Evaluating and Mitigating Bias in an AI recruitment system

Module Name: < COMP2261>

Date: <26/04/2022>

Submitted as part of the degree of Computer Science to the

Board of Examiners in the Department of Computer Sciences, Durham University

1 Introduction

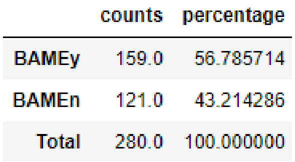
Though data cannot be bias, if AI trains algorithms to make decisions based on previous data, it can use bias to sensitive groups reflected in the data, to draw their conclusions. However, presently we can use machine learning to analyse results from this data to detect and become aware of discrimination and help AI systems mitigate these biases in the future. In this report we will be discussing statistical analysis of a recruitment system, how bias it is, and if we can mitigate this bias using adversarial learning.

2. METHODOLOGY

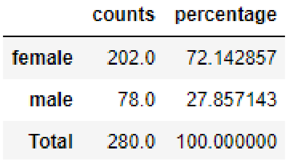
2.1 DataSet

From the dataset we can identify some sensitive attributes that can affect the outcome of the recruitment process. Sensitive attributes can be defined as those who cannot change their attribute, yet they are more/less favoured. According to the Equality Act 2010 protected characteristics of discrimination include sex and race and are protected from discrimination at work and when applying for a job. Therefore, in this data set we can identify 2 sensitive attributes BAMEyn and Gender. We can also determine privileged and unprivileged groups of these sensitive attributes based on discrimination in history where females and BAME groups would fight for equal rights.

As our data is discrete, we can give statistics for our privileged and unprivileged groups such as counts and percentages. By also using .describe() we can get our quartiles and standard deviations.

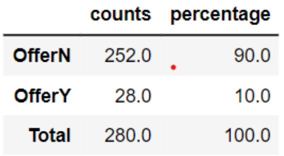


	counts	percentage
BAMEy	159.0	56.785714
BAMEn	121.0	43.214286
Total	280.0	100.000000

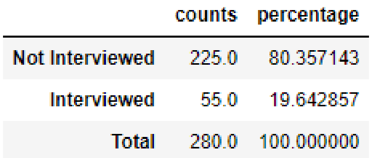


	counts	percentage
female	202.0	72.142857
male	78.0	27.857143
Total	280.0	100.000000

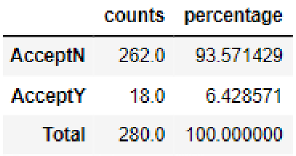
From the data above we can see that more of our applicants are female than male, and we have a majority of BAME applicants than non-BAME. We can also demonstrate statistics of each column as shown below:



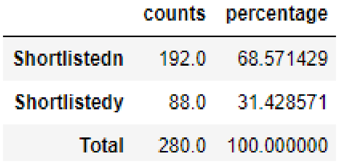
	counts	percentage
OfferN	252.0	90.0
OfferY	28.0	10.0
Total	280.0	100.0



	counts	percentage
Not Interviewed	225.0	80.357143
Interviewed	55.0	19.642857
Total	280.0	100.000000



	counts	percentage
AcceptN	262.0	93.571429
AcceptY	18.0	6.428571
Total	280.0	100.000000



	counts	percentage
Shortlistedn	192.0	68.571429
Shortlistedy	88.0	31.428571
Total	280.0	100.000000

Some interesting statistics we can see are that of those shortlisted, only 63% turned up to the interview, and of those offered a job only 64% accepted.

$$\frac{Pr(Y=1|D=\text{unprivileged})}{Pr(Y=1|D=\text{privileged})}$$

We can also show statistical disparity of our unprivileged and privileged group with a group fairness metric called the adverse impact ratio demonstrated in the formula above. One of the most common measures for employee selection is the four-fifths rule which says that "if the selection rate for a certain group is less than 80 percent of that of the group with the highest selection rate, there is adverse impact on that group." By applying this to our sensitive attributes we get the following results:

Unprivileged & Privileged/Shortlisted	Short listed	Not Shortlisted	Total	Ratio
Male (unprivileged)	38	40	78	45%
Female (privileged)	50	152	202	25%
Total	88	192	280	
BAME yes (unprivileged)	19	102	121	16%
BAME no (privileged)	69	90	159	77%
Total	88	192	280	

From the shortlisting process we can determine our statistical disparity is 25/45 = 55% for gender and 16/77 = 21% for BAMEyn which are both below the threshold of 80% , therefore we can conclude that the being shortlisted has an adverse impact on unprivileged groups: female and BAME

Gender/Offer	Offered	Not Offered	Total	Ratio
Male (unprivileged)	18	60	78	23.08%
Female (privileged)	10	192	202	4.95%

BAME/Shortlisted	Offered	Not Offered	Total	Ratio
BAME yes (unprivileged)	8	113	121	6.61%
BAME no (privileged)	20	139	159	12.58%

From the offer process we can determine our statistical disparity is 4.95/23.08 = 21.5% for gender and 6.61/12.58 = 53% for BAMEyn which are both below the threshold of 80%, therefore we can also conclude that getting an offer has an adverse impact on unprivileged groups: female and BAME.

When it comes to showing if our dataset is biased or not, we can use a chi squared test calculated from the formula: $\chi 2 = \frac{\Sigma(O-E)^2}{E}$. Where O is the observed value (above table) and E is the expected value (table below). First we would state our hypothesis tests:

Using hypothesis testing we can state our null hypothesises:

H0: The shortlisting process is not biased towards Male applicants

Gender/Shortlisted Expected	Short listed	Not Shortlisted	Total
Male (unprivileged)	858/35	1872/35	78
Female (privileged)	2222/35	4848/35	202

Gender/Shortlisted $(O-E)^2/E$	Shortlisted	Not Shortlisted
Male (unprivileged)	7.42	3.4
Female (privileged)	2.86	1.31

H0: The shortlisting process is not biased towards non-BAME applicants.

BAME/Shortlisted Expected	Short listed	Not Shortlisted	Total
BAME yes (unprivileged)	1331/35	2904/35	121
BAME no (privileged)	1749/35	3816/35	159

BAME/Shortlisted $(O-E)^2/E$	Short listed	Not Shortlisted
BAME yes (unprivileged)	9.52	4.36
BAME no (privileged)	7.25	3.32

If we use a significant level of 0.05, we have for Gender a chi squared statistic of 14.99 and a p value of 0.0001 and for BAMEyn a chi squared statistic of 24.45 and p value of less than 0.0001. Since these statistics are both < 0.05 we reject the null hypothesis.

We can also implement this proof in python by using a function from scipy.stats called chi2_contingency(). This takes in our observed value table and outputs our chi squared statistic, p value, degrees of freedom and our expected table and get the same results above.

2.2 Data Cleaning

First when attempting to mitigate bias, you need to create an implementation of a regular classifier. Our first step is data cleaning where we drop irrelevant data and anomalies so that our data input is clean and usable. In our data we can see that there are a lot of null slots, therefore, before training our data we would replace these with 0. Furthermore, we would only include relevant columns in our dataset, hence we would drop: ApplicantCode, AcceptNY, JoinNY.

2.3 Data Preparation

Before applying our predictive algorithm, we would split and train our data. We first declare x and y, our independent and dependent (OfferNY) variables respectively. We would then split our data into test and training set, 70% for our training set and 30% for our test set. We would do this using the train_test_split() function where out inputs would be x,y and split ratio and our outputs would be xtest,xtrain,ytest,ytrain. We would use x_train and y_train as our classifier inputs, x_test as the input for prediction, and y_test as a comparison to our prediction output(y_pred). As our classifier algorithm we would use logistic regression as it is the most popular choice when predicting categorical or binary values. For this model, our parameters are C: the inverse regularisation strength, and max_iter: the maximum number of iterations taken for solvers to converge.

