Abbildung 1:https://www.theguardian.com/higher-education-network/2016/jan/19/how-much-does-your-university-do-for-racial-equality

Pa.06 Case Study

Artificial Intelligence FS 2023

Nobel, Gabriel | von Wartburg, Rebekka

Table of Content

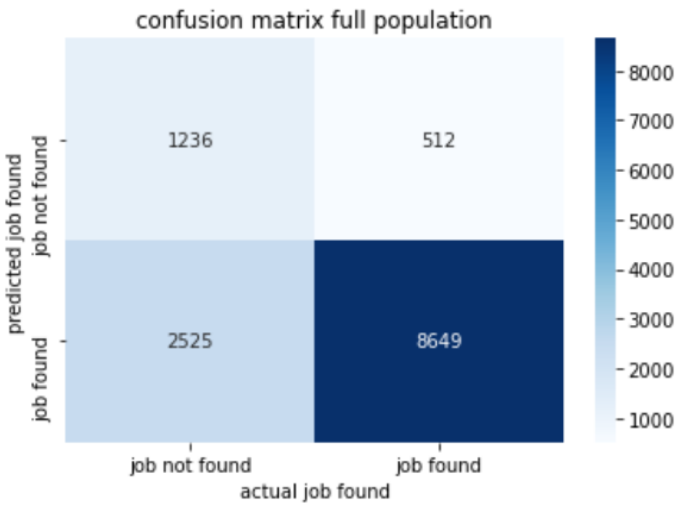
1 How good is the trained model?

We now want to evaluate the performance of the model on the testing data.

1.1 Accuracy of the model

As a preliminary test, calculate the accuracy of the model, using the standard threshold of 0.5 for the score (model.predict() function).

➔ Accuracy of the model: 0.7649744621575607



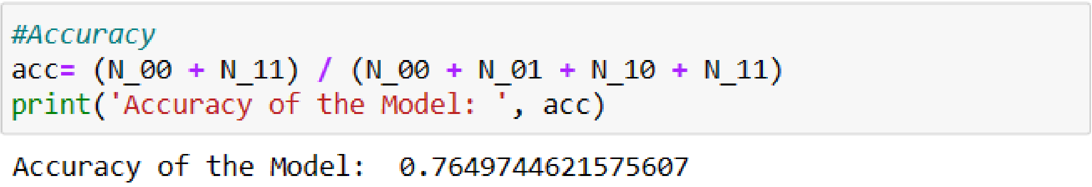
$N11$: 8649

$N01$: 512

$N10$: 2525

$N00$: 1236

Figure 1: Confusion matrix full population



```
#Accuracy
acc= (N_00 + N_11) / (N_00 + N_01 + N_10 + N_11)
print('Accuracy of the Model: ', acc)
```

```
Accuracy of the Model:  0.7649744621575607
```

1.2 Calibration plot

Check the calibration of the model. Hint: Split the scores into 10 bins {[0,0.1),[0.1,0.2),…,[0.8,0.9),[0.9,1]}. For each bin i=1,…,10 , calculate

- The average score of all individuals in this bin: x_i
- The average reemployment rate of all individuals in this bin: y_i

Plot a calibration plot with these (x_i, y_i)-pairs, showing the expected reemployment rate and the actual reemployment rate where the actual reemployment rate also shows the 95% confidence interval. Add a line to the calibration plot that shows what perfect calibration would look like. Write a summary about your findings on the calibration of the prediction model.

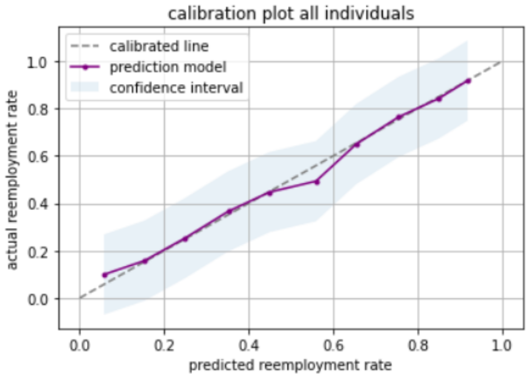


Figure 2: Calibration plot for all individuals with xi = average score and yi = average reemployment rate with a confidence interval of 95%.

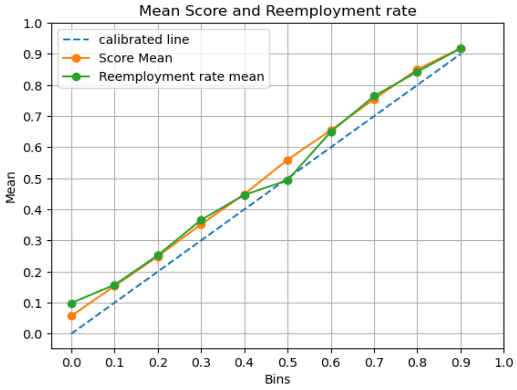


Figure 3: Plot of mean values per bin

1.3 Summary

Overall, all (x_i, y_i)- pairs are within the confidence interval of 95%. Eight of the total ten data-points are even almost on the perfect calibration line. Thus, the model is well calibrated for all individuals. The biggest outlier is at around 50% which could indicate that the model has a hard time predicting edge cases.

2 How well does the model work for the two groups?

Now let's see how well the model works for the two groups of black and white job seekers. We'll again assume that to make binary decisions, we will use the standard threshold of 0.5 (model.predict() function).

2.1 Distributions

Plot the score distribution for both groups. Describe and interpret the results.

2.1.1 Score distribution Caucasians and African- American

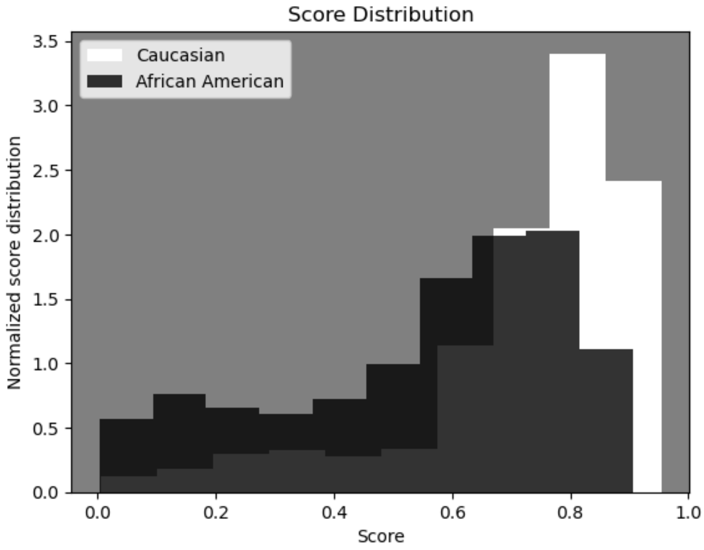


Figure 4: Score Distribution for both sensitive groups

2.1.2 Description and interpretation of the results

The distribution plot of the two groups shows that the caucasian people have significantly more values in the high score range and thus a higher probability to get a job coach than african-american people.

2.2 Check statistically difference of the actual reemployment rate

Is calibration-between-groups achieved?

Score bin value with statistical difference in reemployment rate:

Check for every score bin ({[0,0.1), [0,0.2),…,[0.9,1]} whether there's a statistically significant difference in the actual reemployment rates for the two groups. Note: The groups might have a different expected reemployment rate (i.e., different average scores for the same bin). To simplify this comparison, we'll ignore this potential difference here and just compare actual reemployment rates for each bin.

$(0.0, 0.1] \; -> \; 0.94$

$(0.1, 0.2] \; -> \; 0.98$

$(0.2, 0.3] \; -> \; 0.94$

$(0.3, 0.4] \; -> \; 0.84$

$(0.4, 0.5] \; -> \; 0.85$

$(0.5, 0.6] \; -> \; 0.97$

$(0.6, 0.7] \; -> \; 0.94$

$(0.7, 0.8] \; -> \; 0.96$

$(0.8, 0.9] \; -> \; 0.97$

$(0.9, 1.0] \; -> \; 0.75$

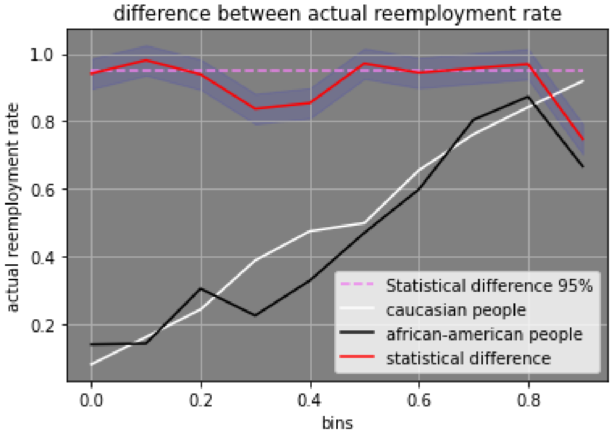


Figure 5: Statistical significance of reemployment rate between groups

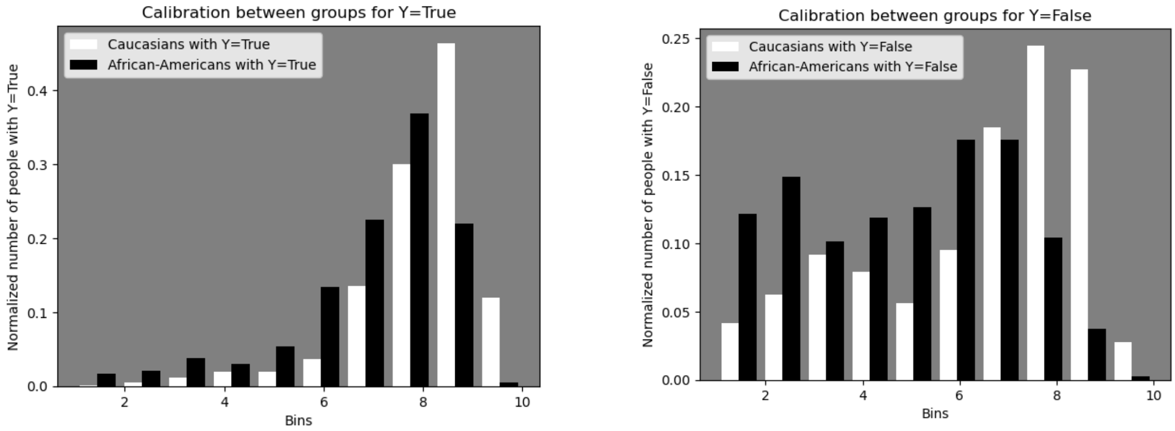


Figure 6: Calibration between groups for Y=True / Y=False

2.3 Calibrations of the two groups

The calibration plot should show the expected reemployment rates on the x-axis (different for the two groups) and the actual reemployment rates on the y-axis (also different for the two groups). Also show the 95% confidence intervals for the actual reemployment rates for both groups. Add a line that shows what perfect calibration would look like. Describe and interpret the results.is. In addition, the confidence interval of 95% is shown to indicate the quality of the calibration.

