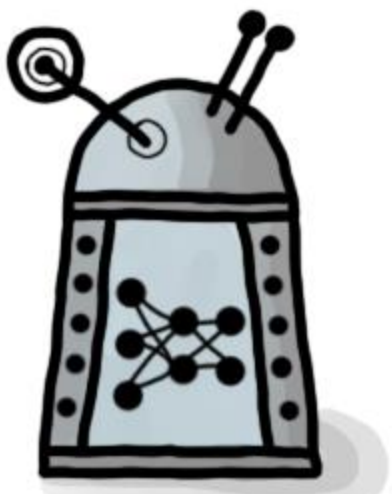
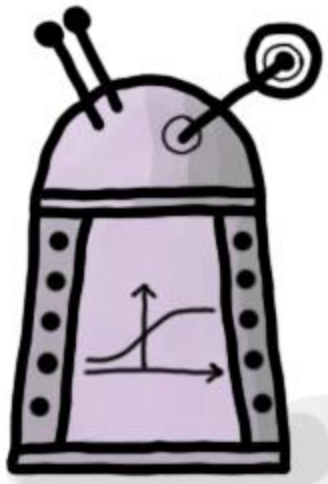
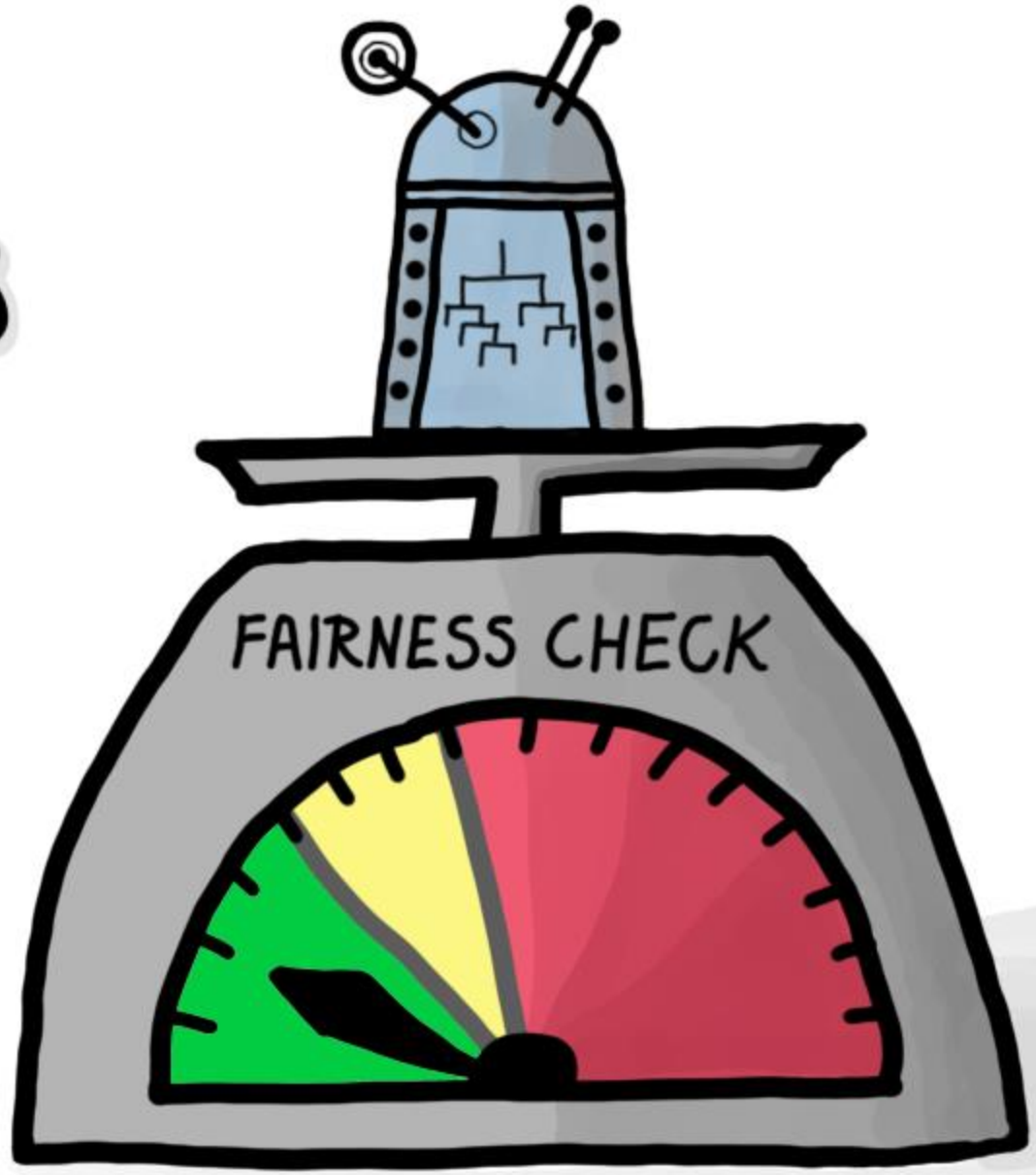
fairmodels







Plan prezentacji

- Czym jest fairness
- Po co fairness
- Jak mierzyć fairness, popularne metryki i ich odpowiedniki w macierzy błędów
- Wnioski z metryk, czyli jak porównywać modele pod kątem fairness
- Jak używać fairmodels
- fairness_check() – główna funkcja
- Wizualizacja fairness
- Mitygacja dyskryminacji modelu
- Tradeoff fairness a performance

Czym jest fairness?

Narzędzia i metody służące wykrywaniu i/lub zmniejszaniu dyskryminacji w regułach decyzyjnych mających wpływ na życie ludzkie

Idealnie alogrytm powinien podobnie traktować różne grupy społeczne, etniczne, płci, etc…

Jako że decyzje były wcześniej podejmowane przez człowieka/organizacje to model nauczony na tych danych może być stronniczy I uprzedzony w stosunku do niektórych grup osób

Po co fairness?

- Etyczność
- Konsekwencje prawne
- Zrozumnienie algorytmu
- Większe zaufanie

Jak mierzyć fairness?

Aby zmierzyć fairness potrzebujemy 3 rzeczy:

- Prawdziwej wartości przewidywanej zmiennej (y)
- Predykcji modelu ($\hat{y}$)
- Zmiennej chronionej/wrażliwej (A)

Potrzebujemy również kilku założeń:

- Zakładać będziemy że zmienna y jest binarna, gdzie 1 oznacza pozytywną, preferowaną decyzję a 0 decyzję przeciwną.
- Zmienna A również posiada dyskretne wartości, może być ich wiele, jednakże w rozważaniach równierz przyjmiemy 0 oraz 1 gdzie 1 oznacza grupę uprzywilejowaną.

