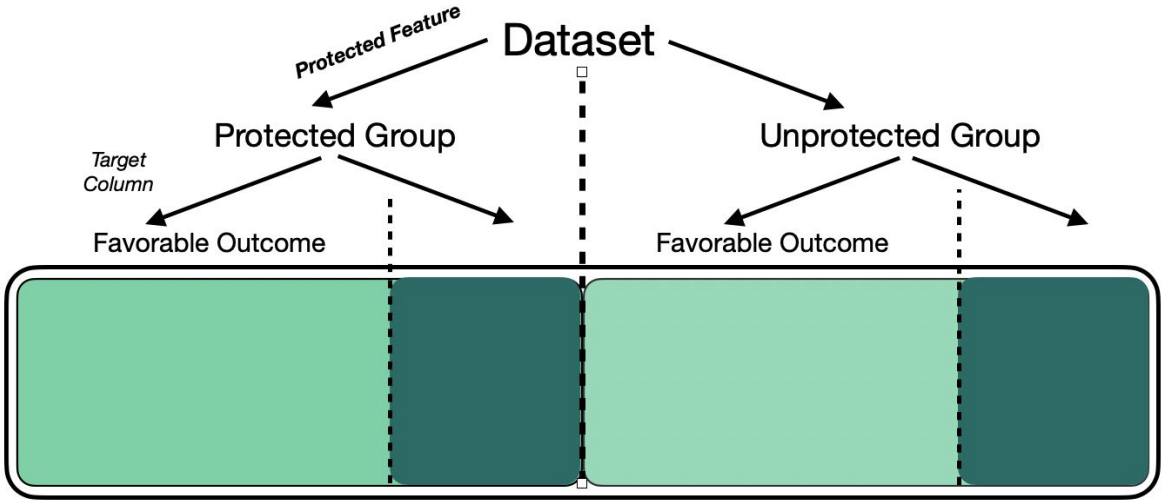


Trusted AI Algorithm

Algorithm Trusted AI Algorithm

function TRUSTED_AI($model, data_{train}, data_{test}, factsheet, config_{map}, config_{weights}$)
    $trust\_score \leftarrow 0$
    $scores\_pillars \leftarrow empty\ dictionary$
    for $p\ in\ pillars$ **do**
        $score_p \leftarrow 0$
        $scores\_metrics \leftarrow empty\ dictionary$
        $metrics \leftarrow get\_metrics(p)$
        for $m\ in\ metrics$ **do**
            $args \leftarrow config_{map}[m]$
            $scores\_metrics[m] \leftarrow get\_score_m(args)$
        for $(m \in keys, v \in values)\ in\ scores\_metrics$ **do**
            $w_m \leftarrow config_{weights}[m]$
            $score_p \leftarrow score_p + w_m * v$
        $scores\_pillars[p] \leftarrow score_p$
    for $(p \in keys, v \in values)\ in\ scores\_pillars$ **do**
        $w_p \leftarrow config_{weights}[p]$
        $trust\_score \leftarrow trust\_score + w_p * v$
    return $trust\_score$

Fairness Computation



Statistical Parity Difference

Algorithm 10 Statistical Parity Difference Metric

```
function STATISTICAL_PARITY_DIFFERENCE(model, data, factsheet)
    protected_feature ← factsheet["protected_feature"]
    protected_values ← factsheet["protected_values"]

    protected_group ← filter(data[protected_feature].is_in(protected_values))
    unprotected_group ← filter(data[protected_feature].not_in(protected_values))

    protected_group_size ← size(protected_group)
    unprotected_group_size ← size(unprotected_group)

    target_column ← factsheet["target_column"]
    favorable_outcomes ← factsheet["favorable_outcomes"]

    favored_protected_group ← filter(protected_group[target_column].is_in(favorable_outcomes))
    favored_unprotected_group ← filter(unprotected_group[target_column].is_in(favorable_outcomes))

    favored_protected_group_size ← size(favored_protected_group)
    favored_unprotected_group_size ← size(favored_unprotected_group)

    favored_protected_ratio ← favored_protected_group_size/protected_group_size
    favored_unprotected_ratio ← favored_unprotected_group_size/unprotected_group_size

    statistical_parity_difference ← |favored_unprotected_ratio − favored_protected_ratio|
    return statistical_parity_difference
```

Equal Opportunity Difference

Algorithm 11 Equal Opportunity Difference Metric

```
function EQUAL_OPPORTUNITY_DIFFERENCE(model, data, factsheet)
    protected_feature ← factsheet["protected_feature"]
    protected_values ← factsheet["protected_values"]

    protected_group ← filter(data[protected_feature].is_in(protected_values))
    unprotected_group ← filter(data[protected_feature].not_in(protected_values))

    target_column ← factsheet["target_column"]
    favorable_outcomes ← factsheet["favorable_outcomes"]

    favored_protected_group ← filter(protected_group[target_column].is_in(favorable_outcomes))
    favored_unprotected_group ← filter(unprotected_group[target_column].is_in(favorable_outcomes))

    favored_protected_group_size ← size(favored_protected_group)
    favored_unprotected_group_size ← size(favored_unprotected_group)

    predicted_favored_protected_group ← filter(favored_protected_group[y_pred].is_in(favorable_outcomes))
    predicted_favored_unprotected_group ← filter(favored_unprotected_group[y_pred].is_in(favorable_outcomes))

    predicted_favored_protected_group_size ← size(predicted_favored_protected_group)
    predicted_favored_unprotected_group_size ← size(predicted_favored_unprotected_group)

    predicted_favored_protected_ratio ← predicted_favored_protected_group_size/favored_protected_group_size
    predicted_favored_unprotected_ratio ← predicted_favored_unprotected_group_size/favored_unprotected_group_size

    equal_opportunity_difference ← |predicted_favored_protected_ratio − predicted_favored_unprotected_ratio|
    return equal_opportunity_difference
```

Equal Opportunity Computation Example

Unprotected Group

P=0	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	8000	3000
$Y=1$	4000	6000

Protected Group

P=1	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	7000	9000
$Y=1$	6000	9000

$$TPR_{unproteced} = Pr\{\hat{Y}=1|P=0,Y=1\} = 6000/10000 = 60\%$$
$$TPR_{protected} = Pr\{\hat{Y}=1|P=1,Y=1\} = 9000/15000 = 60\%$$
$$FPR_{unproteced} = Pr\{\hat{Y}=1|P=0,Y=0\} = 3000/11000 \approx 27.27\%$$
$$FPR_{protected} = Pr\{\hat{Y}=1|P=1,Y=0\} = 9000/16000 \approx 56.25\%$$
$$EOD(\hat{Y},P) = \frac{9000}{16000} - \frac{3000}{11000} \approx 56.25\% - 27.27\% \approx 28.98\%$$