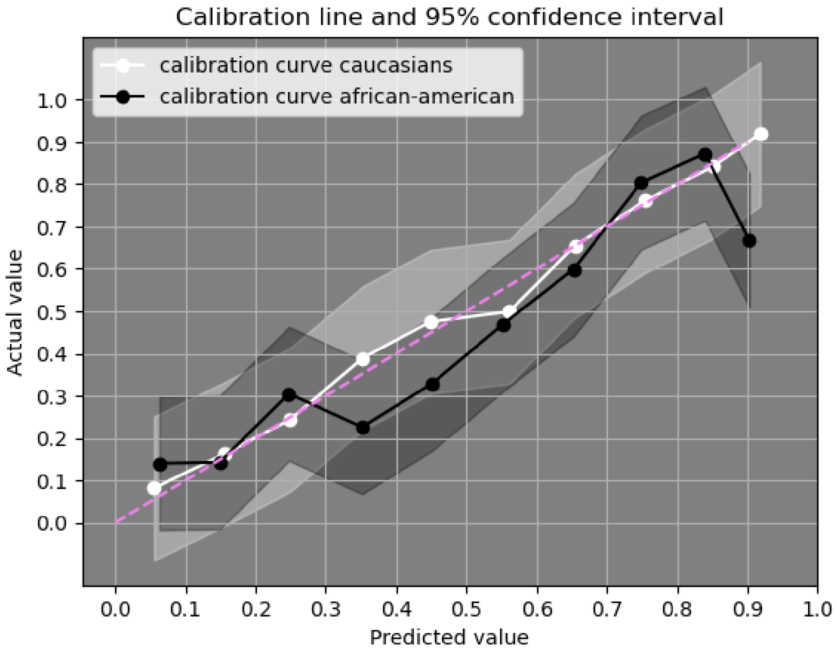


Figure 7: Calibration plot for both groups

2.3.1 Description/ Interpretation of the result

The plot shows the calibration curves of both caucasians- and african-american people with 95% confidence interval and a perfect calibration line. Three things are interesting in this plot:

1. At the lower end african-american people are more likely to be predicted with this score
2. At the upper end the Caucasian people are much more likely to be predicted
3. At the place where Figure 1 showed the most problems we can see a big discrepancy between groups

In conclusion we argue that the calibration between groups is not given, and the model has been trained without consideration to the sensitive attributes which can lead to unethical scores.

2.4 Metrics Comparison of the two groups

Compare the following metrics for both groups: base rate (BR), positive rate (PR), true positiver-rate (TPR), false negative rate (FNR), false positive rate (FPR) and true negative rate (TNR). Is there a statistically significant difference for the metrics between the groups? Describe and interpret the results.

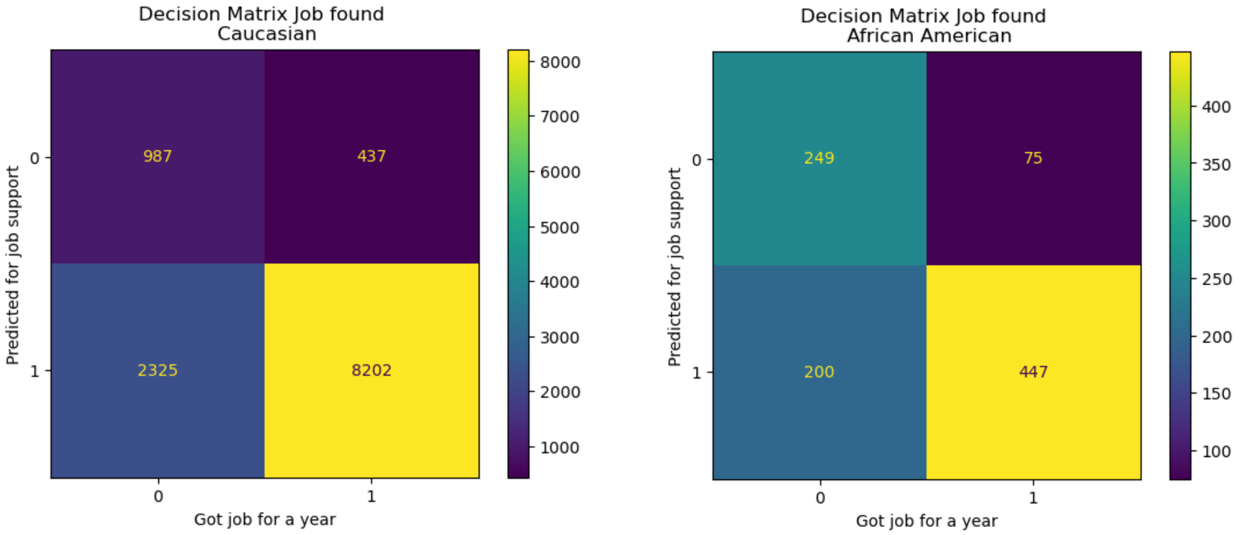
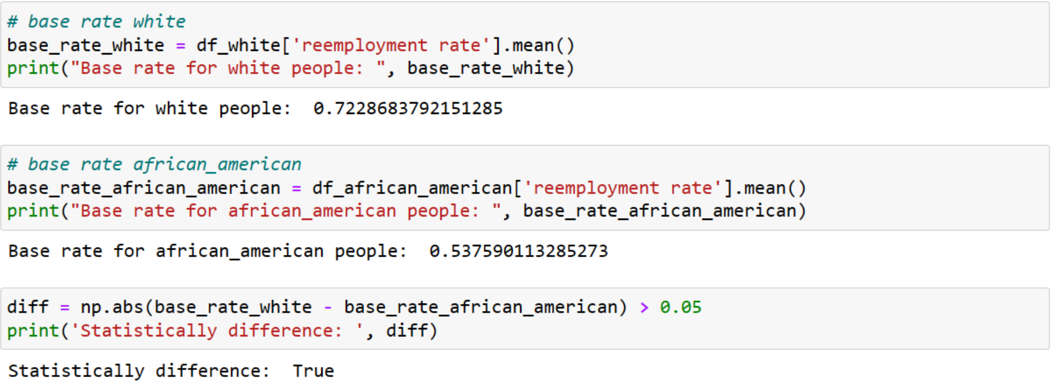


Figure 8: Decision matrices for each group

2.4.1 Base Rate (BR)



```
# base rate white
base_rate_white = df_white['reemployment rate'].mean()
print("Base rate for white people: ", base_rate_white)
```

```
Base rate for white people:  0.7228683792151285
```

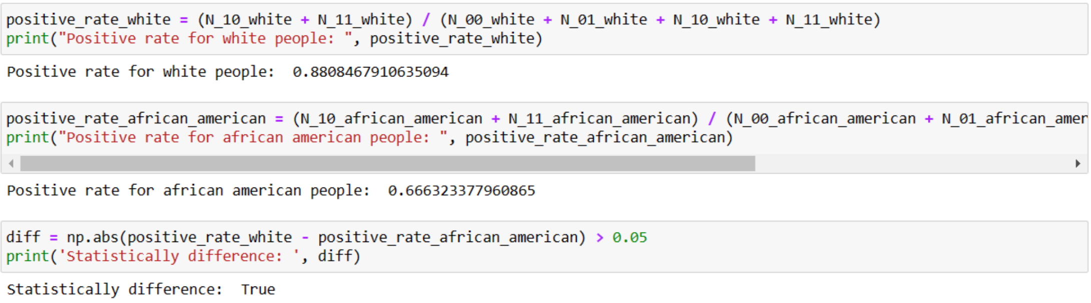
```
# base rate african_american
base_rate_african_american = df_african_american['reemployment rate'].mean()
print("Base rate for african_american people: ", base_rate_african_american)
```

```
Base rate for african_american people:  0.537590113285273
```

```
diff = np.abs(base_rate_white - base_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

2.4.2 Positive Rate (PR)



```
positive_rate_white = (N_10_white + N_11_white) / (N_00_white + N_01_white + N_10_white + N_11_white)
print("Positive rate for white people: ", positive_rate_white)
```

```
Positive rate for white people:  0.8808467910635094
```

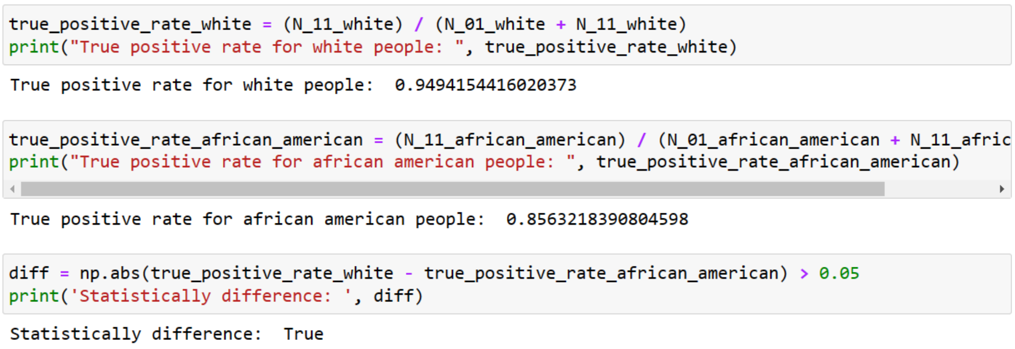
```
positive_rate_african_american = (N_10_african_american + N_11_african_american) / (N_00_african_american + N_01_african_amer
print("Positive rate for african american people: ", positive_rate_african_american)
```