Popularne metryki fairness

Statistical Parity

$$P(\hat{y}=1 \mid A=1) = P(\hat{y}=1 \mid A=0)$$

Equal opportunity

$$P(\hat{y}=1 \mid A=1, y=1) = P(\hat{y}=1 \mid A=0, y=1)$$

Equalized Odds

$$\forall_{y \in \{0,1\}} P(\hat{y}=1 \mid A=1, y) = P(\hat{y}=1 \mid A=0, y)$$

Macierz pomyłek

- Metryki na poprzednim slajdzie można otrzymać z macierzy pomyłek.
- Przykładowo Equal opportunity można wyliczyć dla macierzy pomyłek gdy $A = 1$ oraz gdy $A = 0$

$$P(\hat{y}=1 \mid A=1, y=1) \overset{?}{=} P(\hat{y}=1 \mid A=0, y=1)$$

$A = 1$

	P	N
P	16	8
N	10	21

$A = 0$

	P	N
P	8	20
N	31	34

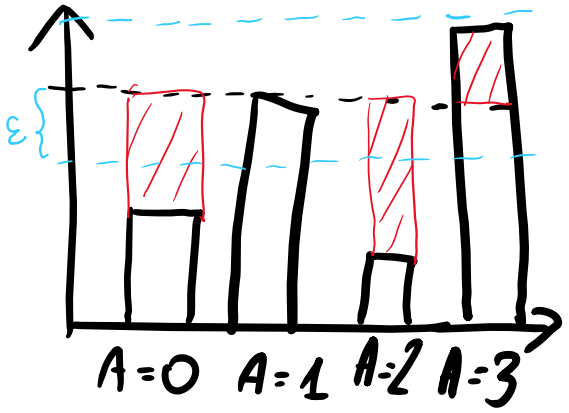
$$\Rightarrow \quad TPR_{A=1} = 0.61$$

$$TPR_{A=0} = 0.21$$

- Wyliczając najpopularniejsze metryki z macierzy pomyłek (np. TPR, TNR, PPV, F1, …) pokrywamy większość metryk fairness

Wnioski – parity loss

- Nie interesuje nas jaki jest poziom jest jakaś metryka, a jak bardzo te metryki się różnią. Jeżeli różnica nie przekroczy jakiejś wartości, to powiemy, że ta metryka jest spełniona.
- Informacja czy metryka dla konkretnej nieuprzywilejowanej grupy przekracza wartość dla grupy uprzywilejowanej jest cenna, ale łatwiej wizualizować sumę odległości euklidesowych między grupami – tym właśnie jest pairty loss



TPR criterion is satisfied when

$$\forall_{i=0,1,\ldots} \left| TPR_{A=i} - TPR_{A=1} \right| < \varepsilon$$

$$TPR_{parityloss} = \sum_{i} \left| TPR_{A=i} - TPR_{A=1} \right|$$

Nomenklatura

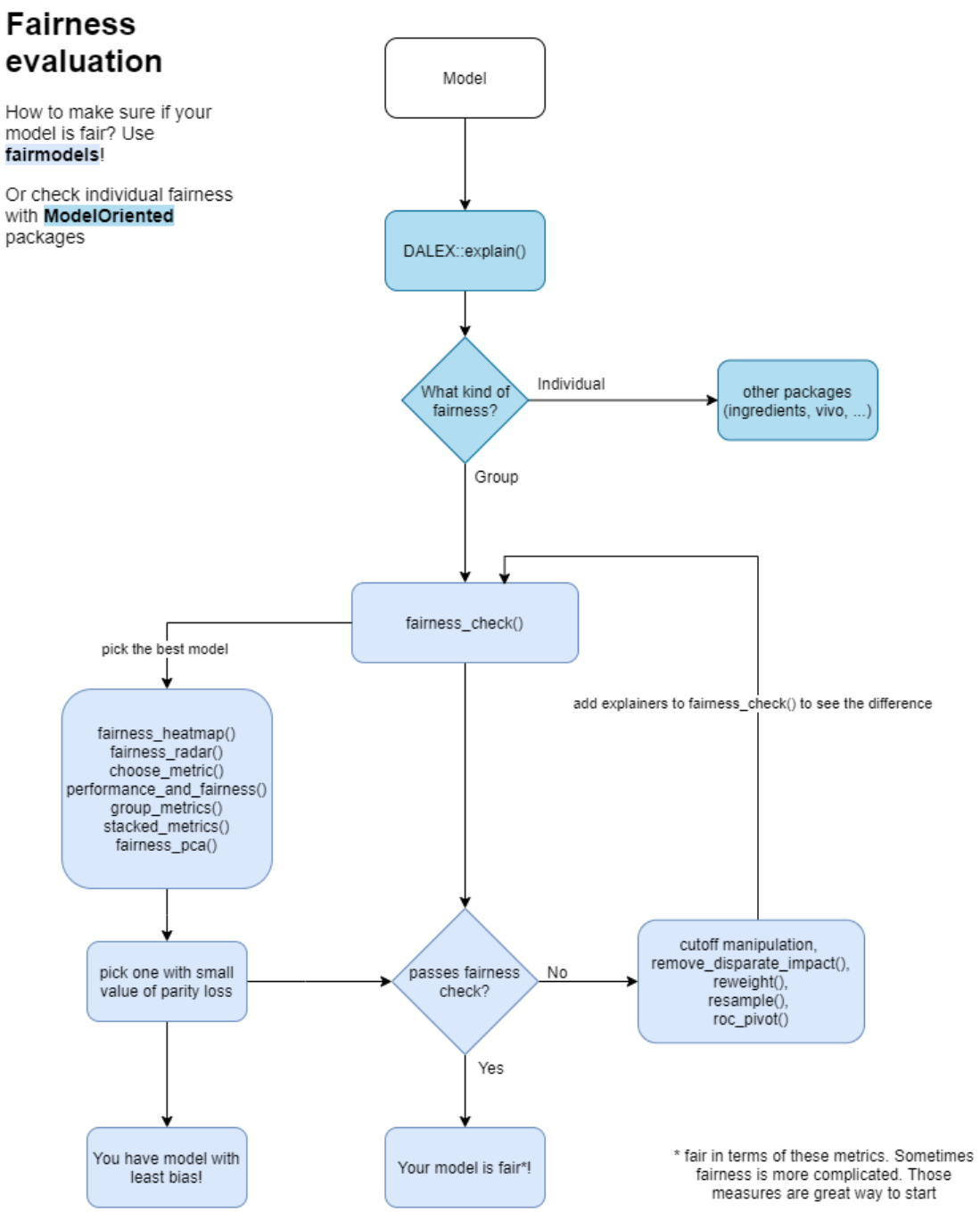
- Różnicę w metrykach dla grupy uprzywilejowanej i nieuprzylejowanej będziemy nazywać *difference*

$$\text{Statistical parity difference} := STP_{A=i} - STP_{A=1}$$

- Sumę modułów różnic będziemy nazywać *parity loss*

$$\text{Statistical parity parity loss} := \sum_i |STP_{A=i} - STP_{A=1}|$$

Jak używać fairmodels



fairmodels

- German credit data

```
> head(german)
  Risk    Sex Job Housing Saving.accounts Checking.account Credit.amount Duration              Purpose Age
1 good   male   2     own       not_known           little          1169        6             radio/TV  67
2  bad female   2     own          little         moderate          5951       48             radio/TV  22
3 good   male   1     own          little        not_known          2096       12            education  49
4 good   male   2    free          little           little          7882       42 furniture/equipment  45
5  bad   male   2    free          little           little          4870       24                  car  53
6 good   male   1    free       not_known        not_known          9055       36            education  35
```

- Tworzymy model i explainer

```
y_numeric <- as.numeric(german$Risk) -1

lm_model <- glm(Risk~.,
                data = german,
                family=binomial(link="logit"))


explainer_lm <- DALEX::explain(lm_model, data = german[,-1], y = y_numeric)
```

fairness_check() – główna funkcja

Input

```
fobject <- fairness_check(explainer_lm,
                          protected = german$Sex,
                          privileged = "male")
```

