

# AI Governance

---

A package for framework agnostic AI Governance

## Calculation

Responsible Index is a measure of the following and is a scale between 1 to 3. 3 being a responsible Model

1. Carbon Emissions
2. Differential Privacy
3. Bias (Imbalance class)
4. Explainability

## Pre-requisites

1. Emission tracking is calculated using [Code Carbon](#). Install Code Carbon using

```
!pip install codecarbon
```

2. Differential privacy is calculated using [Opacus](#). Install Opacus

```
!pip install opacus
```

3. Responsible index is calculated using ResponsibleML. Install ResponsibleML

```
!pip install spectre
```

## How it works

1. AI Governance SPECTRE is the overarching framework for responsible AI
2. SPECTRE is across 7 dimensions - Security, Privacy, Explainability, Trust, bias and Emissions
3. Current Framework support the above 4 for calculating RAI index
4. SPETRE has a. Model List (list of all models that you created for calculating RAI Index) b. Models - every model created by you for the use case.

## Code sample

1. Import the package in your code

```
from aigovernance import spectre
```

## 2. Initiate a Model and a Model List

```
r_model = spectre.responsibleModel(modelname, modeltype)
model_list = spectre.models()
```

3. Bias Information - During data engineering phase of your ML life cycle, send the label data frame (Y) to responsibleModel

```
r_model.calcualte_bias(label_df)
```

4. Model Explainability - given there is no straight forward way to identify if the model explainability is performed, you will have to add it independently

```
r_model.explained(isExplained)
```

5. Carbon Emissions - For calculating carbon emissions, before you start your model training, start the tracker. Once the model training is complete, stop the tracker

```
r_model.track_emissions()

<< your model training >>

r_model.stop_tracking()
```

6. Differential Privacy - [Work in Progress]

7. Responsible Index: You can now retrieve the responsible index of the model using

```
responsible_index = r_model.rai_index()
```

8. Given the nature of ML, you will be building multiple models. Use steps 2 to 7 initiate the model with responsible dimension. You can now add the models you created on to a model list

```
model_list.add_model(r_model)
```

9. You can list all the models and their responsible score using

```
model_list.list_models()
```

10. finally, to identify the most responsible model that you can use for your production use case, you can get the models ranked by their responsibility index

```
model_list.rank_models()
```