```
Positive rate for african american people:  0.666323377960865
```

```
diff = np.abs(positive_rate_white - positive_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

2.4.3 True Positive Rate (TPR)



```
true_positive_rate_white = (N_11_white) / (N_01_white + N_11_white)
print("True positive rate for white people: ", true_positive_rate_white)
```

```
True positive rate for white people:  0.9494154416020373
```

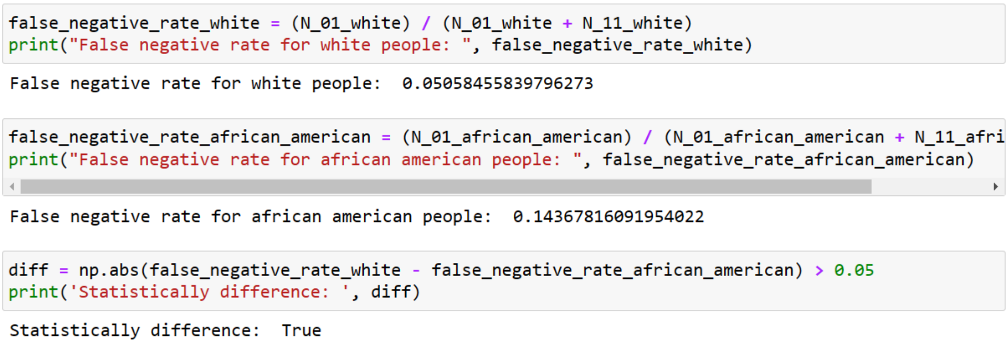
```
true_positive_rate_african_american = (N_11_african_american) / (N_01_african_american + N_11_afric
print("True positive rate for african american people: ", true_positive_rate_african_american)
```

```
True positive rate for african american people:  0.8563218390804598
```

```
diff = np.abs(true_positive_rate_white - true_positive_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

2.4.4 False Negative Rate (FNR)



```
false_negative_rate_white = (N_01_white) / (N_01_white + N_11_white)
print("False negative rate for white people: ", false_negative_rate_white)
```

```
False negative rate for white people:  0.05058455839796273
```

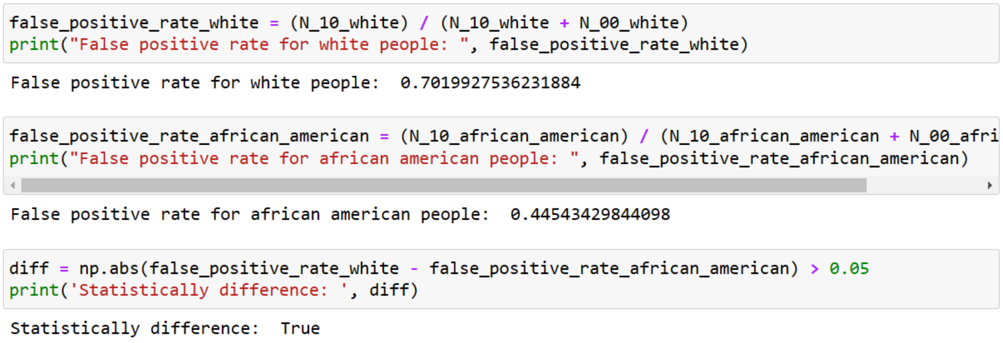
```
false_negative_rate_african_american = (N_01_african_american) / (N_01_african_american + N_11_afri
print("False negative rate for african american people: ", false_negative_rate_african_american)
```

```
False negative rate for african american people:  0.14367816091954022
```

```
diff = np.abs(false_negative_rate_white - false_negative_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

2.4.5 False Positive Rate (FPR)



```
false_positive_rate_white = (N_10_white) / (N_10_white + N_00_white)
print("False positive rate for white people: ", false_positive_rate_white)
```

```
False positive rate for white people:  0.7019927536231884
```

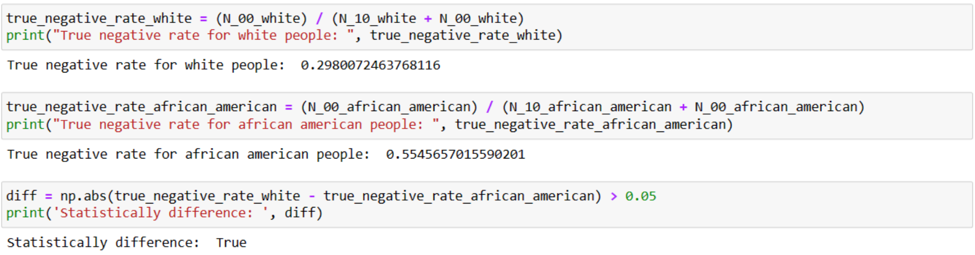
```
false_positive_rate_african_american = (N_10_african_american) / (N_10_african_american + N_00_afri
print("False positive rate for african american people: ", false_positive_rate_african_american)
```

```
False positive rate for african american people:  0.44543429844098
```

```
diff = np.abs(false_positive_rate_white - false_positive_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

2.4.6 True Negative Rate (TNR)



```
true_negative_rate_white = (N_00_white) / (N_10_white + N_00_white)
print("True negative rate for white people: ", true_negative_rate_white)
```

```
True negative rate for white people:  0.2980072463768116
```

```
true_negative_rate_african_american = (N_00_african_american) / (N_10_african_american + N_00_african_american)
print("True negative rate for african american people: ", true_negative_rate_african_american)
```

```
True negative rate for african american people:  0.5545657015590201
```

```
diff = np.abs(true_negative_rate_white - true_negative_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

2.4.7 Description/ Interpretation of the results

All metrics used for comparison show a statistically significant difference.

If all fairness metrics show a statistically significant difference, this indicates that the model has differences with respect to the respective group. These differences are no longer random due to the number of unequal metrics but are fixed in the data in such a way that the model acts unfairly.

3 Utility of the decision maker

As the unemployment agency, you have to be careful about how you spend taxpayers' money: One of your goals is to reduce the amount of money that you spend. Based on this goal, what's the utility matrix of the decision maker in monetary terms for the two years after each decision? Explain your choice to the public.

3.1 Decision maker utility matrix V

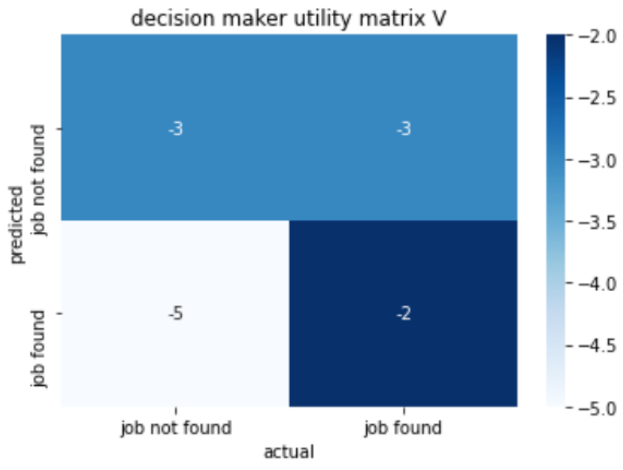


Figure 9: Decision maker utility matrix

3.1.1 Explanation of the choice to the public

Explanation of the selected values of the decision maker utility matrix V:

- D=0 is always -3 because the taxpayers have to pay 30k on average on a long time unemployed person
- For D=1 and Y=0 the utility is -5 because the taxpayers had to pay 30k for being unemployed and 20k for a failed coaching program
- For D=1 and Y=1 the utility is -2 because the taxpayers don't need to pay following 30k a year, but the coaching costed 20k

Given is:

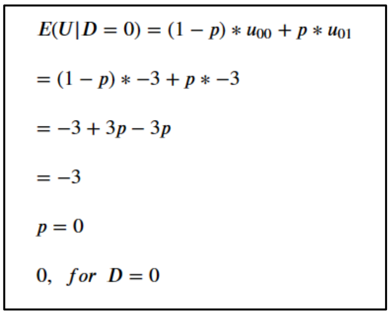
- D = 1, Y = 0 > D = 0, Y = 0
- D = 1, Y = 1 > D = 0, Y = 1

3.2 Threshold that maximizes the decision maker's utility

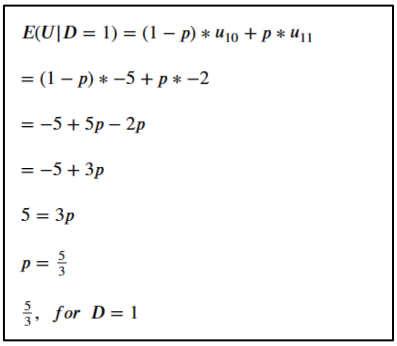
What would be the single threshold that maximizes the decision maker's utility?

3.2.1 Calculating threshold theoretically

What should it be theoretically, assuming that the predicted probability was equal to the true probability?