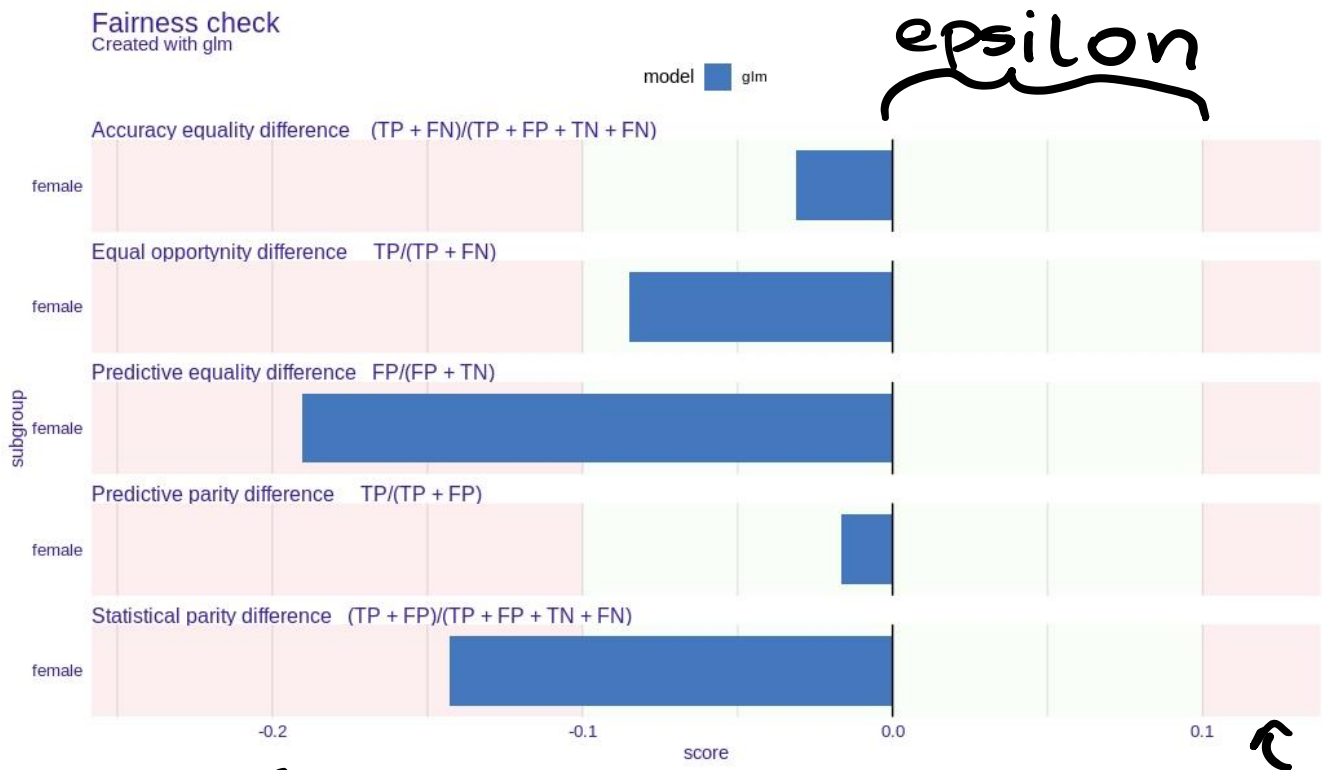
```
fobject <- fairness_check(explainer_lm,
                          protected = german$Sex,
                          privileged = "male",
                          cutoff = list(male = 0.5, female = 0.5),
                          label = 'glm',
                          epsilon = 0.1,
                          verbose = TRUE,
                          colorize = TRUE)
```

```
> fobject <- fairness_check(explainer_lm,
+                           protected = german$Sex,
+                           privileged = "male")
Creating fairness object
-> Privileged subgroup          : character ( Ok  )
-> Protected variable           : factor ( Ok  )
-> Cutoff values for explainers : 0.5 ( for all subgroups )
-> Fairness objects             : 0 object
-> Checking explainers          : 1 in total (  compatible  )
-> Metric calculation           : successful
 Fairness object created succesfully
```

fairness_check() – główna funkcja

print(fobject) and plot(fobject)

```
> fobject

Fairness check for models: glm

glm passes 3/5 metrics
Total loss:  0.4662677
```



