# Bias in AI

By Owen Shaw

## Dataset Analysis

1.1 There are two sensitive attributes in the dataset, Gender and BAMEyn. At all stages (Shortlist,00 Interview and Offer), one group was positively selected far less than expected and the other group was positively selected far more than expected. For BAMEyn, there was significantly less than expected selected at the shortlist stage for one group, however at interview and offer stages, from those who got to the respective stage, this was no longer apparent. This can be shown in the stats following.

For the machine learning (ML) algorithm for predicting job offers, I only considered gender as a sensitive attribute, as there was no evidence for bias present against either ethnicity group.

1.2 Below shows the statistics relevant to the privileged and unprivileged groups.

| **Gender** | Male | Female |
|---|---|---|
| Shortlisted | 48.7% | 24.8% |
| Interviewed | 71.1% | 56.0% |
| Offered | 66.7% | 35.7% |

*Table 1: This shows the percentage of males and females being shortlisted, interviewed, and offered a place of those who got to the respective stage. Males had a significantly higher chance in each.*

| **BAMEyn** | Non-BAME | BAME |
|---|---|---|
| Shortlisted | 43.4% | 15.7% |
| Interviewed | 60.9% | 68.4% |
| Offered | 47.6% | 61.5% |

*Table 2: This shows the percentage of BAME and Non-BAME people passing each stage. At the shortlist stage non-BAME applicants had a significantly higher chance of being shortlisted, however this was not the case at the interview and offer stage.*

1.3 As previously stated, there is no statistical disparity between the privileged and unprivileged groups at the interview and offer stage for BAME and non-BAME candidates, so only the gender values are shown.

As can be seen below, at interview stage, males are significantly more frequently than expected to be interviewed, and females are less frequently than expected to be interviewed. Showing a clear statistical disparity between the privileged and unprivileged group when determining who gains an interview.

| Interview (Gender) | Male | Female | Totals |
|---|---|---|---|
| Interviewed | 27 (23.75) | 28 (31.25) | 55 |
| Not Interviewed | 11 (14.25) | 22 (18.75) | 33 |
| Totals | 38 | 50 | 88 |

*Table 3: This shows the observed values for each gender at the interview stage. In brackets is the expected value, calculated by multiplying the row total by the column total, divided by the overall total.*

As can be seen in the table above, at offer stage, the same statistical disparity is apparent between the two groups.

| Offer (Gender) | Male | Female | Totals |
|---|---|---|---|
| Offered | 18 (13.75) | 10 (14.25) | 28 |
| Not Offered | 9 (13.25) | 18 (13.75) | 27 |
| Totals | 27 | 28 | 55 |

*Table 4: This shows the distribution of offers for males and females, with the expected values in brackets.*

1.4 Next, I tried to prove that the dataset is biased towards the privileged group at the shortlisting stage.

| Shortlist (Gender) | Male | Female | Totals |
|---|---|---|---|
| Shortlisted | 38 (24.51) | 50 (63.49) | 88 |
| Not Shortlisted | 40 (53.49) | 152 (138.51) | 192 |
| Totals | 78 | 202 | 280 |

*Table 5: This shows the shortlisted and not shortlisted values for each gender and in brackets it shows the expected values.*

To do this I used the data in the table above to conduct a Chi-Squared hypothesis test. In this test the null hypothesis is that the expected values (values in brackets) in each category are true, and the alternative hypothesis is that these values are not equal to the expected values. The significance level is 5%, which will be halved when compared against the p-value, as this is a 2-tailed test. The chi-squared test works by squaring the difference between the observed and expected values and summing them and using the knowledge that there is only 1 degree of freedom to calculate a p-value, which is the probability that this result randomly occurred with the null hypothesis being true. The chi-squared value for these data is 15.00, which mean that the p-value is 0.00011, significantly lower than half of 0.05, the significance level, suggesting that we can reject the null-hypothesis and accept the alternative hypothesis. This indicates that the dataset is biased towards the privileged group at the shortlisting stage.