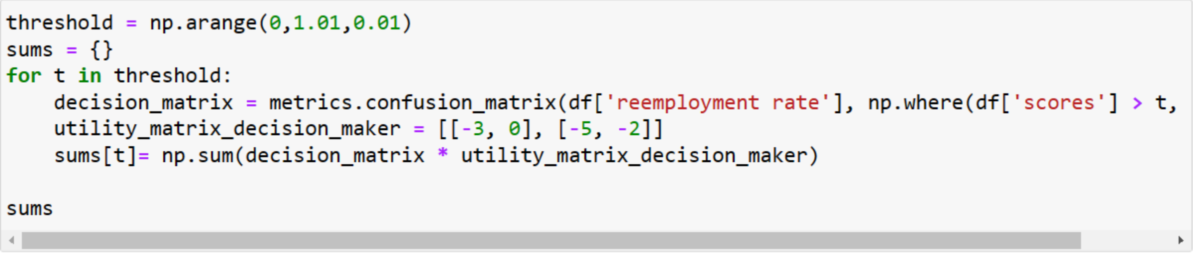
$$E(U|D=0) = (1-p)*u_{00} + p*u_{01}$$
$$= (1-p)*-3 + p*-3$$
$$= -3 + 3p - 3p$$
$$= -3$$
$$p = 0$$
$$0,\ \ for\ \ D = 0$$



$$E(U|D=1) = (1-p)*u_{10} + p*u_{11}$$
$$= (1-p)*-5 + p*-2$$
$$= -5 + 5p - 2p$$
$$= -5 + 3p$$
$$5 = 3p$$
$$p = \frac{5}{3}$$
$$\frac{5}{3},\ \ for\ \ D = 1$$

3.2.2 Maximizing threshold programmatically

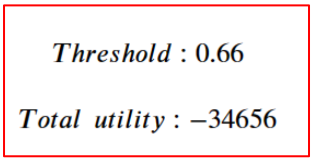
What is the one that you find to maximize the total utility on the testing data, testing the thresholds {0,0.01,0.02,…0.98,0.99,1}? What is the total utility for this threshold?



```
threshold = np.arange(0,1.01,0.01)
sums = {}
for t in threshold:
    decision_matrix = metrics.confusion_matrix(df['reemployment rate'], np.where(df['scores'] > t,
    utility_matrix_decision_maker = [[-3, 0], [-5, -2]]
    sums[t]= np.sum(decision_matrix * utility_matrix_decision_maker)

sums
```

3.2.2.1 Total Utility of the finding



$$Threshold : 0.66$$
$$Total\ \ utility : -34656$$

4 Moral Analysis

As a government agency, you (and the people who you serve) don't only care about monetary values though – you also want to be fair to decision subjects. You therefore define a fairness criterion that you use to evaluate how fair your decision-making system is. For this, you follow the pattern of defining a utility matrix V, the sensitive attribute A and the justifier J with the relevant value j. You'll have to carefully choose these variables based on the application context.

V: What's the utility matrix of the decision subjects? Justify your choice to the public.

A: The relevant groups are given in the dataset: Black and white job seekers.

J: Is there a justifier J that (from the decision subject perspective) justifies differences in the utilities the individuals receive from the decision-making system (i.e., the process that distributes the ??)? Justify your choice to the public.

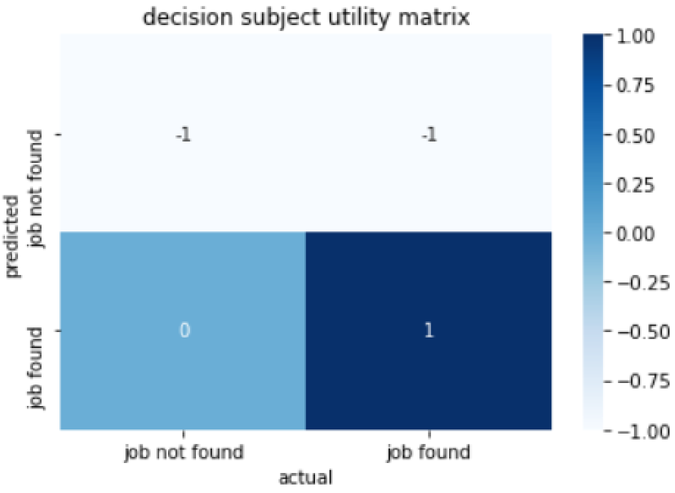


Figure 10: Decision subject utility matrix

The decision subject matrix is as follows:

- If D=0 it always is bad / depressing for the unemployed person, therefore his utility is -1
- If D=1 and Y=0 the person is not employed after a year but got the chance to work with a coach so the utility is 0
- If D=1 and Y=1 the unemployed person has a job and is happy so the utility is 1

4.1.1 Justification of the choice of J to the public

An appropriate justifier J could be DIS (Disability recode) which has following values:

- 1: With a disability
- 2: Without a disability

The reasoning behind the selected justifier is that people are not responsible for their disability and still have the ethical right to be equal in society. For this they need special support in everyday life, e.g., in finding a job.

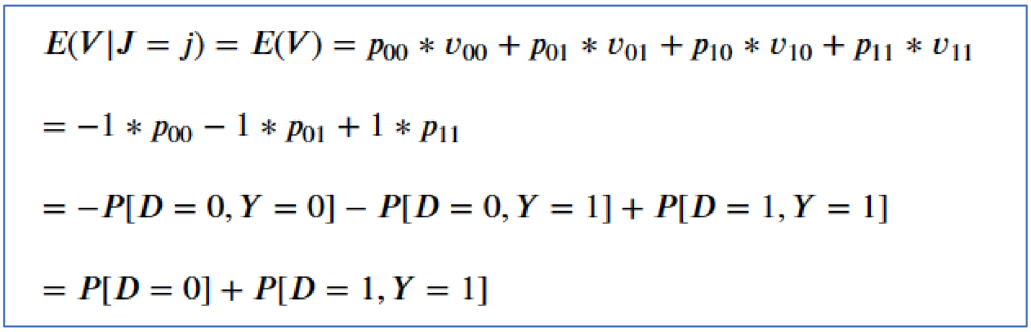
4.2 Fairness Criterion

Based on the decision subject utility matrix V, the sensitive attribute A and the specified justifier, a fairness criterion is to be defined.

Fairness criterion:

- Write down the resulting fairness criterion as an equation and simplify it as much as possible. Is your fairness criterion one of the well-known criteria derived from the decision matrix?
- Is the fairness criterion fulfilled under the experimentally best decision rule from task 3? Measure the metric for both groups and check for statistical significance.

4.2.1 Equation of the resulting fairness criterion



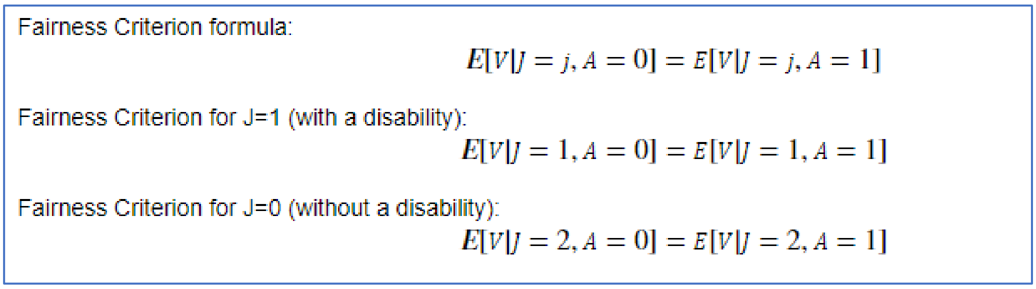
$$E(V|J=j) = E(V) = p_{00} * v_{00} + p_{01} * v_{01} + p_{10} * v_{10} + p_{11} * v_{11}$$

$$= -1 * p_{00} - 1 * p_{01} + 1 * p_{11}$$

$$= -P[D=0, Y=0] - P[D=0, Y=1] + P[D=1, Y=1]$$

$$= P[D=0] + P[D=1, Y=1]$$

Figure 11: Equation of fairness criterion



Fairness Criterion formula:

$$E[V|J=j, A=0] = E[V|J=j, A=1]$$

Fairness Criterion for J=1 (with a disability):

$$E[V|J=1, A=0] = E[V|J=1, A=1]$$

Fairness Criterion for J=0 (without a disability):

$$E[V|J=2, A=0] = E[V|J=2, A=1]$$

Figure 12: Fairness criterion formula with justifier

4.2.1.1 Explanation (Result one of the well-known criteria derived from the decision matrix?)

In our opinion it is not a well-known criteria but is nearest to sufficiency with **j ∈ {1,2}**

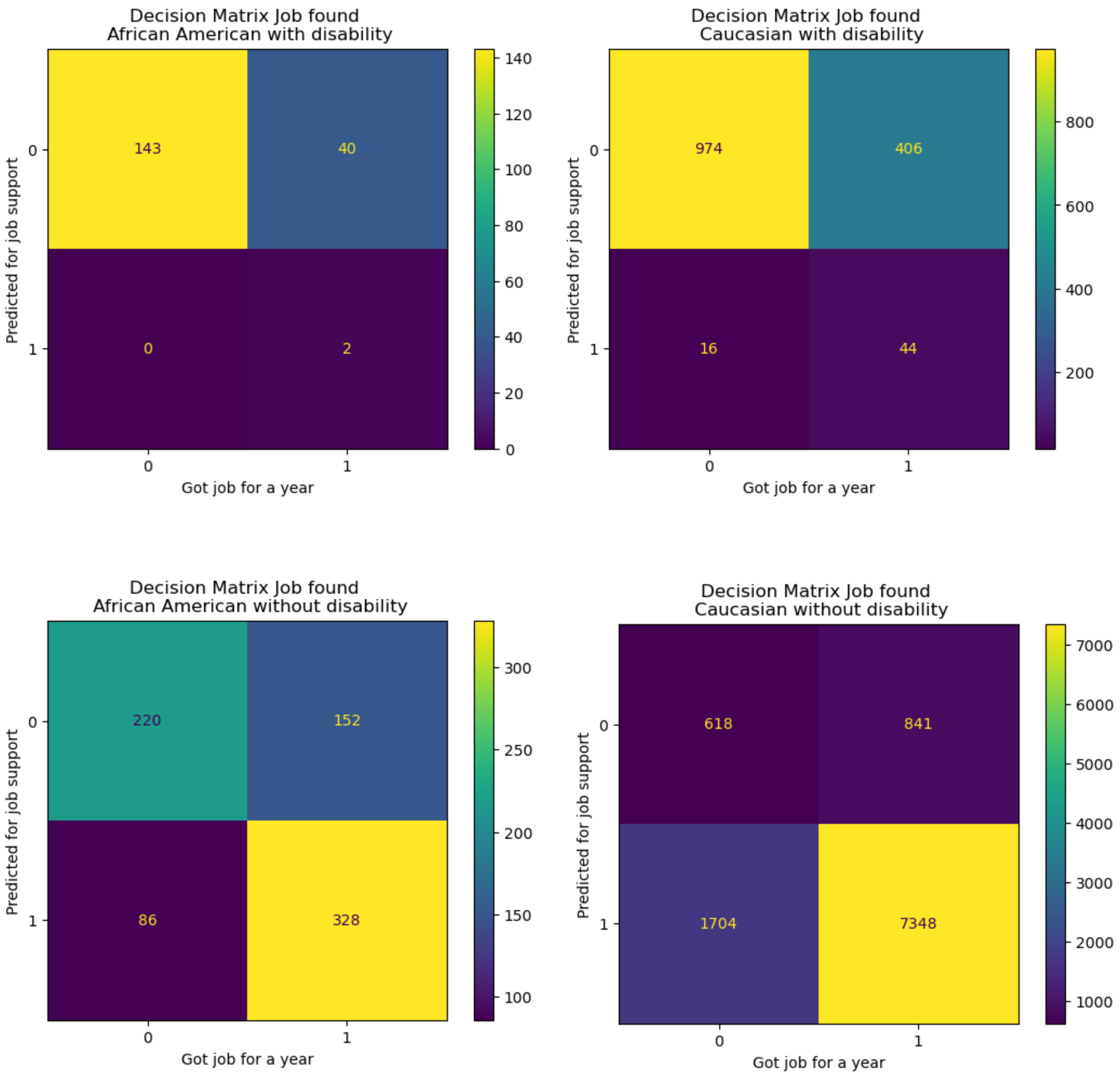
4.2.2 Fairness criterion fulfilled

False Omission Rate with J=1, A=Caucasian is: 0.29
False Omission Rate with J=1, A=African American is: 0.22
Statistical Significance (95%): True

