Bias in AI Scientific Report

Anonymous Author
Department of Computer Science
Durham University
Durham, United Kingdom

***Abstract*—**

***Index Terms*—**

I. Project proposal

What I'd like to do for my course project is to analyse the human-centric German-Credit dataset for assessing credit risk [].

What I intend to implement is one of the two methods of modifying data that [2] proposes, called Combinatorial and Geometric repair.

The motivation for this project is to replicate the paper's claim that the repair performs favourably in terms of training classifiers that are both accurate and unbiased.

The reasons I believe that the implementation suits this submodule as bias in artificial intelligence is that it lets me develop a fair machine learning ecosystem to detect, reduce, and eventually mitigate different types of bias (such as the "80% rule" of disparate impact) that exist in the final outcome of the algorithm in various ways

The concrete tasks I plan to do are as follows:

1) Perform "cleaning", binning, bucketing, and discrete-to-continuous feature transformations on the dataset.
2) Split the dataset into different demographic groups, and compute and report information for each feature for each group.
3) Observe any interesting differences between the different subgroups' statistics, and describe any bias observed and explain the reasons why this bias has happened from my point of view.
4) Describe my chosen conventional algorithm to implement on the biased dataset, as well as describe the justification for selection.
5) Naively split the dataset into training and testing sets.
6) Train the model and see how it generalises to the testing dataset, as well as explain my approach and findings.
7) Subsample a new testing dataset in an unbiased way, then retrain the model and see how it generalises to these new testing conditions.
8) Compare my findings with the previous results and explain my approach.
9) Describe any sort of bias I observed, and explain the reasons why this bias has happened from my point of view.
10) Implement one of Combinatorial or Geometric repair, and describe the algorithm.
11) Test the performance of the trained model for the minority groups, and compare it with the performance of the model over the majority group.
12) Describe any reduction in algorithmic bias I observe.
13) State whether I get roughly the same results as the project paper, and, if not, reconsider my code or justify the reasons.
14) Describe from my point of view any sort of reduction in accuracy I observe.

The final work product of these tasks will be appropriate and proper plots and graphs that demonstrate any reduction in algorithmic bias and any sort of reduction in accuracy I observe after the fair machine learning implementation.

As I formulate this project, the particular context I'm thinking about is applying for credit loans.

The technologies I plan to use in my implementation are as follows: the Python programming language, the Pandas package for data analysis, and the Scikit-learn package for the implementation of the AI algorithm.

II. Project progress report

A. Data analysis

In terms of “cleaning”, binning, bucketing, and discrete-to-continuous feature transformations I did on the dataset, I did only bucketing: I converted the column ‘Age’ with integer values to the column ‘Age_Group’ with four categorical values. The categorical value (age group) and range of integers (age range), inclusive, that they correspond to are as follows: **Young,** 19-29; **Young Adults,** 30-40; **Senior,** 41-55; **Elder,** 55+.

Then, we split the dataset into different demographic groups, by writing down the size of the groups, the average value for each (numeric) feature, the variance of each (numeric) feature, the mode for each categorical feature, and the three most frequent values for each categorical feature, each computed on the different demographic subgroups. The two different demographic groups are age group and sex, and their different demographic subgroups, respectively, are young, young adults, senior, or elder, and male or female. The different subgrouos’ statistics are presented in Tables I, II, and III.

TABLE I
Table of the size of each demographic subgroup.

	male	female	Young	Young Adults	Senior	Elder
size	690	310	371	355	203	71

TABLE II
Table of the average value and variance of each (numeric) feature for the different demographic subgroups.

		Job	Credit amount	Duration
male	average	1.9	3448.0	21.6
	variance	0.4	8412806.3	154.7
female	average	1.8	2877.8	19.4
	variance	0.5	6776346.3	122.1
Young	average	1.8	3089.0	20.8
	variance	0.3	7261837.7	142.6
Young Adults	average	2.0	3375.5	21.5
	variance	0.4	7646336.1	139.2
Senior	average	1.9	3366.4	20.2
	variance	0.4	7986564.4	146.1
Elder	average	1.8	3430.4	20.5
	variance	0.7	13329819.2	192.5

The interesting differences between the different subgroups’ statistics that I observe are as follows: there are more than twice as many males than females; the sizes of the ‘Young’ and ‘Young Adults’ subgroups are each larger than the ‘Senior’ and ‘Elder’ subgroups combined; and, ‘Young Adults’ are the only subgroup where the mode of the ‘Checking account’ feature is ‘moderate’, and not ‘little’.

I have observed bias in both the gender and age demographic groups: a greater proportion of females than males were assigned a ’bad’ credit risk; and, a greater proportion of 19-29 year-olds (the ‘Young’ subgroup) were assigned a ‘bad’ credit risk compared to other age groups. This data is presented in Table IV.

From my point of view, the reasons why this happened are because there are fewer females than males in the data, and the size of the ‘Young’ subgroup is greater than any of the other age groups, so the data is not representative.

B. Conventional implementation

At this stage of the project, I have a biased dataset. I implement a conventional ML algorithm that is widely used (and is potentially biased) to solve a classification problem — whether to assign someone a ‘good’ or ‘bad’ credit risk.

My chosen algorithm is the support-vector machine (SVM) [1]. An SVM is a discriminative classification algorithm which finds the line or curve (in two dimensions) or manifold (in multiple dimensions) that divides classes of data from each other with the maximum margin. In two dimensions, the margin is the width of the separating line or curve, which can be the perpendicular distance to the nearest data point when plotted on a graph.

The justification for selecting this algorithm is that it is the same conventional ML model as used in the suggested paper [2].