2.4 Hyperparameter Tuning

When using our default parameters, we get an accuracy score of 0.91667 however, we may be able to increase our accuracy by hyperparameter tuning to solve the issue of overfitting and underfitting. By using GridSearchCV() function we can hyperparameter tune to find the best score and parameters to reapply to our model. By using new parameters of C: 0.2336 and max_iter: 100 our accuracy score increases to 0.9642. Before and after hyperparameter tuning:

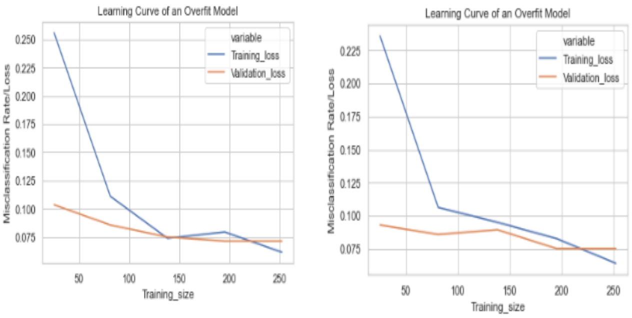
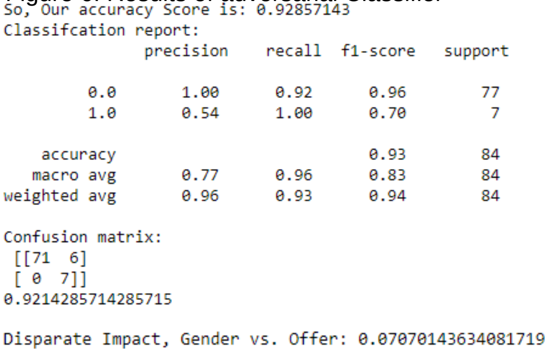


Figure 5: Learning curves before and after hyperparameter tuning

3. REGULAR CLASSIFICATION MODEL EVALUATION

To see how well our chosen algorithms perform we will compare our results from hyperparameter tuning using a classification report, confusion matrix and disparate imapct as evaluation metrics.

Figure 6: Results of adversarial Classifier



```
So, Our accuracy Score is: 0.92857143
Classifcation report:
               precision    recall  f1-score   support

         0.0       1.00      0.92      0.96        77
         1.0       0.54      1.00      0.70         7

    accuracy                           0.93        84
   macro avg       0.77      0.96      0.83        84
weighted avg       0.96      0.93      0.94        84

Confusion matrix:
 [[71  6]
 [ 0  7]]
0.9214285714285715

Disparate Impact, Gender vs. Offer: 0.07070143634081719
```

For evaluation model performance, we use F1 score, and accuracy to analyse, how good/ bias our model is. We can see that our model is accurate however, we can also see that are disparate impact is very low. According to our 4/5th rule this makes the system extremely bias therefore we will not plan on mitigating this. When also using our Demographic parity function we get a high answer of 58.