Positive predictive value with J=1, A=Caucasian is: 0.73
Positive predictive value with J=1, A=African American is: 1.0
Statistical Significance (95%): True

False Omission Rate with J=2, A=Caucasian is: 0.58
False Omission Rate with J=2, A=African American is: 0.41
Statistical Significance (95%): True

Positive predictive value with J=2, A=Caucasian is: 0.81
Positive predictive value with J=2, A=African American is: 0.79
Statistical Significance (95%): False


Decision Matrix Job found African American with disability

Decision Matrix Job found Caucasian with disability

Decision Matrix Job found African American without disability

Decision Matrix Job found Caucasian without disability

Figure 13: Decision matrices by sensitive attribute and justifier

5 Trade-off between DM utility and DS fairness

Use the FairnessLab to create a Pareto front and pick (group-specific) thresholds that you could justify to the public.

Here are the steps you should take:

- Create a dataset that fulfills the requirements of the FairnessLab (check the FAQ section of the FairnessLab).
- Feed this dataset to the FairnessLab and configure it, so that it represents your DM utility matrix (from task 3) and your fairness score (from task 4).
- Check the Pareto plot for 101 thresholds per group (“How many thresholds do you want to test for each group?”) and explore different points (i.e., threshold combinations) in the plot. Choose one that offers a good trade-off in your opinion.
- What threshold did you end up choosing? Justify your choice to the public.

5.1 Check of the pareto plot

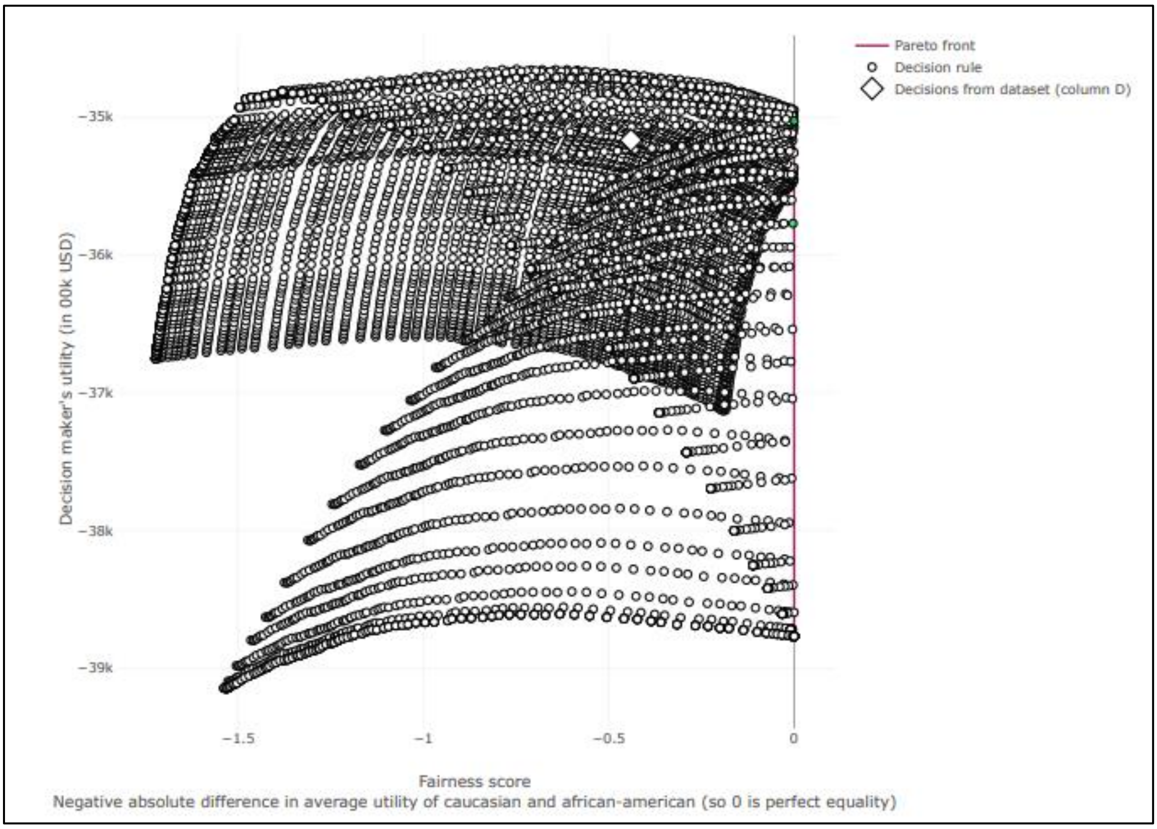
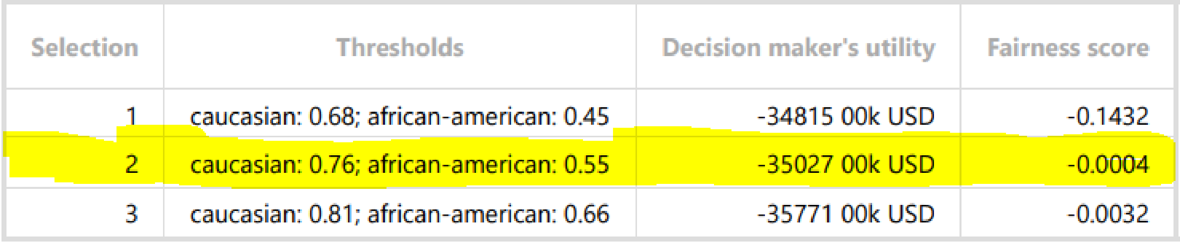


Figure 14: 101 Pareto plot

The 101 pareto plot is very limited. Selection 2 has no benefit for the decision maker and low benefit for the decision subject and is overall the best decision rule.



Selection	Thresholds	Decision maker's utility	Fairness score
1	caucasian: 0.68; african-american: 0.45	-34815 00k USD	-0.1432
2	caucasian: 0.76; african-american: 0.55	-35027 00k USD	-0.0004
3	caucasian: 0.81; african-american: 0.66	-35771 00k USD	-0.0032

Figure 15: Selected decision rule

5.2 Chosen Threshold

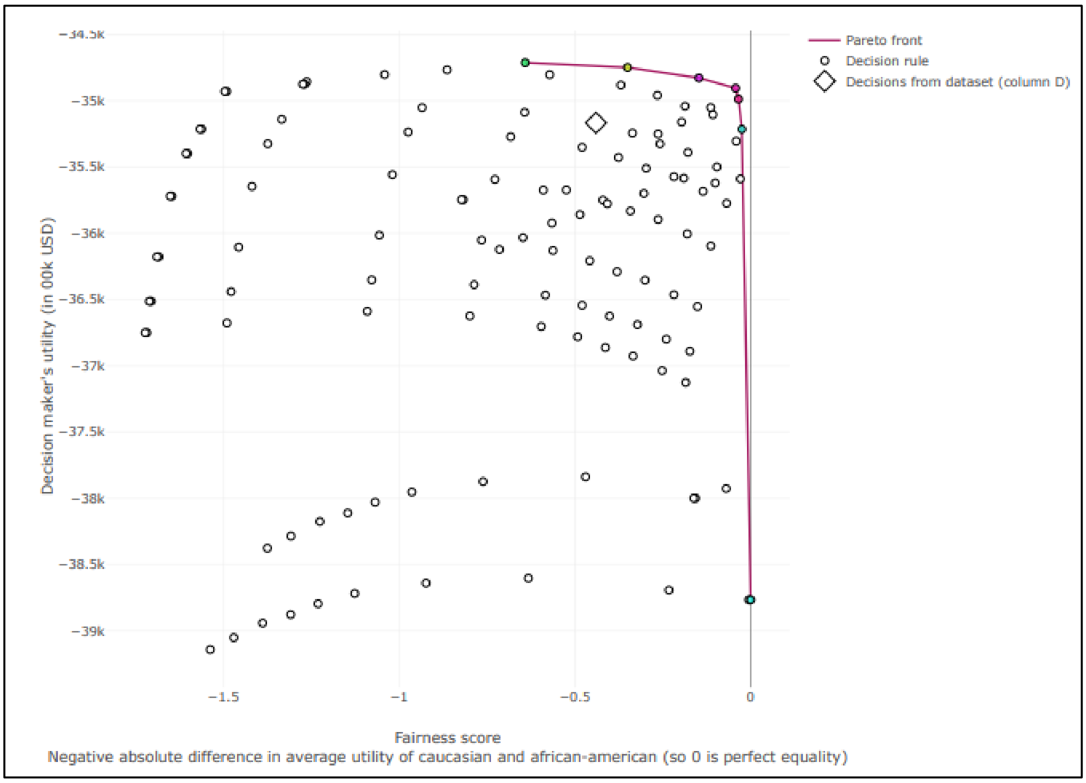
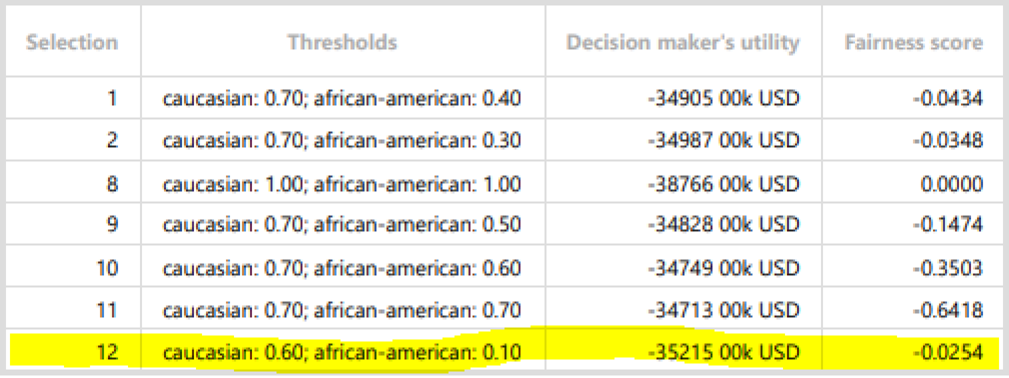