epsilon

bias towards unprivileged

bias towards privileged

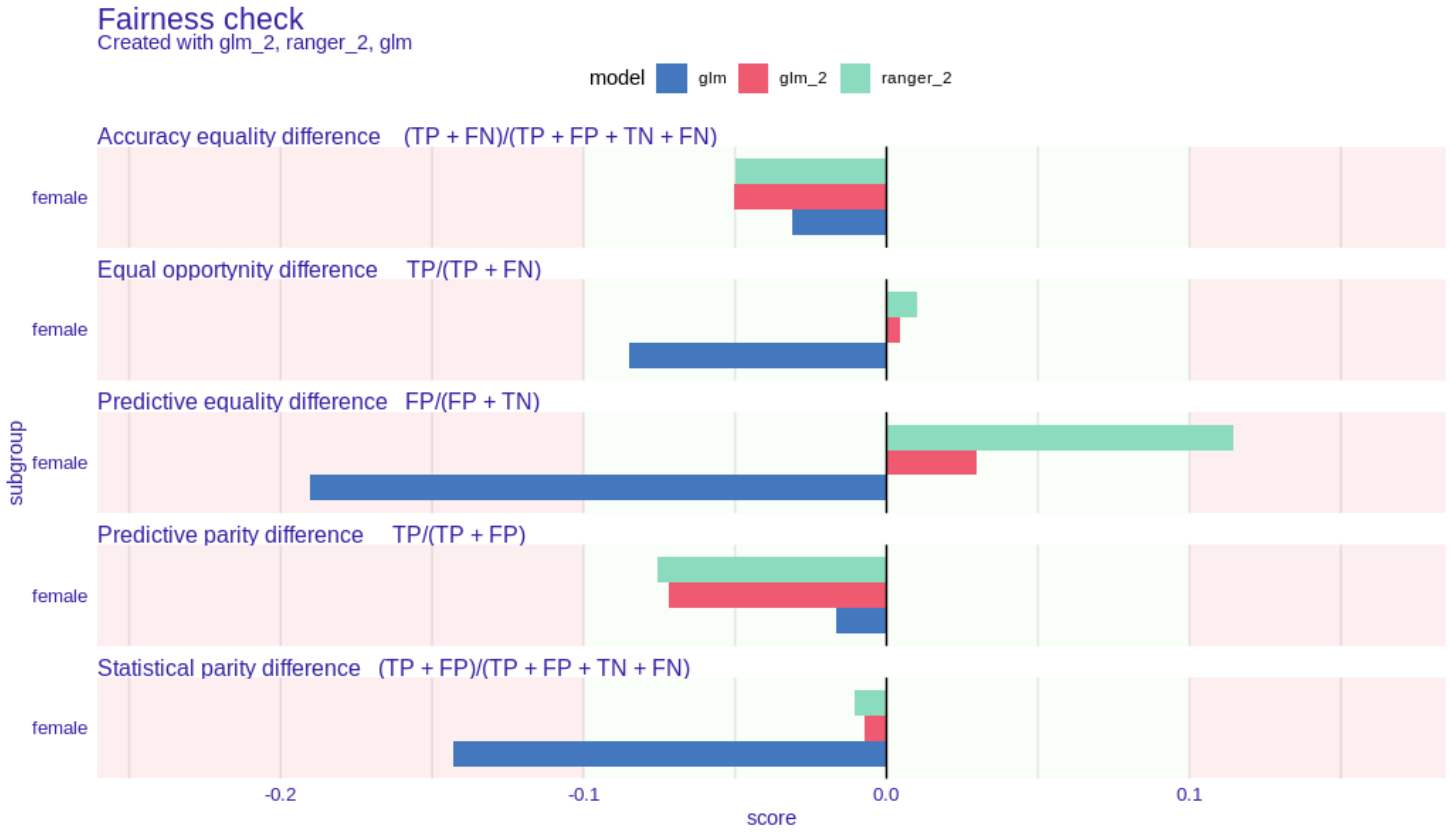
fairness object jest elastyczny

- Do fairness_check() pod warunkiem że modele przewidują tę samą zmienną możemy dodawać explainery oraz wcześniejsze fairness_object'y.
- Warunkiem jest, że każdy explainer ma mieć inną wartość parametru *label*, a fairness_object'y posiadają ten sam parameter *privileged* i *protected.*
- Parametry *cutoff* i *label* wpływają tylko na explainery, dane z fairness_object'ów są sklejane razem z przetworzonymi danymi z explainerów

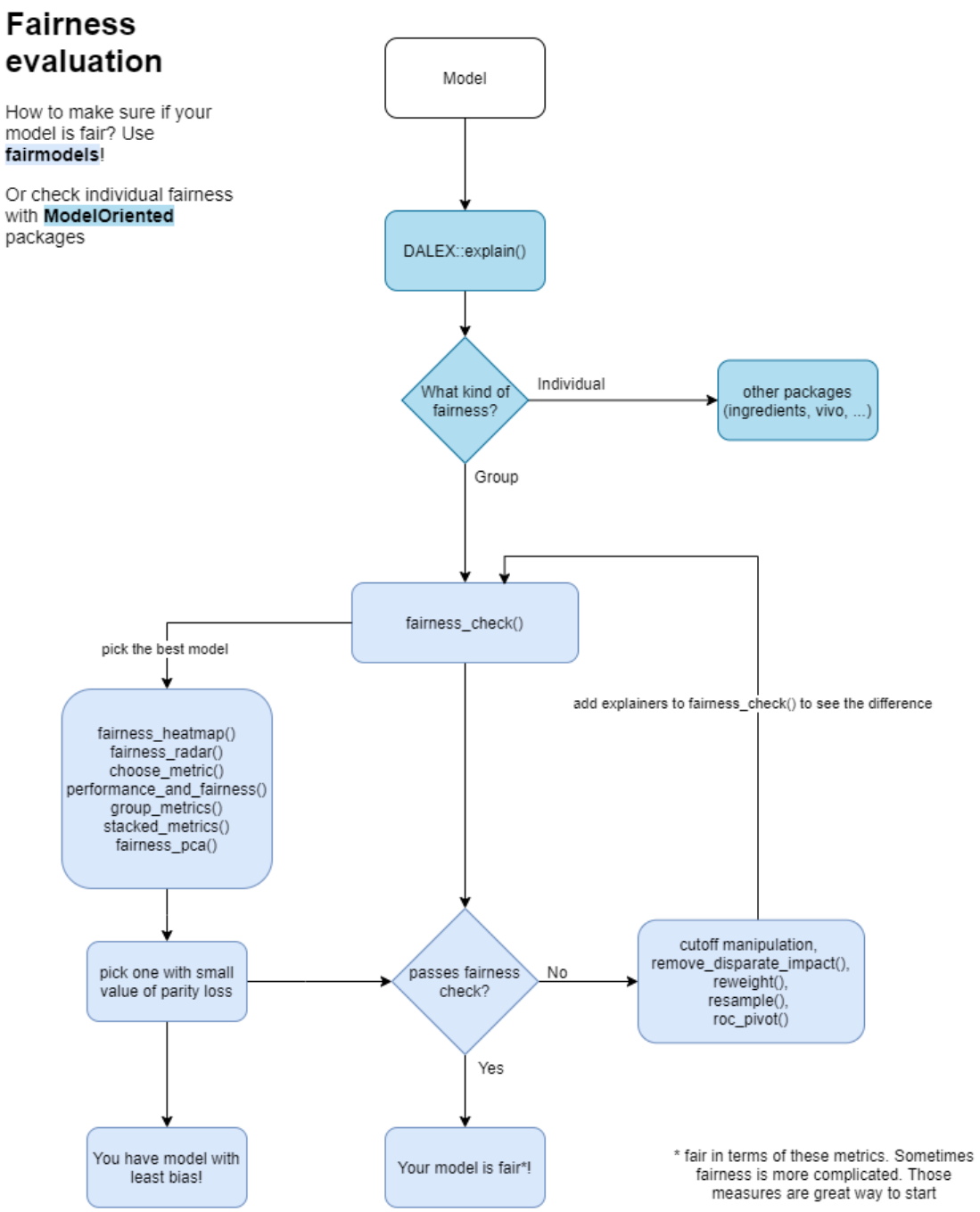
```
fobject <- fairness_check(explainer_lm, explainer_rf, fobject,
