Machine Learning Fairness

Outlines

- The problem statement
- Technical stack
    A. Facts about the two fairness libraries used in the analysis; FairLearn by Microsoft and AIF360 by IBM
    B. Definitions of the fairness metrics
- Steps to be followed
- Conclusion
- References (citation).

The problem statement

- Machine learning fairness is a critical and evolving concept that addresses the ethical concerns associated with the development and deployment of machine learning models. It emphasizes the need for algorithms to be unbiased and equitable, treating all individuals or groups fairly. The goal is to prevent the reinforcement of existing societal biases and discrimination in the data used to train these models. Achieving fairness in machine learning involves careful consideration of various aspects, including data collection, model training, and decision-making processes.
- How to detect bias in the machine learning models? If bias exists, how to mitigate it?

Technical stack

- There are fairness libraries in python such as Fairlearn and AIF360.
- Below there are facts about the two libraries
- Definitions for the fairness metrics are given below as well

Facts about the fairness libraries - FairLearn

- Fairlearn tests group fairness only.
- Fairlearn tests fairness in the model performance only.
- Fairlearn provides tools to assess fairness of predictors for classification and regression.
- Fairlearn also provides tools that mitigate unfairness in classification and regression.

Facts about the fairness libraries – AIF360

- AIF360 library is available in both Python and R.
- AIF360 provides a comprehensive set of metrics to test both group and individual bias in the model as well as dataset.
- AIF360 provides algorithms to mitigate both group and individual bias in the model as well as dataset.
- One can run the bias analysis either in a notebook, or on the IBM cloud by creating IBM Watson Studio account

Definitions of the Fairness Metrics

- Fairlearn package is most applicable to two kinds of harms:

1. **Allocation harms** can occur when AI systems extend or withhold opportunities, resources, or information.

2. **Quality-of-service harms** can occur when a system does not work as well for one person as it does for another.

- For the WLC model, we focus on allocation harm so that the model is fair in selecting the customers.

- **Selection Rate:** Percentage of samples with positive selection
- **Demographic parity:** Measures the allocation harm.
- **Demographic Parity Difference:** The maximum absolute difference between groups for the demographic parity
- **Demographic Parity Ratio:** The minimum ratio between groups for the demographic parity
- For our bias analysis, we use demographic parity difference as both metrics represent the same result.

Different Type of Fairness

- **Individual Fairness:** Pairs of individuals that are equal, except for their membership of a protected group, are put in the same situation. If the member of a protected group is treated less favorably, this is regarded as discrimination.
- **Group Fairness:** A classifier satisfies this definition if subjects in both protected and unprotected groups have equal probability of being assigned to the positive predicted class.

Individual Metrics

1. **Generalized Entropy Index:** Generalized entropy index measures inequality (non-randomness) over a population. Generalized entropy index is proposed as a unified individual and group fairness measure in [1]. Index=0 means non-randomness =0 or in other words randomness is maximum and thus the model fairness is maximum. The bigger the index the worse the fairness.

   Theil index and the coefficient of variation measure the same thing as generalized entropy index. The generalized entropy index and the coefficient of variation are especially sensitive to the existence of large instances, whereas Theil index is especially sensitive to the existence of small instances

   aif360.sklearn.metrics.generalized_entropy_error — aif360 0.5.0 documentation

2. **Consistency Score:** The metric computes the consistency score. Individual fairness metric from [1] that measures how similar the (predicted) labels are for similar instances (records). It compares a model's classification prediction of a given data item x to its k-nearest neighbors, kNN(x). It applies the kNN function to the full set of examples to obtain the most accurate estimate of each point's nearest neighbors

   Learning Fair Representations (mlr.press).

3. **Generalized Entropy Error:** The metric computes the generalized entropy. The discrepancy between i's preference for the outcome i truly deserves (i.e., y_i), and i's preference for the outcome the learning algorithm assigns (i.e., $\hat{y}_i$).

[A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual & Group Unfairness via Inequality Indices (acm.org)]

Individual Metrics Based on Sample Distortion Metric

1. **Average Euclidean Distance:** Difference of the averages of Euclidean Distance between the samples from the two datasets.
2. **Average Mahalanobis Distance:** Difference of the averages of Mahalanobis Distance between the samples from the two datasets. The Mahalanobis distance accounts for how correlated the variables are to one another.
3. **Average Manhattan Distance:** Difference of the averages of Manhattan Distance between the samples from the two datasets. Manhattan Distance is preferred over the Euclidean distance metric as the dimension of the data increases. This occurs due to the ‘curse of dimensionality’.

Group Metrics

1. **Statistical Parity Difference:** Difference in selection rates (Percentage of samples with positive selection)

$$Pr(\hat{Y} = \text{pos\_label}|D = \text{unprivileged}) - Pr(\hat{Y} = \text{pos\_label}|D = \text{privileged})$$

2. **Disparate Impact Ratio:** Ratio of selection rates = $\frac{Pr(\hat{Y} = \text{pos\_label}|D = \text{unprivileged})}{Pr(\hat{Y} = \text{pos\_label}|D = \text{privileged})}$

3. **Equal Opportunity Difference:** Returns the difference in recall scores (TPR) between the unprivileged and privileged groups. A value of 0 indicates equality of opportunity.