Figure 16: Pareto plot with threshold 11



Selection	Thresholds	Decision maker's utility	Fairness score
1	caucasian: 0.70; african-american: 0.40	-34905 00k USD	-0.0434
2	caucasian: 0.70; african-american: 0.30	-34987 00k USD	-0.0348
8	caucasian: 1.00; african-american: 1.00	-38766 00k USD	0.0000
9	caucasian: 0.70; african-american: 0.50	-34828 00k USD	-0.1474
10	caucasian: 0.70; african-american: 0.60	-34749 00k USD	-0.3503
11	caucasian: 0.70; african-american: 0.70	-34713 00k USD	-0.6418
12	caucasian: 0.60; african-american: 0.10	-35215 00k USD	-0.0254

Figure 17: Selected decision rule

Our chosen threshold is 11, because of two reasons:

- The pareto plot is less noisy and returns a better decision rule.
- 11 is the default threshold for our chosen dataset (thoughts have certainly been made).

6 Deployment four years later

The model is now deployed in the same state for the next four years. We now want to evaluate how well the model does at that point (so four years later).

Go through tasks 1 through 5 again but use the data from 2018 (full dataset, without train-test-split) to evaluate the model from 2014. For tasks 3 and 4, the DM utility matrix and fairness criterion will stay the same. What you should reevaluate here is the optimal threshold for the DM on the 2018 dataset and whether the fairness criterion is fulfilled. Think about why things changed if they changed. You can of course copy and paste your code from tasks 1 through 5.

What's important here is the reinterpretation of the results.

6.1 How good is the trained model?

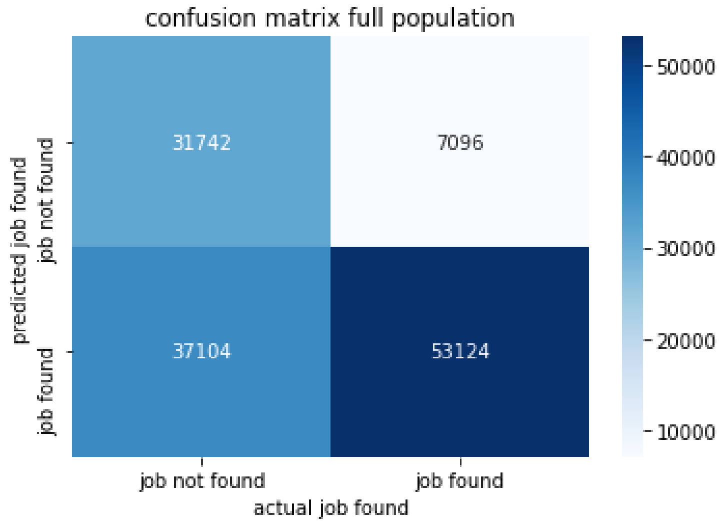
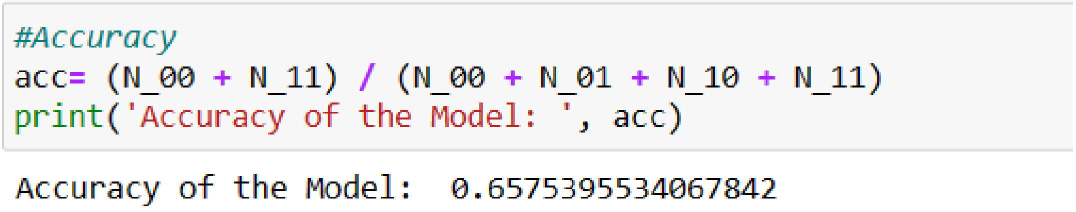


Figure 18: Confusion matrix full population



```
#Accuracy
acc= (N_00 + N_11) / (N_00 + N_01 + N_10 + N_11)
print('Accuracy of the Model: ', acc)

Accuracy of the Model:  0.6575395534067842
```

The accuracy is 0.66 that is almost 0.1 worse than before.

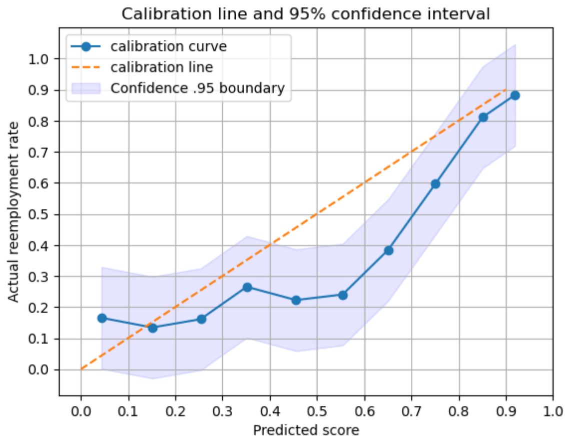


Figure 19: Calibration plot for all individuals with xi = average score and yi = average reemployment rate with a confi-dence interval of 95%.

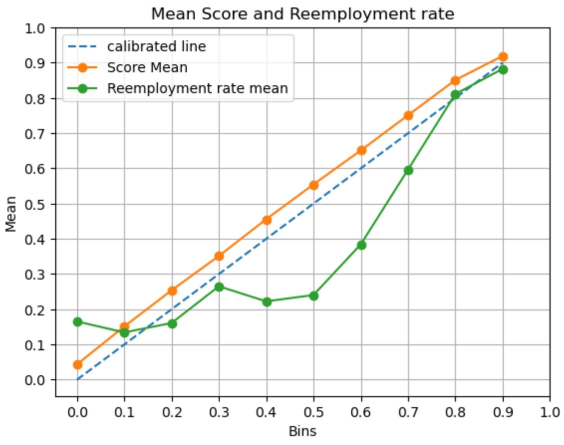


Figure 20: Plot of mean values per bin

Summary: The calibration line is much worse than with the previous data. This means that the prediction model must be retrained. It is interesting that the curve colapsed the most around 0.5. The whole line looks like an amplification of the last graph.

6.2 How well does the model work for the two groups?

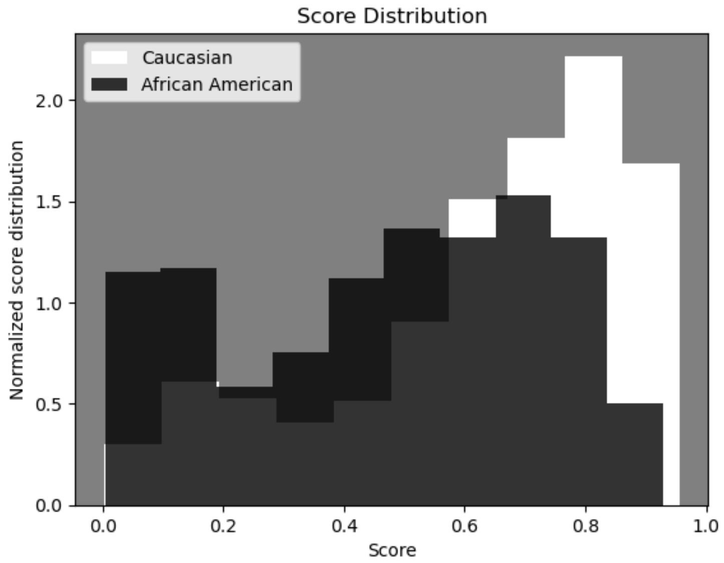
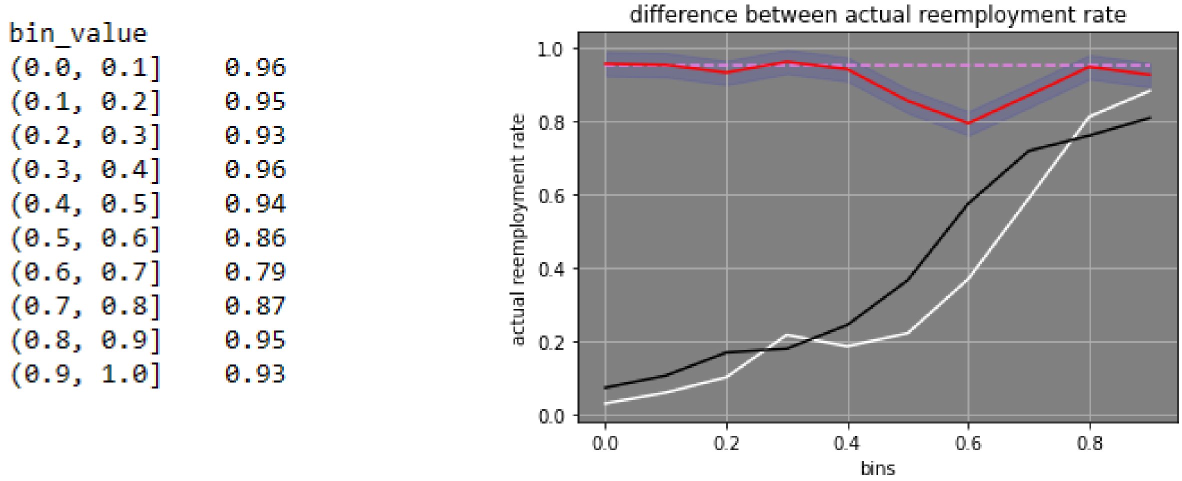


Figure 21: Score Distribution for both sensitive groups

It seems like the distribution for both groups is not given but a little bit better than before. The white people still have a higher probability to get a coach than black people.



```
bin_value
(0.0, 0.1]    0.96
(0.1, 0.2]    0.95
(0.2, 0.3]    0.93
(0.3, 0.4]    0.96
(0.4, 0.5]    0.94
(0.5, 0.6]    0.86
(0.6, 0.7]    0.79
(0.7, 0.8]    0.87
(0.8, 0.9]    0.95
(0.9, 1.0]    0.93
```