                          protected = german$Sex,
                          privileged = "male",
                          label = c("glm_2","ranger_2"),
                          cutoff = list(female = 0.4))

Creating fairness object
-> Privileged subgroup          : character ( ok  )
-> Protected variable           : factor ( ok  )
-> Cutoff values for explainers : female: 0.4, male: 0.5
-> Fairness objects             : 1 object (  compatible  )
-> Checking explainers          : 3 in total (  compatible  )
-> Metric calculation           : successful
 Fairness object created succesfully
```

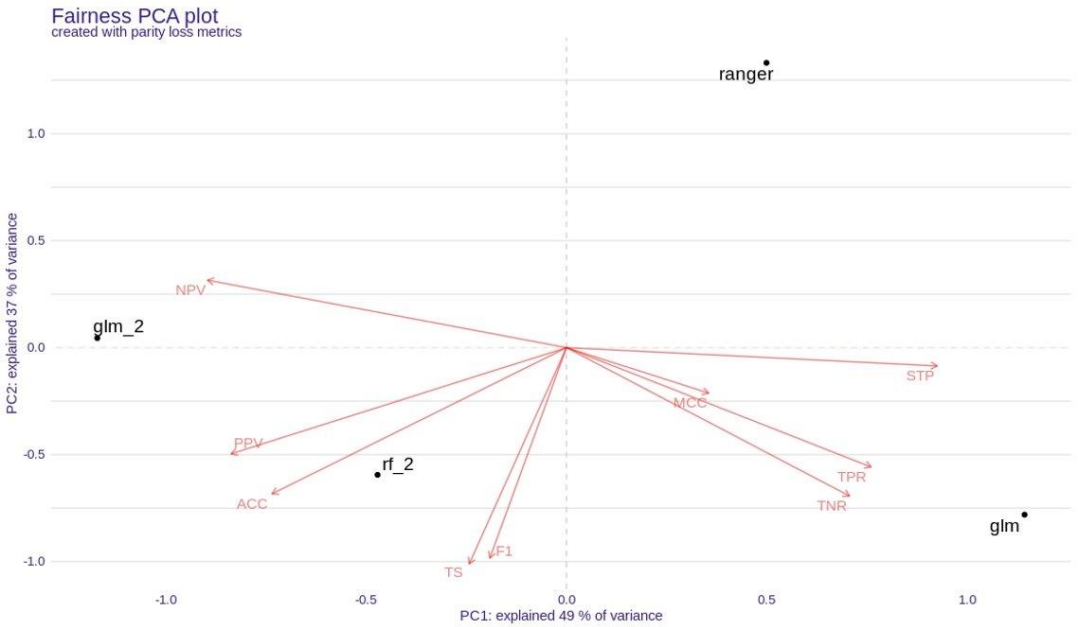
```
> fobject$cutoff
$glm_2
$glm_2$female
[1] 0.4

$glm_2$male
[1] 0.5
```