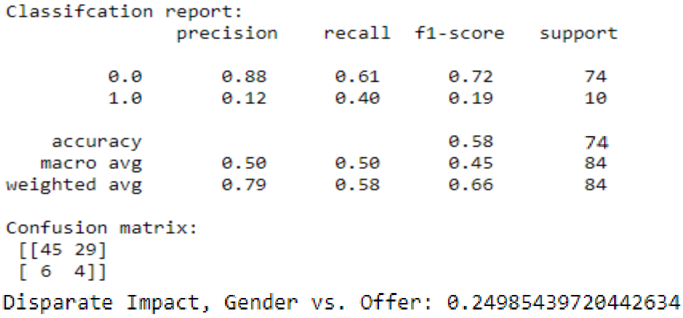
4. MITIGATING BIAS

To mitigate bias, we will use adversarial learning. When implementing this, we first must consider our fairness notion. What makes this data set difficult is how there are two sensitive attributes and as part of fairness, we need to make sure that our algorithm performs impartially for all groups, ie BAME and gender. When mitigating bias using adversarial networks, we first consider our inputs: **X** as all fair contributors to an offer outcome therefore: “Shortlisted”,”Interviewed” and “FemaleONpanel”. We included the third feature as we do not have enough evidence to claim that this would cause much bias. For **Y** we set as what we want to predict : hence “Offer” and **Z** we set as our sensitive attributes “Gender” and “BAME”. Using these input definitions, we design our system based on the following from Brian Hu Zhang’s research paper: *“The input to the network X, … produces a prediction Y..while the adversary tries to model a protected variable Z... The objective is to maximize the predictors’ ability to predict Y while minimizing the adversary’s ability to predict Z.”* Hence, we are trying to use X (unbiased features) to predict Y(Offer) without also being able to predict Z (our sensitive attributes). Now knowing our inputs we can look into how we can achieve fairness: Demographic Parity, Equality of Odds and Equality of Opportunity. Demographic Parity ensures that predicting a job Offer is independent of our sensitive attributes. Equality of odds means that there should be as many job offers for BAME and females as there are for Non Bame and males and Equality of Opportunity applies this vice versa in terms of a Job Rejection. When implementing our mode, we are impolementing an equation:

$$\nabla W\ LP - \mathrm{proj}\nabla W\ LA\ \nabla W\ LP - \alpha\nabla W\ LA$$

Where $\mathrm{proj}\nabla W\ LA\ \nabla W\ LP$ decreases adversary loss, $\alpha\nabla W\ LA$ increases it and $\nabla W\ LP$ can represent our simple classifier. In our code we tune these hyperparameters as “lambda”. To reach fairness in our code, we are following Zhang’s paper with preprocessing and in processing with adversarial learning which includes training our model uniquely.

5 EVALUATIONS/CONCLUSIONS



```
Classifcation report:
               precision    recall  f1-score   support

         0.0       0.88      0.61      0.72        74
         1.0       0.12      0.40      0.19        10

    accuracy                           0.58        74
   macro avg       0.50      0.50      0.45        84
weighted avg       0.79      0.58      0.66        84

Confusion matrix:
 [[45 29]
 [ 6  4]]
Disparate Impact, Gender vs. Offer: 0.24985439720442634
```

Figure 7: Results of adversarial Bias

As demonstrated above, our figures have adjusted dramatically from the regular classifier. As we can first observe, our confusion matrix demonstrates less bias results for females. However, as a result, our f1 score and accuracy have suffered due to mitigating the biasness. Another observation is our disparity score, which is greater than before, however, it still has a long way to pass the 80% threshold for statistical disparity. Furthermore, when using our Demographic parity function we get an answer of 18 much less than the regular classifier.

We wanted to investigate if we could mitigate bias using in processing techniques. Outlining our sensitive attributes and analysing our statistics using the 80% disparity rule and chi squared, we were able to conclude how bias our data was for every outcome. When applying a logistic regression as a regular classifier, this further proved how bias our model was therefore, we planned to use adversarial learning to mitigate our bias. Looking into fairness notions anAs a result, we found that, though we were able to mitigate some bias, we cannot guarantee it overall. A further approach could be to have blind interviews/resumes to make the system fairer.

REFERENCES

[1] Mitigating Bias in AI with AIF360 | by Bryan Truong | Towards Data Science

[2] An Introduction to Fairness and Bias Mitigation with AllenNLP | by Arjun Subramonian Blog (allenai.org)

[3] Copy of Mitigating Unwanted Biases in Word Embeddings with Adversarial Learning.ipynb - Colaboratory (google.com)

[4] https://towardsdatascience.com/parameters-and-hyperparameters- aa609601a9ac 2021

[5] https://www.analyticssteps.com/blogs/what-are-model-parameters- and-evaluation-metrics-used-machine-learning 2021

[6] RaffaeleAns (Raffaele Anselmo) (github.com)