Figure 22: Statistical significance of reemployment rate between groups

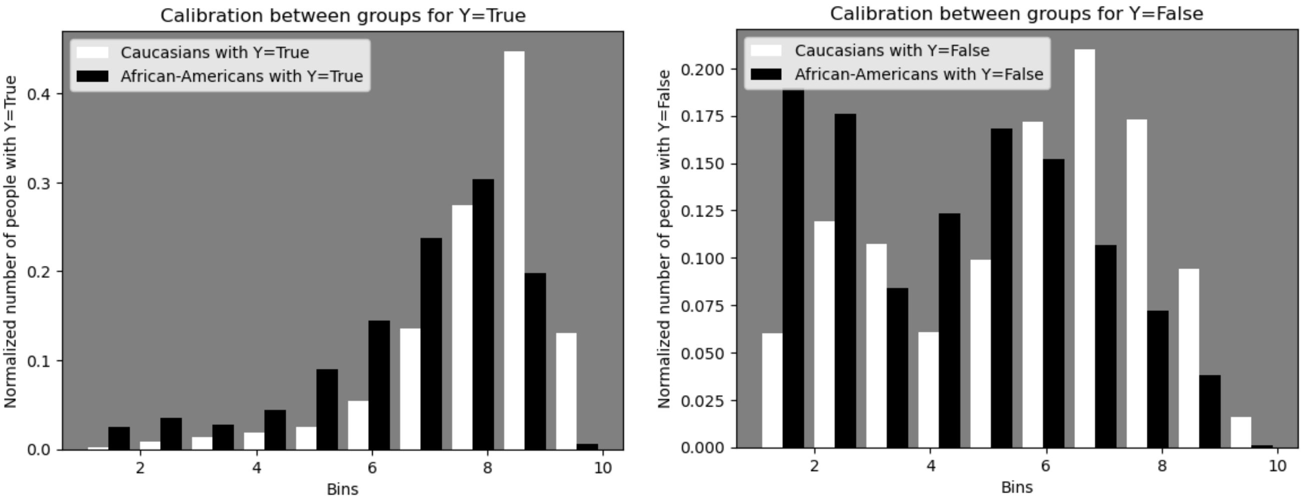


Figure 23: Calibration between groups for Y=True / Y=False

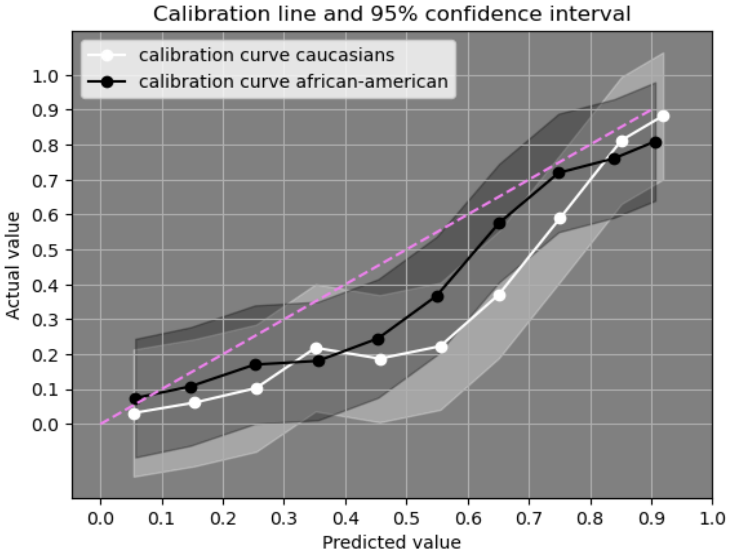
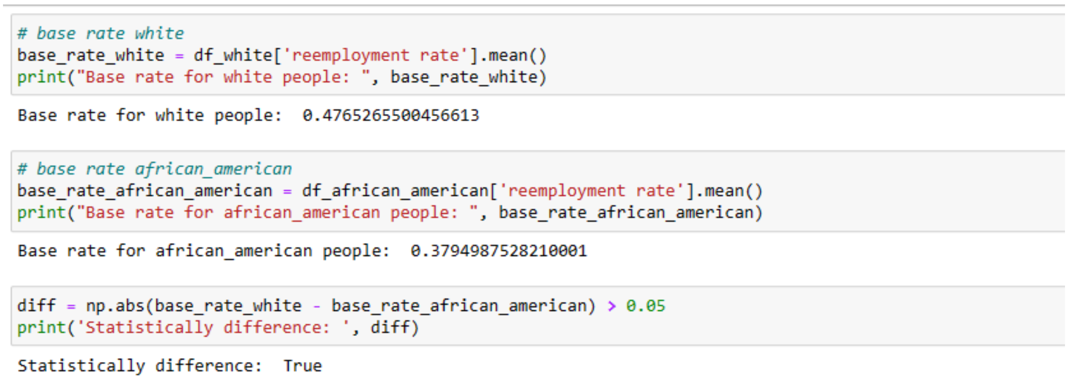


Figure 24: Calibration plot for both groups

The general prediction is much worse for the new data, which indicates that the significance of the features has changed over the years.

Metrics Comparison of the two groups:

Base Rate (BR)



```
# base rate white
base_rate_white = df_white['reemployment rate'].mean()
print("Base rate for white people: ", base_rate_white)
```

```
Base rate for white people:  0.4765265500456613
```

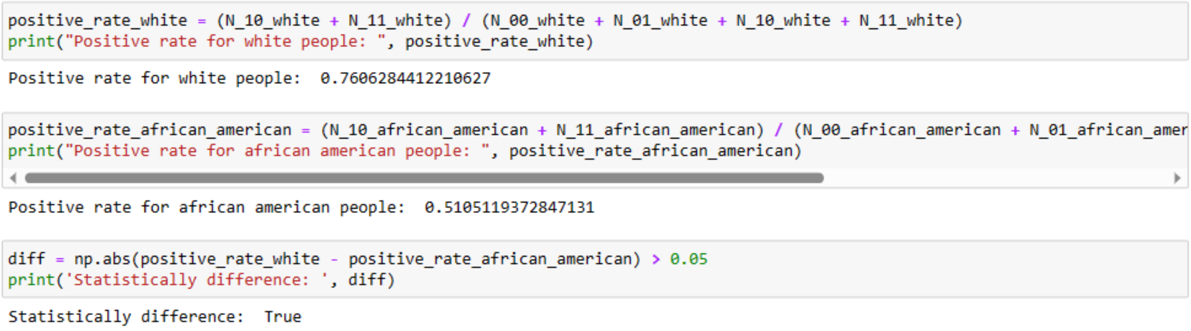
```
# base rate african_american
base_rate_african_american = df_african_american['reemployment rate'].mean()
print("Base rate for african_american people: ", base_rate_african_american)
```

```
Base rate for african_american people:  0.3794987528210001
```

```
diff = np.abs(base_rate_white - base_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

Positive Rate (PR)



```
positive_rate_white = (N_10_white + N_11_white) / (N_00_white + N_01_white + N_10_white + N_11_white)
print("Positive rate for white people: ", positive_rate_white)
```

```
Positive rate for white people:  0.7606284412210627
```

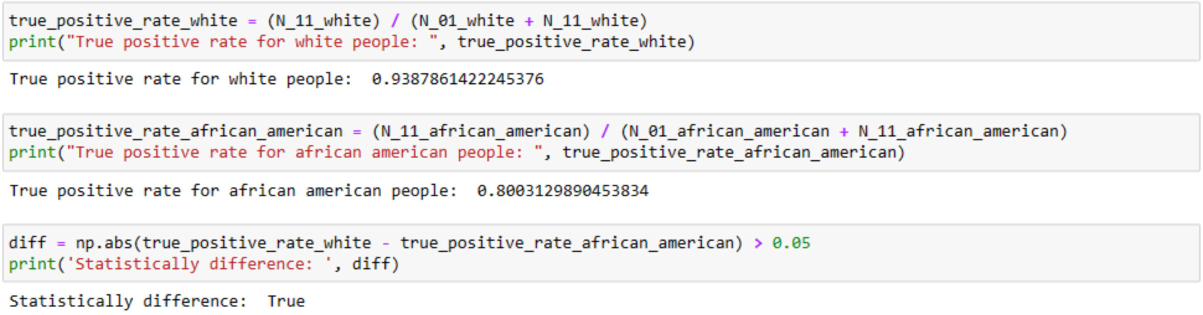
```
positive_rate_african_american = (N_10_african_american + N_11_african_american) / (N_00_african_american + N_01_african_amer
print("Positive rate for african american people: ", positive_rate_african_american)
```

```
Positive rate for african american people:  0.5105119372847131
```

```
diff = np.abs(positive_rate_white - positive_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

True Positive Rate (TPR)



```
true_positive_rate_white = (N_11_white) / (N_01_white + N_11_white)
print("True positive rate for white people: ", true_positive_rate_white)
```

```
True positive rate for white people:  0.9387861422245376
```

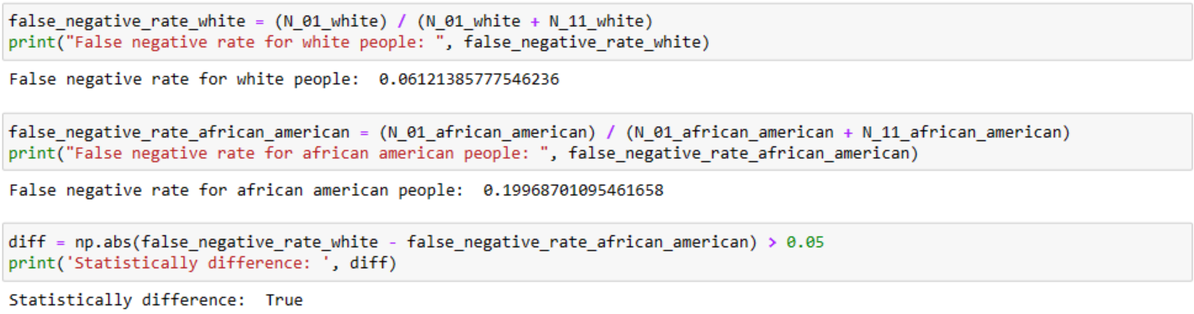
```
true_positive_rate_african_american = (N_11_african_american) / (N_01_african_american + N_11_african_american)
print("True positive rate for african american people: ", true_positive_rate_african_american)
```

```
True positive rate for african american people:  0.8003129890453834
```

```
diff = np.abs(true_positive_rate_white - true_positive_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

False Negative Rate (FNR)