1.5 The chi-squared hypothesis test can be found in the stats.py file using the chi_squared_p_value function. The function uses the crosstab function from the Pandas library to generate a table for the observed values which can then be inputted into the chi squared function. The function chi2_contigency was imported from the stats module from the SciPy package, which was necessary to generate the chi-squared value, p-value, degrees of freedom and the expected table.

## Adversarial Bias Mitigation

2.1. Firstly, after having imported the data in the .xls file into a Pandas data frame, the blank values, which indicate that the candidate did not get to that stage, are replaced with a 0, as if they did not get to the stage then they obviously will not pass to the stage after that. It was not possible to make a classifier using solely candidates that got to the offer stage as there were not enough (55) to make a reliable algorithm.

The data is then split into the data frames X, which contains features, and y, which contains the target values (OfferNY). These data frames are then split into training and testing sets using the train_test_split function from the model_selection module from sklearn. The split is at 70:30 for training: testing and this is stratified so that the same proportion of each group are in the training and testing set.

I used a KNearestNeighbours (KNN) classifier as it had better results overall, when considering accuracy, sensitivity, and specificity, than a Linear Support Vector classifier and other Support Vector classifiers. I specified

$$K = \sqrt{Number\ of\ entries\ in\ training\ set}$$

for the *KNN* classifier as it is a rule of thumb optimum value which should result in a model that neither overfits nor underfits to the data [1]. Using a dynamic value allowed more flexibility in the model later. The data was feature scaled which is necessary for this algorithm, as otherwise certain features would dominate. For this I used Z-Score Normalisation, which is commonly used for *KNN* classifiers [2], which rescales the features to have zero-mean and unit-variance.

The model does not generalise very well to the testing dataset, has a high overall accuracy but struggles to accurately predict someone without a job offer. This can be seen further in the data that follows.

2.2 As can be seen in the tables below, the standard model does well overall with an accuracy of 88%. The table also shows precision and recall which are the percentage of people who gained an offer that were predicted correctly and the percentage of people who did not gain an offer who were predicted correctly, respectively. There is a strong bias towards predicting "No Offer" due to the imbalance of the data, which means that the model is currently overfitting the data.

| Classification Report | Precision | Recall | F1-Score |
|---|---|---|---|
| No Offer | 0.95 | 0.92 | 0.93 |
| Offer | 0.4 | 0.5 | 0.44 |
| Accuracy | | | 0.88 |
| Macro Average | 0.67 | 0.71 | 0.69 |
| Weighted Average | 0.89 | 0.88 | 0.89 |

*Table 6: This shows the classification report for the standard algorithm. There is high precision and recall for those without an offer, however it struggles for those with an offer. There is high accuracy overall however this is due to many more candidates without an offer skewing this value.*

| | | Actual | | |
|---|---|---|---|---|
| | Male Female | No Offer | Offer | Totals |
| Predicted | No Offer | 64% 94% | 18% 0% | 82% 94% |
| | Offer | 5% 5% | 14% 2% | 18% 6% |
| | Totals | 68% 98% | 32% 2% | 100% 100% |

*Table 7: This shows the confusion matrix for the offer stage for both males and females. This further shows the lack of accuracy that can be seen in table 4 and shows the model is predicting offers for males significantly higher than females. The values in this table will be useful for determining whether the chosen fairness criterion is satisfied.*

2.3. The fairness notion chosen was equality of odds [3], which is defined as predictor $\hat{Y}$ satisfies equalized odds with respect to protected attribute A and outcome Y, if $\hat{Y}$ and A are independent conditional on Y, which can be shown in the equation below.

$$P(\hat{Y} = 1 \,|A = 1, Y) = P(\hat{Y} = 1 \,|A = 0, Y)$$

This means that our predictions should be independent of our chosen protected attribute, Gender. This was chosen as it simultaneously focuses on a high quality model whilst aiming to eliminate discrimination. When compared to the other main fairness notions, such as demographic parity which focuses solely on matching the gender distribution, but the accuracy of the model is an afterthought, would not be useful here. Equalized opportunity is better but is only concerned with people who were offered a job, which would not be as useful here as I want the model to be good at predicting people with or without offer.