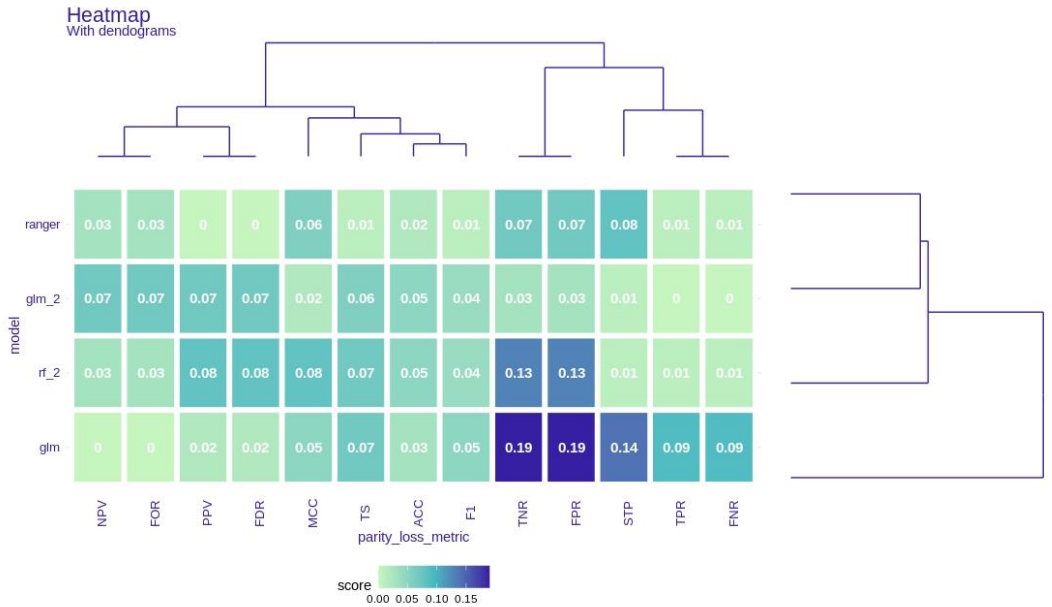
fairness object jest elastyczny



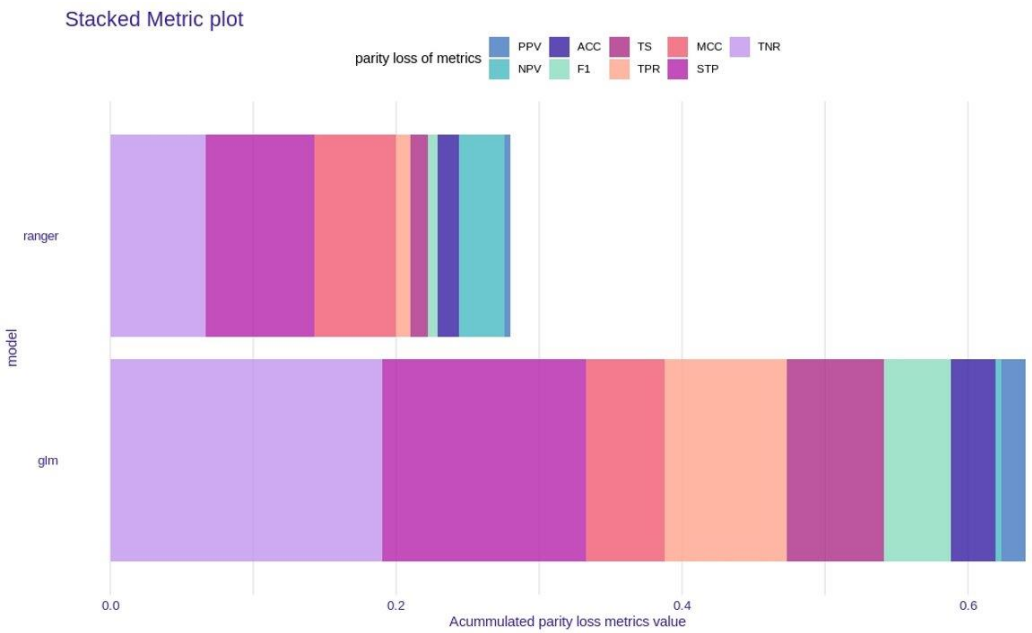
Jak używać fairmodels

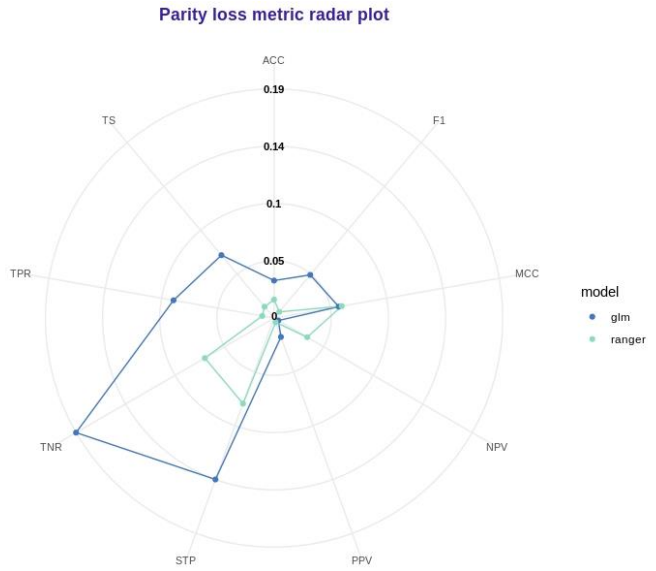




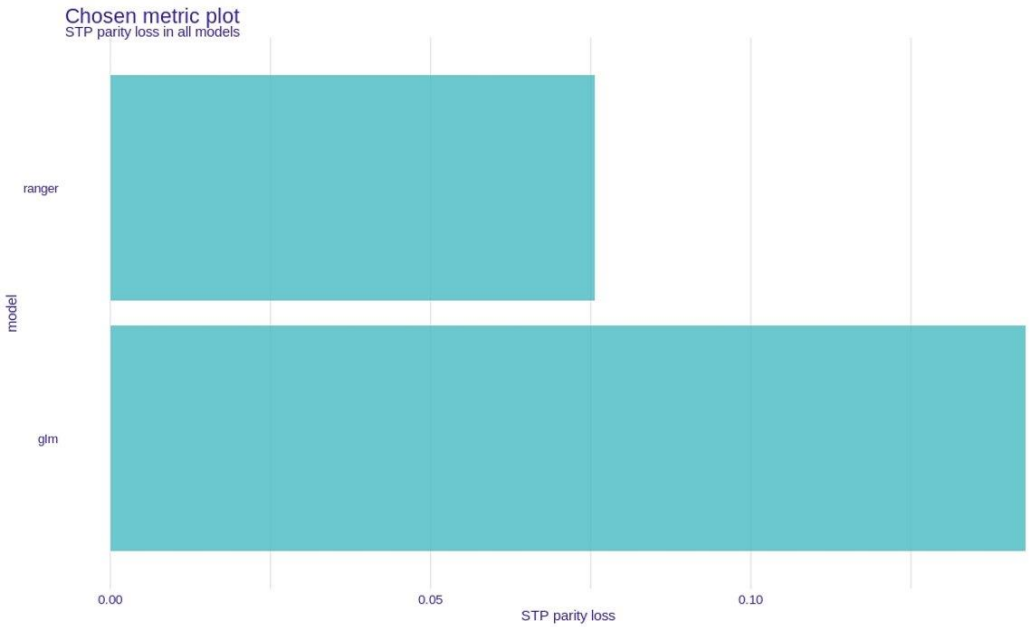


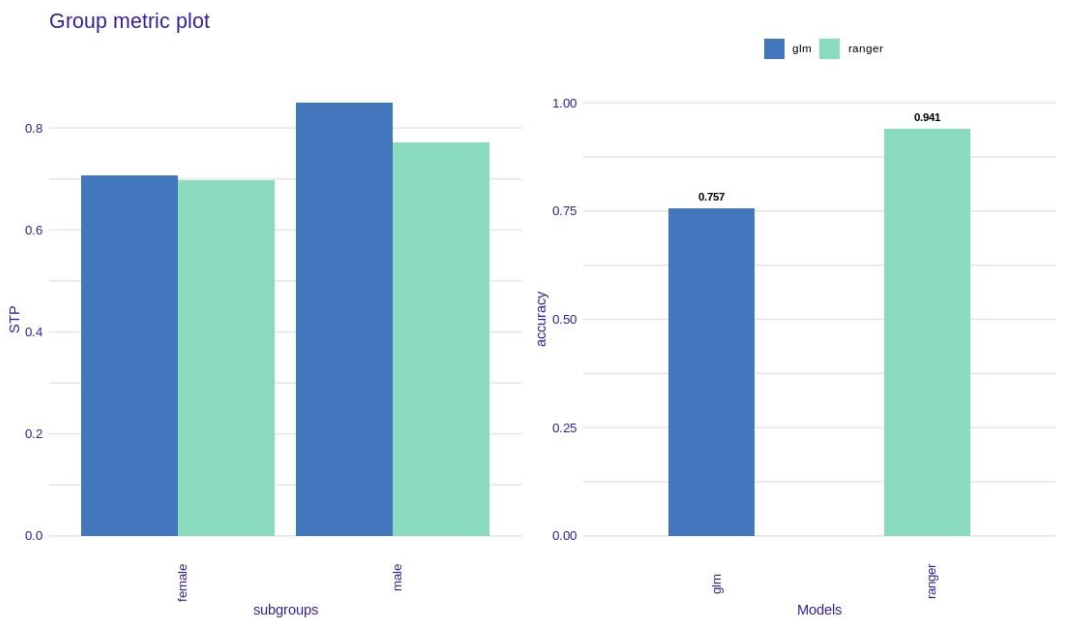
Wizualizacja fairness