```
false_negative_rate_white = (N_01_white) / (N_01_white + N_11_white)
print("False negative rate for white people: ", false_negative_rate_white)
```

```
False negative rate for white people:  0.06121385777546236
```

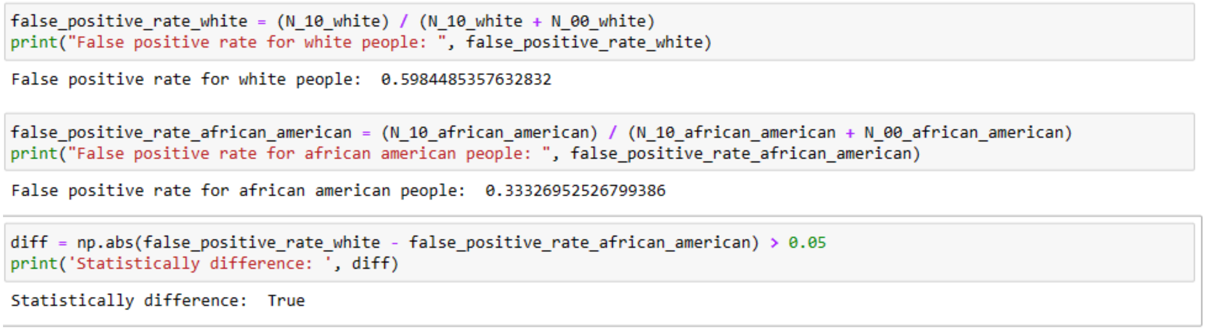
```
false_negative_rate_african_american = (N_01_african_american) / (N_01_african_american + N_11_african_american)
print("False negative rate for african american people: ", false_negative_rate_african_american)
```

```
False negative rate for african american people:  0.19968701095461658
```

```
diff = np.abs(false_negative_rate_white - false_negative_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

False Positive Rate (FPR)



```
false_positive_rate_white = (N_10_white) / (N_10_white + N_00_white)
print("False positive rate for white people: ", false_positive_rate_white)
```

```
False positive rate for white people:  0.5984485357632832
```

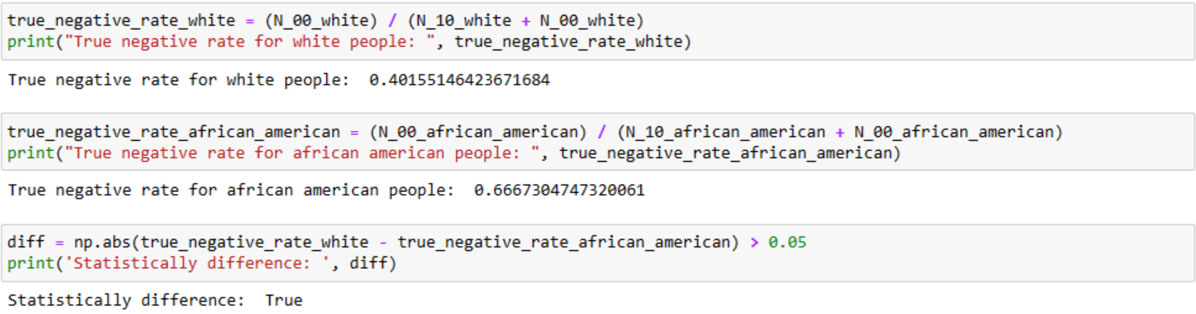
```
false_positive_rate_african_american = (N_10_african_american) / (N_10_african_american + N_00_african_american)
print("False positive rate for african american people: ", false_positive_rate_african_american)
```

```
False positive rate for african american people:  0.33326952526799386
```

```
diff = np.abs(false_positive_rate_white - false_positive_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

True Negative Rate (TNR)



```
true_negative_rate_white = (N_00_white) / (N_10_white + N_00_white)
print("True negative rate for white people: ", true_negative_rate_white)
```

```
True negative rate for white people:  0.40155146423671684
```

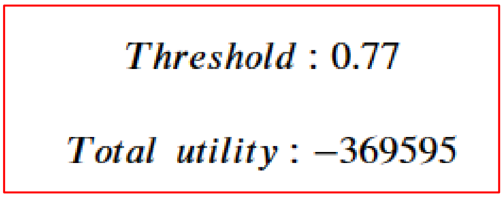
```
true_negative_rate_african_american = (N_00_african_american) / (N_10_african_american + N_00_african_american)
print("True negative rate for african american people: ", true_negative_rate_african_american)
```

```
True negative rate for african american people:  0.6667304747320061
```

```
diff = np.abs(true_negative_rate_white - true_negative_rate_african_american) > 0.05
print('Statistically difference: ', diff)
```

```
Statistically difference:  True
```

6.3 Utility of the decision maker



$$Threshold : 0.77$$

$$Total\ \ utility : -369595$$

This threshold confirms that the prediction model is less accurate.

To have a good utility the decision maker has to take less risks and therefore set a higher threshold.

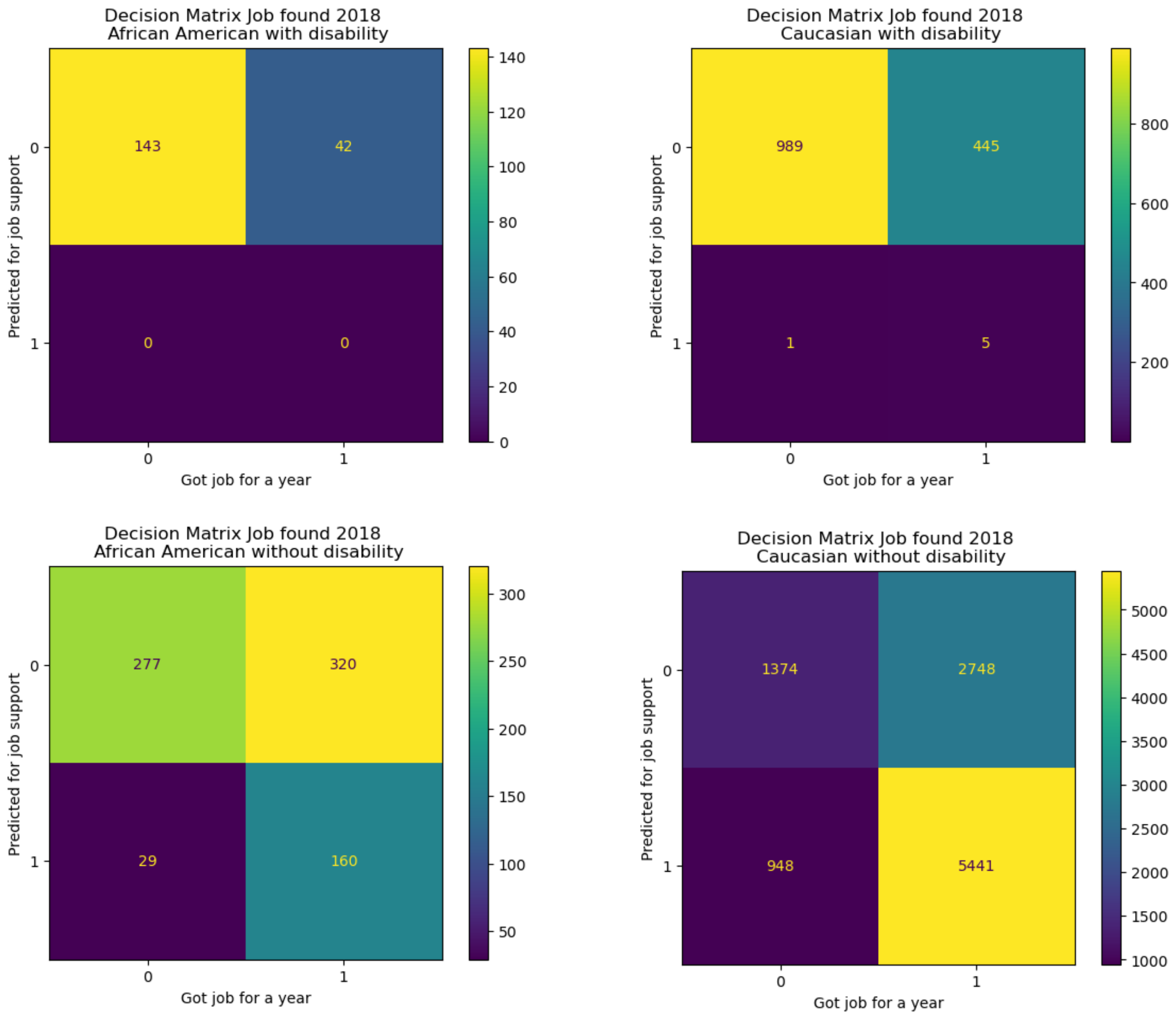
6.4 Moral analysis

False Omission Rate with J=1, A=Caucasian is: 0.31
False Omission Rate with J=1, A=African American is: 0.23
Statistical Significance (95%): True

Positive predictive value with J=1, A=Caucasian is: 0.83
Positive predictive value with J=1, A=African American is: nan
Statistical Significance (95%): False

False Omission Rate with J=2, A=Caucasian is: 0.67
False Omission Rate with J=2, A=African American is: 0.54
Statistical Significance (95%): True

Positive predictive value with J=2, A=Caucasian is: 0.85
Positive predictive value with J=2, A=African American is: 0.85
Statistical Significance (95%): False



Decision Matrix Job found 2018
African American with disability

Decision Matrix Job found 2018
Caucasian with disability

Decision Matrix Job found 2018
African American without disability

Decision Matrix Job found 2018
Caucasian without disability

Figure 25: Decision matrices by sensitive attribute and justifier

The more data we have the better we can see that the justifier is needed, because without it (see Figure 25) disabled people get almost no chance in reintegration into the job market.