Aditya Sanjay Mhaske

SP MGMT_Assignment - Module: AI Fairness

In [1]:
```
#importing the required libaries
import numpy as np
from aif360.datasets import GermanDataset
from aif360.metrics import BinaryLabelDatasetMetric
from aif360.algorithms.preprocessing import Reweighing
```

```
WARNING:root:No module named 'tempeh': LawSchoolGPADataset will be unavailabl
e. To install, run:
pip install 'aif360[LawSchoolGPA]'
WARNING:root:No module named 'tensorflow': AdversarialDebiasing will be unavai
lable. To install, run:
pip install 'aif360[AdversarialDebiasing]'
WARNING:root:No module named 'tensorflow': AdversarialDebiasing will be unavai
lable. To install, run:
pip install 'aif360[AdversarialDebiasing]'
WARNING:root:No module named 'fairlearn': ExponentiatedGradientReduction will
be unavailable. To install, run:
pip install 'aif360[Reductions]'
WARNING:root:No module named 'fairlearn': GridSearchReduction will be unavaila
ble. To install, run:
pip install 'aif360[Reductions]'
WARNING:root:No module named 'fairlearn': GridSearchReduction will be unavaila
ble. To install, run:
pip install 'aif360[Reductions]'
```

In [2]:
```
# Loading the dataset to work with bias on age
dataset_orig = GermanDataset(
    protected_attribute_names=['age'],
    privileged_classes=[lambda x: x >= 25],
    features_to_drop=['personal_status', 'sex']
   )
```

In [3]:
```
#train test split
dataset_orig_train, dataset_orig_test = dataset_orig.split([0.7], shuffle=True)
```

In [4]:
```
privileged_groups = [{'age': 1}]
unprivileged_groups = [{'age': 0}]
```

In [5]:
```
# calculating the bias
metric_orig_train = BinaryLabelDatasetMetric(dataset_orig_train,
                                             unprivileged_groups=unprivileged_g
                                             privileged_groups=privileged_group

print("Difference in mean outcomes between unprivileged and privileged groups =
```

```
Difference in mean outcomes between unprivileged and privileged groups = -0.13
1727
```

In [6]:
```
# process the input data for model to mitigate pre processing bias.
#This transforms the dataset to have more equity in positive outcomes on the pr
#privileged and unprivileged groups.
RW = Reweighing(unprivileged_groups=unprivileged_groups,
```

```
                privileged_groups=privileged_groups)
dataset_transf_train = RW.fit_transform(dataset_orig_train)
```

In [7]:
```
metric_transf_train = BinaryLabelDatasetMetric(dataset_transf_train,
                                               unprivileged_groups=unprivileged
                                               privileged_groups=privileged_gro

print("Difference in mean outcomes between unprivileged and privileged groups =
```

```
Difference in mean outcomes between unprivileged and privileged groups = -0.00
0000
```

Hence, the bias is effectively mitigated.

Using alternate attributes (sex)

In [8]:
```
## Loading the dataset to work with bias on sex
label_map = {1.0: 'Good Credit', 0.0: 'Bad Credit'}
protected_attribute_maps = [{1.0: 'female', 0.0: 'male'}]
gd = GermanDataset(protected_attribute_names=['sex'], privileged_classes=[['fem
```

In [9]:
```
# calculating the bias
privileged_groups = [{'sex': 1}]
unprivileged_groups = [{'sex': 0}]
metric_orig_train = BinaryLabelDatasetMetric(gd,
                                             unprivileged_groups=unprivileged_g
                                             privileged_groups=privileged_group

print("Difference in mean outcomes between unprivileged and privileged groups =
```

```
Difference in mean outcomes between unprivileged and privileged groups = 0.074
801
```

Here, we can see that there is a positive bias towards females. So we will reweight the model and mitigate the bias

In [10]:
```
# process the input data for model to mitigate pre processing bias.
#This transforms the dataset to have more equity in positive outcomes on the pr
#privileged and unprivileged groups.
RW = Reweighing(unprivileged_groups=unprivileged_groups,
                privileged_groups=privileged_groups)
gd = RW.fit_transform(gd)
```

In [11]:
```
metric_transf_train = BinaryLabelDatasetMetric(gd,
                                               unprivileged_groups=unprivileged
                                               privileged_groups=privileged_gro

print("Difference in mean outcomes between unprivileged and privileged groups =
```

```
Difference in mean outcomes between unprivileged and privileged groups = 0.000
000
```

Hence, the bias is effectively mitigated.

By utilizing the AI Fairness 360 toolkit in a credit scoring dataset, I gained knowledge on how to identify and reduce bias. The tutorial allowed me to comprehend the impact of bias in credit scoring and its consequences on various groups. Bias is an error in a model's

predictions that occurs due to the algorithm or training data, resulting in unfair treatment and negative outcomes such as discrimination and injustice. Pre-processing techniques like re-sampling, and post-processing techniques such as threshold adjustments can help mitigate bias. To detect and reduce bias in models, I employed the AI Fairness 360 toolkit, an open-source library. During the process, I encountered an error while using conda, so I opted to install the toolkit with pip instead. Additionally, loading data for the sex field posed a challenge

Bias refers to a systematic error in a models predictions that occours due to its training data or algorithm. Bias resutls in the unfair treatment of certain groups of individuals, leading to negative consequences such as discrimination and injustice.

Bias can be mitigated with pre-processing techniques such as re-sampling and post-processing techniques such as threshold adjustments. Tutorial uses AI Fairness 360 toolkit, an open-source library, to help detect and mitigate bias in models.

I faced error while using conda so I installed it with pip insted. It was also challenging to load data for sex field so I used https://aif360.readthedocs.io/en/latest/modules/generated/aif360.datasets.GermanDataset.htm for reference. I also went on to study the concepts of bias and its types might in detail to understand the tutorial better.