4. **Average Odds Difference:** Returns the average of the difference in FPR and TPR for the unprivileged and privileged groups. A value of 0 indicates equality of odds.

$$\frac{(FPR_{D=\text{unprivileged}} - FPR_{D=\text{privileged}}) + (TPR_{D=\text{unprivileged}} - TPR_{D=\text{privileged}})}{2}$$

5. **Average Odds Error:** Returns the average of the absolute difference in FPR and TPR for the unprivileged and privileged groups. A value of 0 indicates equality of odds.

$$\frac{|FPR_{D=\text{unprivileged}} - FPR_{D=\text{privileged}}| + |TPR_{D=\text{unprivileged}} - TPR_{D=\text{privileged}}|}{2}$$

6. **Class Imbalance:** Compute the class imbalance = $(N_u - N_p)/(N_u + N_p)$ where N_u is the number of samples in the unprivileged group and N_p is the number of samples in the privileged group.

7. **Conditional Demographic Disparity:** (1) Across the entire affected population, which protected groups could I compare to identify potential discrimination? (2) How do these protected groups compare to one another in terms of disparity of outcomes? CDD answers both of these questions by providing measurements for making comparisons across protected groups in terms of the distribution of outcomes.

S. Wachter, B. Mittelstadt, and C. Russell, "Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI," Computer Law & Security Review, Volume 41, 2021.

Conditional Demographic Disparity (CDD) - Amazon SageMaker

8. **Smoothed EDF**: Smoothed EDF has a particularly elegant intersectionality property such as protecting higher-level groups and protecting intersectional subgroups.

1807.08362.pdf (arxiv.org)

9. **DF Bias Amplification:** It is a measure of the extent to which the classifier increases the unfairness over the original data

1807.08362.pdf (arxiv.org)

10. **Between Group Generalized Entropy Error:** Compute the between-group generalized entropy. Between-group generalized entropy index is proposed as a group fairness measure in [1] and is one of two terms that the generalized entropy index decomposes to.

T. Speicher, H. Heidari, N. Grgic-Hlaca, K. P. Gummadi, A. Singla, A. Weller, and M. B. Zafar, "A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual and Group Unfairness via Inequality Indices," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018.

- **Group Metrics Based on Classification Metric**

1. **Between all Groups Generalized Entropy Index:** It measures the between-group discrepancy between the true and predicted labels that uses all combinations of groups.
2. **Error Rate Difference:** Difference in error rates (1-accuracy) for unprivileged and privileged groups, $ERR_{D=unprivileged} - ERR_{D=privileged}$.
3. **Rich Subgroup:** Audit dataset with respect to rich subgroups defined by linear thresholds of sensitive attributes. Auditing for rich subgroup fairness means finding the subgroup for whom the statistical fairness constraint was most violated and return the gamma disparity with respect to the fairness_def . The fairness_def which sets the statistical fairness constraint is 'FP' or 'FN' for rich subgroup wrt to false positive or false negative rate. 1808.08166.pdf (arxiv.org)
4. **Performance Measures:** Compute various performance measures on the dataset, optionally conditioned on protected attributes. *This generates a list of metrics, so I did not include it in the graph.*

Generic Metrics

1. **Specificity Score:** Compute the specificity or true negative rate.
2. **Sensitivity Score:** Alias of recall score for binary classes only.
3. **Base Rate:** Compute the base rate, Pr(Y=pos_label) = P/(P+N).
4. **Selection Rate:** Compute the selection rate, Pr(Y^=pos_label) = (TP+FP)/(P+N).
5. **Smoothed Base Rate:** Compute the smoothed base rate = $(P+\alpha)/(P+N+|R_Y|\alpha)$. Different version of base rate
6. **Smoothed Selection Rate:** Compute the smoothed selection rate = $(TP+FP+\alpha)/(P+N+|R_Y|\alpha)$. Different version of selection rate.
7. **Generalized FPR:** Return the ratio of generalized false positives to negative examples in the dataset, GFPR=GFP/N. Generalized confusion matrix measures such as this are calculated by summing the probabilities of the positive class instead of the hard predictions.
8. **Generalized FNR:** Return the ratio of generalized false negatives to positive examples in the dataset, GFNR=GFN/P. Generalized confusion matrix measures such as this are calculated by summing the probabilities of the positive class instead of the hard predictions.

Steps to be followed

1. Collect the dataset in randomly using one of the sampling techniques.
2. Follow the rules in collecting the data to avoid societal bias
3. Test the dataset for bias using the pre-processing fairness metrics
4. If bias detected, use the preprocessing mitigation algorithm to reduce the bias
5. Train the model, use the in-processing mitigation algorithms to reduce the in-processing bias if exists.
6. After completing the model, use the model performance fairness metrics to detect bias
7. If exists, use the post-processing mitigation algorithms to reduce the bias

Conclusion

- Machine learning bias is inherited in the machine learning lifecycle. It can hit the model during data collection, training the model, and after model competing the model training. There are python and R libraries to detect the ML bias. If bias detected, there are mitigation algorithms in these libraries can be used to reduce the bias exist throughout the machine learning lifecycle either pre-processing, in-processing, and post-processing.

References (citation).

- https://fairlearn.org/
- https://github.com/fairlearn/fairlearn
- https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/
- https://www.microsoft.com/en-us/research/uploads/prod/2020/05/Fairlearn_WhitePaper-2020-09-22.pdf
- https://github.com/Trusted-AI/AIF360
- https://aif360.res.ibm.com/
- https://aif360.readthedocs.io/en/latest/
- https://www.ibm.com/opensource/open/projects/ai-fairness-360/
- https://ai-fairness-360.org/