My first approach was as follows:

1) Naively split the dataset into training and testing sets by randomly sampling some of the data, with $\frac{2}{3}$ for training and $\frac{1}{3}$ for testing (the same ratio used in [2]).
2) Standardise numeric features by subtracting the mean and scaling to unit variance by dividing by the standard deviation.
3) Encode categorical features using a one-hot scheme, by creating a binary column for each category where 1 denotes an instance of the category and 0 does not.
4) Encode target labels with values 1 for ‘good’ and ‘0’ for ‘bad’.
5) Exhaustively consider all parameter combinations by grid search to find the best for the SVM with L_2 regularisation.
6) Train the model on the training dataset.
7) Make sure the model is not over-fitting, by cross-validation.
8) See how the model generalises to the testing dataset, by calculating an accuracy score.

My findings were as follows: the best parameter combination for the classifier was a regularisation parameter (C) of value 0.95 and a polynomial kernel function of degree 4; the mean accuracy on the training data was 81.1%; the mean accuracy after cross-validation was 73.9%; and, the accuracy on the testing data was 72.5%.

My next approach was to subsample a new testing dataset in an unbiased way, followed by retraining the model on the training dataset, and then seeing how it generalises to the testing dataset with new conditions. I chose to ensure gender diversity by resampling the data so that the ratio of males to females was 1 to 1. I did this by randomly sampling a subset of the male data, with size equal to the female data, and then

TABLE III
TABLE OF THE THREE MOST FREQUENT VALUES FOR EACH CATEGORICAL FEATURE FOR THE DIFFERENT DEMOGRAPHIC SUBGROUPS.

		Housing	Saving accounts	Checking account	Purpose	Risk
male	mode	own (517)	little (409)	little (186)	car (243)	good (499)
	2nd most frequent	free (89)	moderate (71)	moderate (183)	radio/TV (195)	bad (191)
	3rd most frequent	rent (84)	quite rich (47)	rich (43)	furniture/equipment (107)	-
female	mode	own (196)	little (194)	little (88)	car (94)	good(201)
	2nd most frequent	rent (95)	moderate (32)	moderate (86)	radio/TV (85)	bad (109)
	3rd most frequent	free (19)	rich (19)	rich (20)	furniture/equipment (74)	-
Young	mode	own (248)	little (242)	little (115)	radio/TV (117)	good(234)
	2nd most frequent	rent (113)	moderate (42)	moderate (112)	car (102)	bad (137)
	3rd most frequent	free (10)	quite rich (19)	rich (24)	furniture/equipment (84)	-
Young Adults	mode	own (278)	little (201)	moderate (100)	car (128)	good(264)
	2nd most frequent	free (39)	moderate (41)	little (81)	radio/TV (93)	bad (91)
	3rd most frequent	rent (38)	quite rich (24)	rich (18)	furniture/equipment (58)	-
Senior	mode	own (143)	little (117)	little (57)	car (79)	good (150)
	2nd most frequent	free (40)	moderate (16)	moderate (39)	radio/TV (51)	bad (53)
	3rd most frequent	rent (20)	quite rich (15)	rich (15)	furniture/equipment (36)	-
Elder	mode	own (44)	little (43)	little (21)	car (28)	good (52)
	2nd most frequent	free (19)	rich (5)	moderate (18)	radio/TV (19)	bad (19)
	3rd most frequent	rent (8)	quite rich (5)	rich (6)	business (9)	-

TABLE IV
TABLE OF THE PERCENTAGE OF EACH DEMOGRAPHIC SUBGROUP WHERE 'RISK' IS 'BAD'.

	male	female	Young	Young Adults	Senior	Elder
%bad	27.7	35.2	36.9	25.6	26.1	26.8

discarding the remaining male data. When splitting the dataset into training and testing sets, I applied stratified sampling to preserve the percentage of samples for the male and female classes (i.e. 1 to 1).

My new findings were as follows: the best parameter combination for the classifier was a C of value 1 and a polynomial kernel function of degree 3; the mean accuracy on the training data was 78.5%; the mean accuracy after cross-validation was 68.8%; and, the accuracy on the testing data was 67.6%

Compared to the previous results, these new results have worse accuracy scores at each stage.

I observed there to be bias towards females. From my point of view, the reason why this bias has happened is because, in the first approach, sex contributed to how the target was classified.

C. Fair machine learning implementation

In this step, I implement one of the fair ML methods of mitigating bias. I chose only one algorithm to implement: Geometric repair. The justification for this choice is that this seems the easier of the two solutions that were provided in the suggested project paper [2].

The proposed algorithm, Geometric repair, is as follows:

1) Given a biased dataset D, with unprotected columns (attributes) Y and stratifying (protected) columns S, make all stratified groups (all possible combinations of protected attributes), and let these be the values of S.
2) Store the size of each stratified group; if a group has size 0, ignore it.
3) Pick a number of quantiles, with the maximum number equal to the size of the smallest stratified group, so that there will be at least one entry per quantile.
4) For each Y column with orderable values, over its unique values, for each stratified group, find the median value at the 1^{st} quantile.
5) For each Y column, find the median value of the median values, preferring the smaller item in the case of even-length lists, and call this the target value for all values of the column in the 1^{st} quantile of each stratified group.
6) For each Y column, for each original value in the 1^{st} quantile, update the original value to the repair value,
$$rv = ((1-\lambda) * original) + (\lambda * target).$$
7) Proceed similarly for each remaining quantile.

We test the performance of our trained model for the minority groups. Compared with the performance of our model over the majority group, ...

We observe a reduction in algorithmic bias, which is .... We plot the results demonstrating this below:

We do get roughly the same results as in the project paper. We justify the reasons by ...

REFERENCES

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[2] Michael Feldman. Computational fairness: Preventing machine-learned discrimination. 2015.