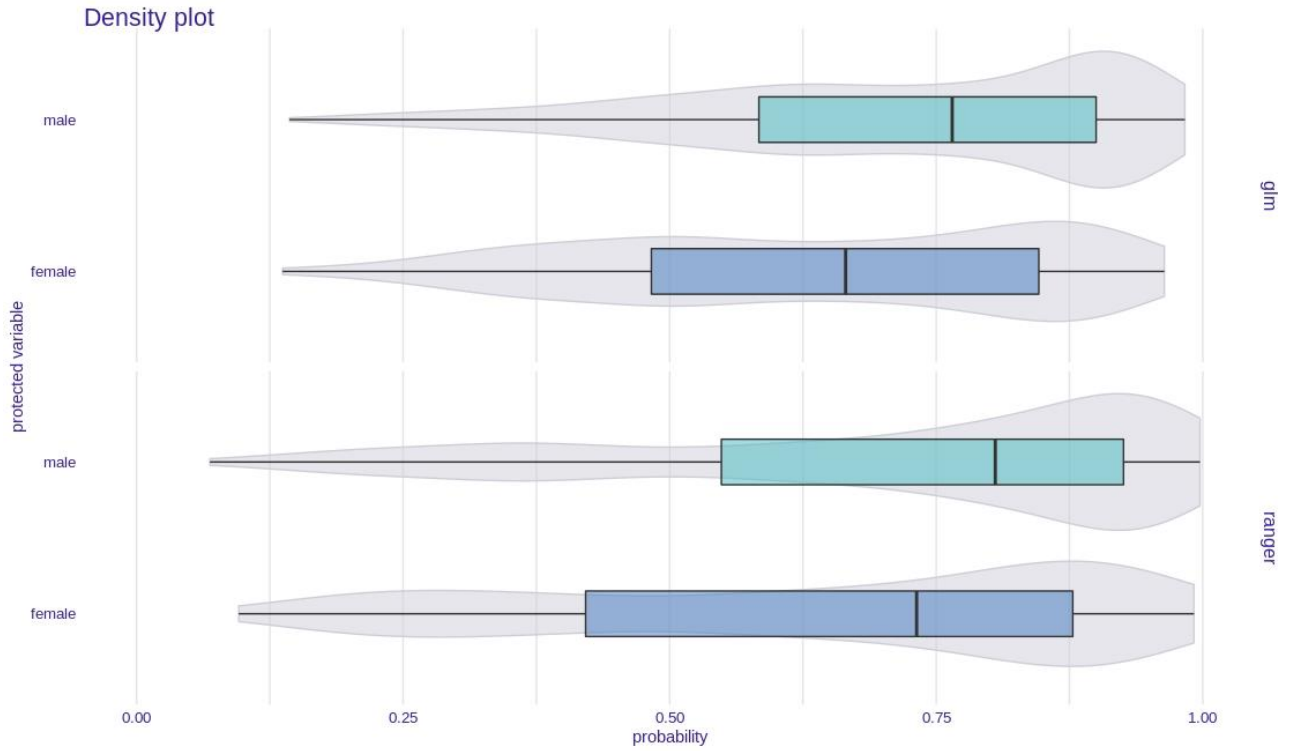


Wizualizacja fairness

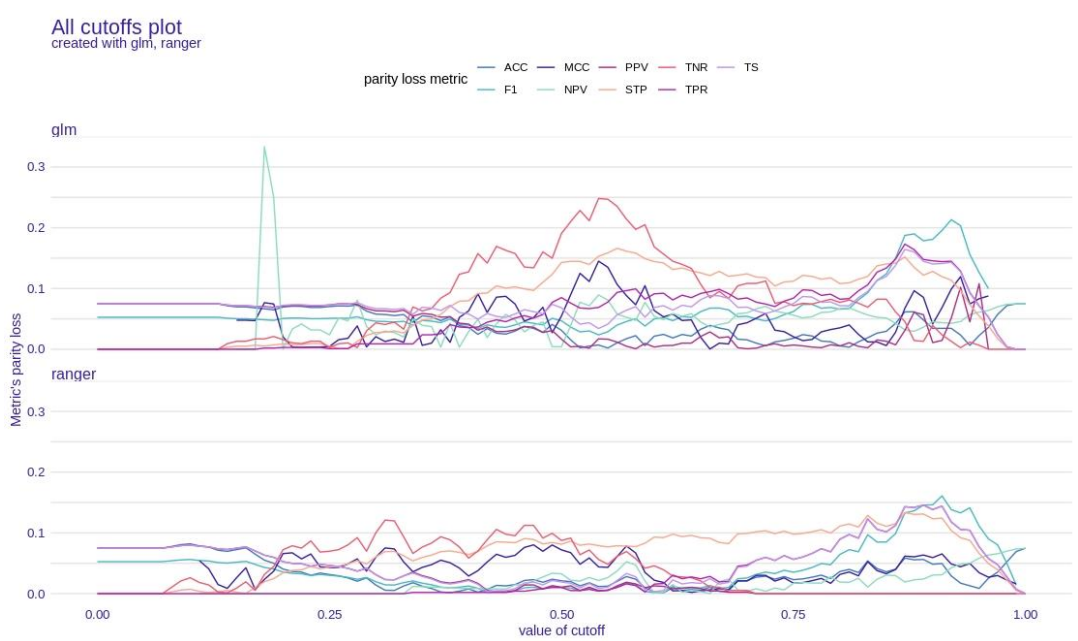


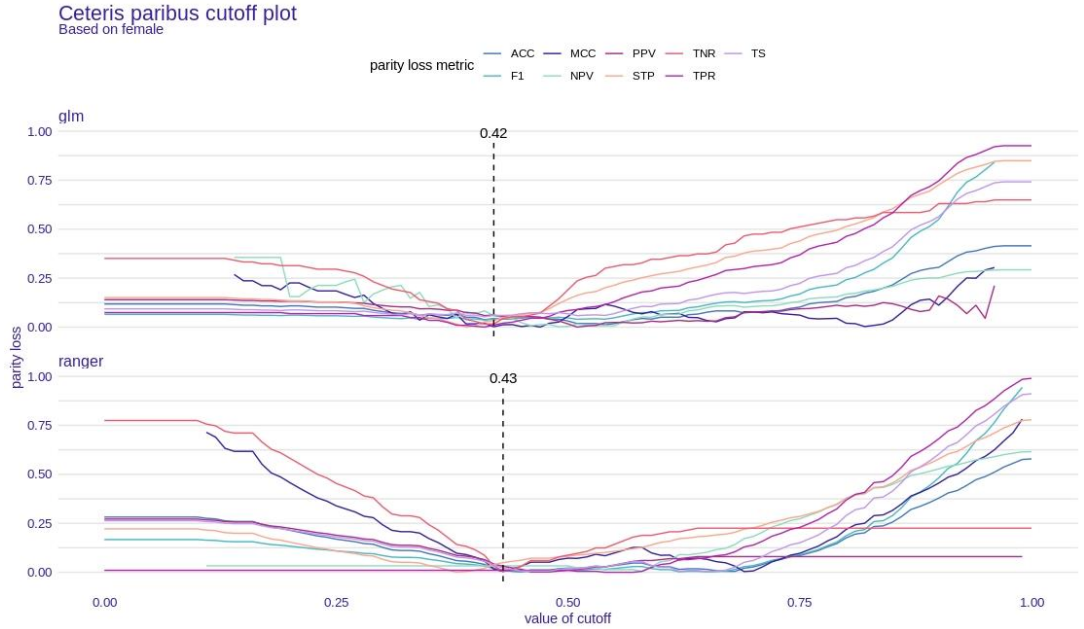


Wizualizacja fairness



Wizualizacja fairness

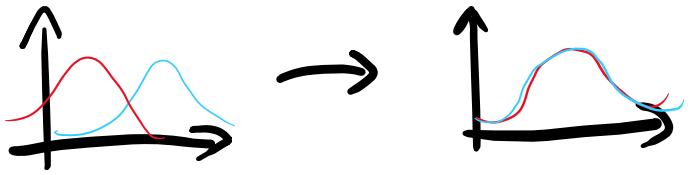




Wizualizacja fairness

Co zrobić gdy mój model nie spełnia fairness?