The adversarial learning component comes in the form of pre-processing, which involves transforming data

before the *ML* algorithm; in-processing, which constrains the *ML* algorithm whilst it learns; post-processing, which aims to make the predictions from the *ML* model fair, without knowing what the *ML* model is doing (black-box), or a combination of those 3. Generally, the best practice is to start the fairness intervention as early as possible in the process and then assess the necessity of further intervention along the data pipeline.

The adversarial learning first consisted of a pre-processing element. The aim of this was to reduce the bias in the data before it reached the classification model. This goal can be shown in the equation below, where Y is the target variable and A is the sensitive attribute.

$$P(Y) = P(Y \mid A = 1) = P(Y \mid A = 0)$$

To do this I used universal resampling. This involves scaling each entry relative to its frequency in comparison to its expected frequency in the data frame. Having found each group's observed and expected frequency I used this equation to scale them as needed, where x is the entry being scaled, N is the frequency, $x_y$ is the target value for x and $x_a$ is the sensitive attribute value for x.

$$Scale(x) = \frac{N_{exp}(Y = x_y, A = x_a)}{N_{obs}(Y = x_y, A = x_a)}$$

Since the values for the scale are decimals, the scale values need to be rounded to the nearest whole number. To minimise the error in this, they are first multiplied by $10^2$, which is a good compromise between accuracy and computational expense. The entries are then scaled as specified into a new data frame which will be used for the *ML* algorithm.

As can be seen in the statistics that follow, pre-processing was the only adversarial learning component needed for this task.

2.4.

| Classification Report | Precision | Recall | F1-Score |
|---|---|---|---|
| No Offer | 0.99 | 1 | 1 |
| Offer | 1 | 0.92 | 0.96 |
| Accuracy | | | 0.99 |
| Macro Average | 1 | 0.96 | 0.98 |
| Weighted Average | 0.99 | 0.99 | 0.99 |

*Table 8: This is the classification report after the adversarial learning component has been added. There is a significant improvement in accuracy and the relevant scores for both "no offer" and "offer".*

| | | Actual | | |
|---|---|---|---|---|
| | Males<br>Females | No Offer | Offer | Totals |
| Predicted | No Offer | 91%<br>90% | 0% 0% | 91%<br>90% |
| | Offer | 3% 0% | 7% 10% | 9% 10% |
| | Totals | 93%<br>90% | 7% 10% | 100%<br>100% |

*Table 9: This is the confusion matrix for both males and females after the adversairal learning component has been added. It can be seen, like in table 6, that the accuracy has improved for both groups, and that the distribution of values are very similar for both groups.*

When comparing the classification report and confusion matrix with their counterparts from before the adversarial component was introduced, they show a significant improvement across all attributes. Accuracy, recall, and precision has improved for both classes. The distribution shown in table 9 achieves very demographic parity, within a small margin of error. When combined, this shows that equalization of odds has been achieved, within a small margin of error.

2.5. Overall, the bias mitigation strategy implemented was mostly successful as the aim of equality of odds was satisfied within a small error. This can be most clearly seen in the comparison between the 2 confusion matrices (Table 7 and 9), which shows the probability of each outcome before and after the adversarial component was implemented. In table 7, the difference in probability of outcome between males and females is very large and this is significantly reduced in table 9. In addition to this, the accuracy, precision and recall of the model is improved, which can be seen when comparing the 2 classification reports (table 6 and table 8).

References:

[1] Band, A., How to find the optimal value of K in KNN?, towardsdatasci-ence.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb, 2020.

[2] Imron, M. A., Prasetyo, B., Improving Algorithm Accuracy K-Nearest Neighbor Using Z-Score Normalization and Particle Swarm Optimization to Predict Customer Churn, JOSCEX, vol. 1, no. 1, pp. 56-62, 2020.

[3] Hardt, M., Price, E., Srebro, N., Equality of Opportunity in Supervised Learning, arXiv:1610.02413, 2016.