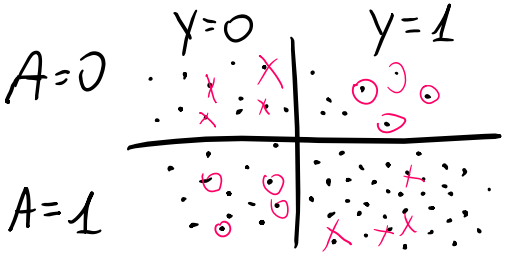
- disparate impact remover



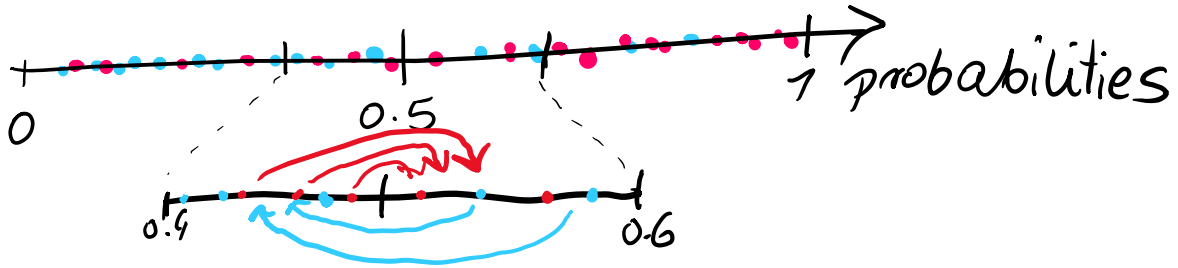
- reweighting

A	y	w
0	0	0.43
0	1	1.56
1	0	2.24
1	1	0.81

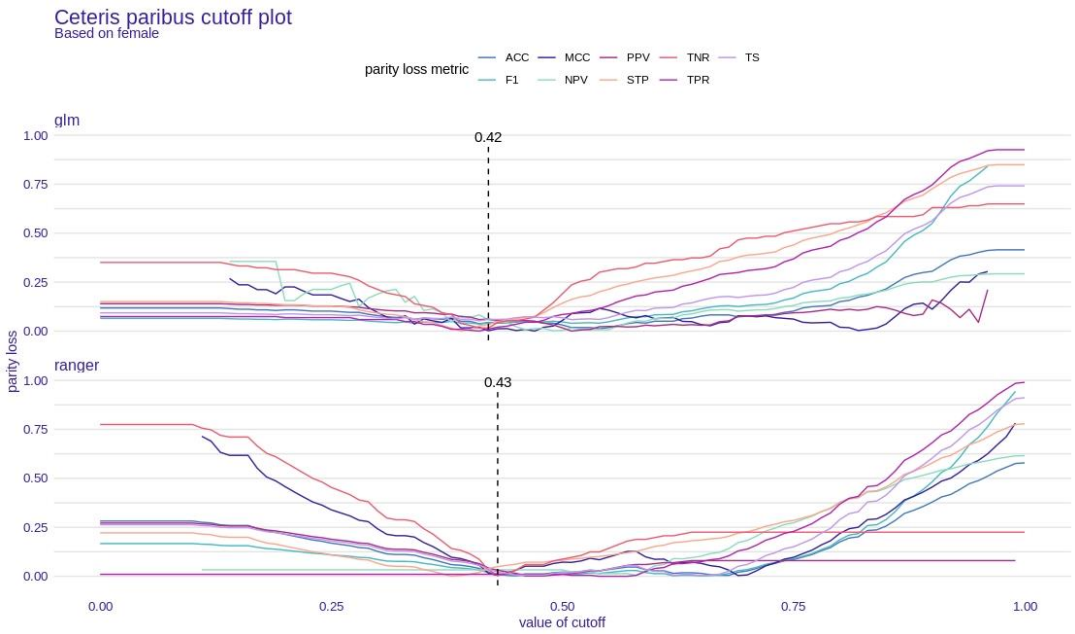
- resampling

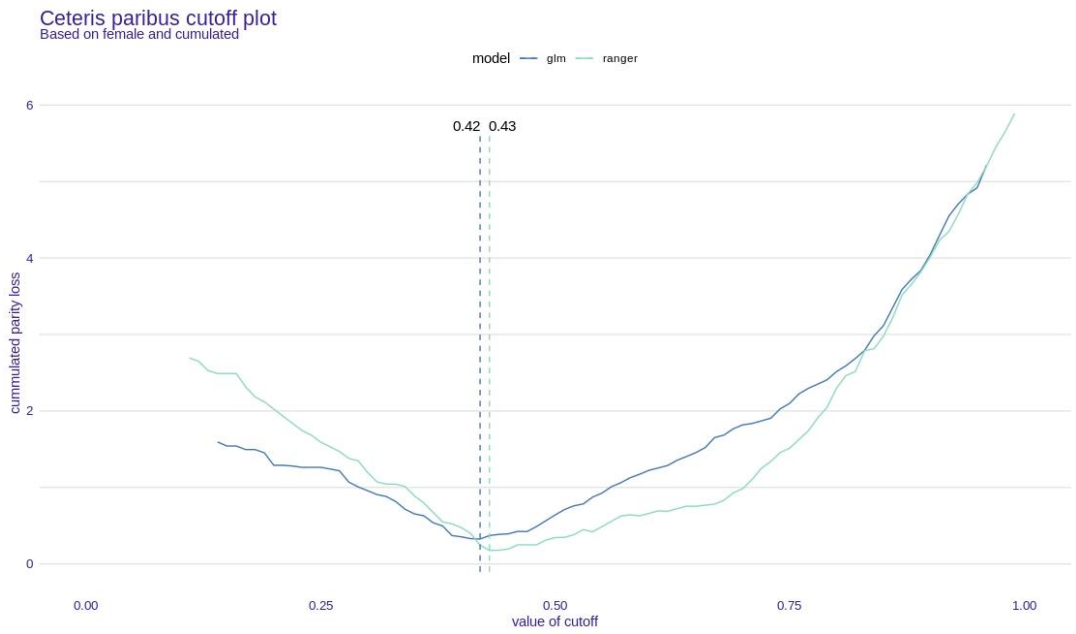


- Reject Option based Classification (pivot)

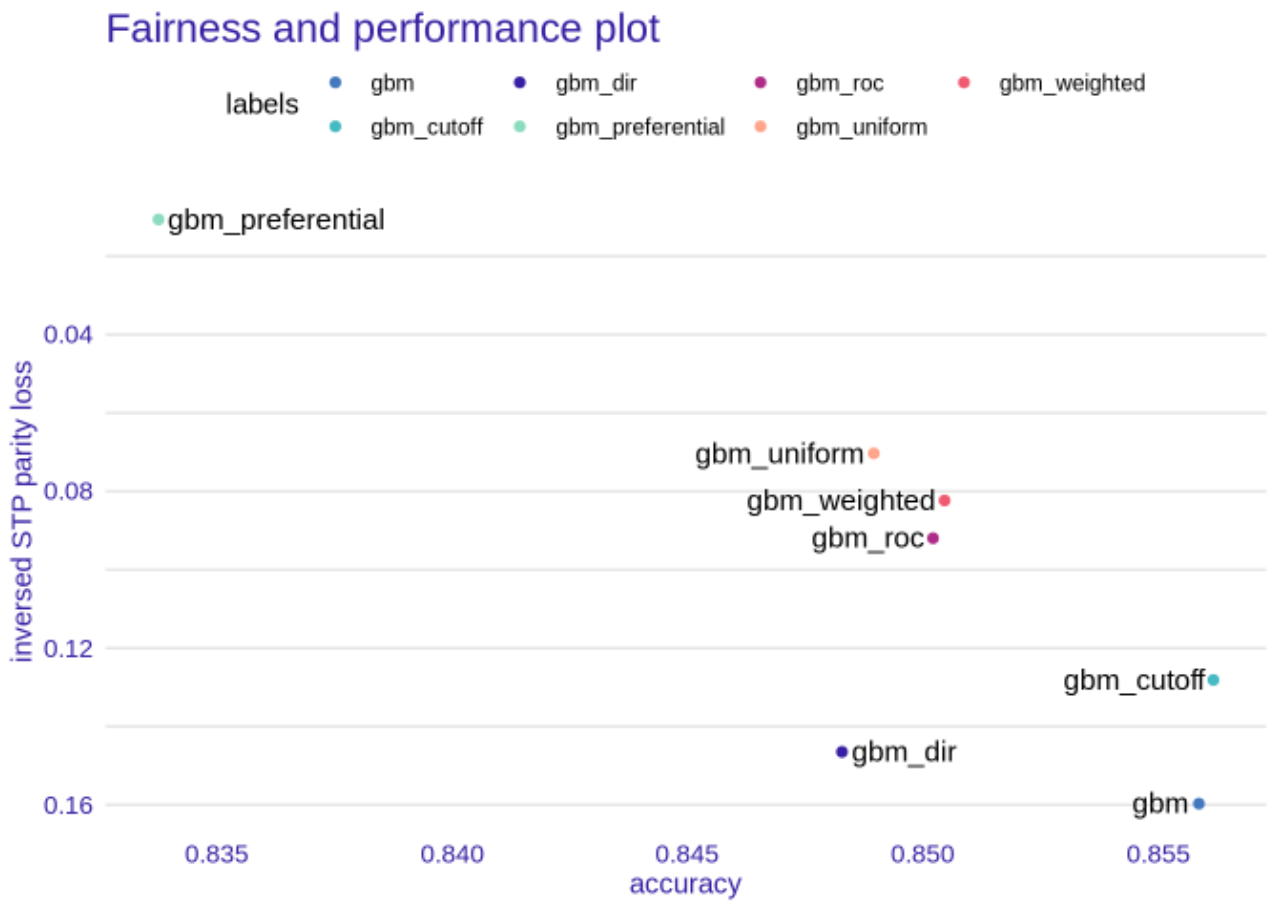


- manipulacja cutoffem





Manipulacja cutoffem



Tradeoff między fairness a performance

Q&A