

DAV-Herbsttagung 2023, 21.11. Maritim Hannover Flughafen

Simpson's Paradoxon oder: Die Tücken des diskriminierungsfreien Pricings



Bildquelle: Pixabay

Prof. Dr. Fabian Transchel

Zur Person

An der HS Harz seit 2020

- Stiftungsprofessur „Data Science“ der E+S Rückversicherung AG
- Studiengangskoordinator M.Sc. Data Science (alle Varianten)

Berufspraxis: Senior Data Scientist in der Versicherungsbranche

- Kfz-Telematik / Fahrassistenzsysteme / Autonomes Fahren

Forschungsschwerpunkte

- Actuarial Data Science
- Kfz-Versicherung / Mobility
- Erklärbarkeit / Regulatorik von KI-Modellen

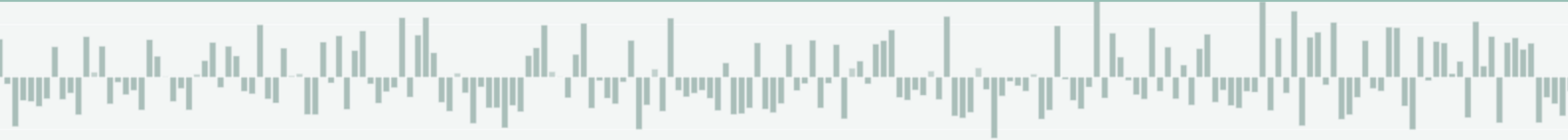
Bildquelle: Hochschule Harz



Simpson's Paradoxon oder: Die Tücken des diskriminierungsfreien Pricings - Agenda

- Einleitung: Diskriminierungsfragen im Pricing
- Rekurs: Das Simpson-Paradoxon
- Use Case: Kfz-Pricing
- Kennzahlen zur Evaluation von Diskriminierung (bzw. Disparate Impact)
- Fazit und Ausblick

Diskriminierungsfragen im Pricing



Diskriminierungsfragen im Pricing

Diskriminierung i.e.S.

§1(1) AGG (Allgemeines Gleichbehandlungsgesetz):

„Ziel des Gesetzes ist, Benachteiligungen aus Gründen der Rasse oder wegen der ethnischen Herkunft, des Geschlechts, der Religion oder Weltanschauung, einer Behinderung, des Alters oder der sexuellen Identität zu verhindern oder zu beseitigen.“

Benachteiligungen sind untersagt, soweit sie an eines der folgenden personenbezogenen Merkmale anknüpfen (§2 AGG):

- „Rasse“ oder ethnische Herkunft
- Geschlecht
- Religion und Weltanschauung
- Behinderung
- Alter (jedes Lebensalter)
- Sexuelle Identität

Diskriminierungsfragen im Pricing

Prämissen des Vortrags

Dass AGG steht im Ausgleichsverhältnis mit dem VAG:

„Eine Ungleichbehandlung bedarf also eines sachlichen Grundes. Liegt ein solcher vor, so ist nach allgemeiner Auslegung der Gesetzesnorm eine Ungleichbehandlung ausdrücklich erlaubt. Gleichbehandlung heißt also, gleiche Risiken gleich zu behandeln, aber auch unterschiedliche Risiken unterschiedlich.“[1]

- Trotzdem gibt es immer wieder Diskussionen über die Entfernung (Nichtverwendung) von geschützten Attributen und eine weitergehende Behandlung von Indirekter Diskriminierung sowie dem sog. Disparate Impact.
- Das Thema kam im CADS-Curriculum bisher zu kurz.
- AI Act könnte VAG ggf. „überstimmen“.

[1] https://aktuar.de/fachartikelaktuaraktuell/2020_AA49_Diskriminierung.pdf

Diskriminierungsfragen im Pricing

Glossar

- Was ist **Diskriminierung**?
 - Eine „Ungleichbehandlung von Versicherungsnehmern ohne sachlichen Grund“.
- Was ist ein **geschütztes Attribut**?
 - *Ethnische Herkunft, Geschlecht, Religion, Behinderung, Alter, Sexueller Identität*
- Was ist **indirekte** Diskriminierung?
 - Ungleichbehandlung von Versicherungsnehmern auf Basis von Merkmalen, die mit geschützten Attributen in Beziehung (d.h. i.d.R. mehr oder weniger starke Korrelation)
- Was ist „**Disparate Impact**“?
 - Ursprung in US-Gesetzgebung: „**Disparate impact** in the law of the United States refers to practices in employment, housing, and other areas that adversely affect one group of people of a protected characteristic more than another, even though rules applied by employers or landlords are formally neutral.”[1]
 - Moderne Lesart bzgl. ML: **Disparate Impact beschreibt adverse Effekte für Personen mit vermeintlich nachteiligen geschützten Attribute, die erst durch die Anwendung von Antidiskriminierungsmaßnahmen Nachteile erleiden**[2].

[1] https://en.wikipedia.org/wiki/Disparate_impact

[2] z.B. H. Wang, B. Ustun and F. P. Calmon, "On the Direction of Discrimination: An Information-Theoretic Analysis of Disparate Impact in Machine Learning," 2018 ISIT, Vail, CO, USA, 2018, pp. 126-130, doi: 10.1109/ISIT.2018.8437661.

Diskriminierungsfragen im Pricing

Lösungsansätze

Disparate Impact and Unfairly Discriminatory Insurance Rates

DISCRIMINATION-FREE INSURANCE PRICING

BY

SIMPSON'S PARADOX, CONFOUNDING VARIABLES,
AND INSURANCE RATEMAKING

Neue Lösungsansätze gibt es von mir heute keine*.

Dafür aber:

Weitere Fragen, die m.E. bisher *unzureichend behandelt* sind und zu deren Betrachtung Aktuare in ihrer besonderen Perspektive und Verantwortung *beitragen können und sollten*.

* Bis auf den Vorschlag von Kennzahlen zur Bemessung von Disparate Impact im Pricing.

[1] Miller, M. J. (2009) Disparate Impact and Unfairly Discriminatory Insurance Rates, The 2009 CAS Ratemaking Call Papers, Las Vegas, Nevada

[2] Lindholm, M., Richman, R., Tsanakas, A., & Wüthrich, M. (2022). DISCRIMINATION-FREE INSURANCE PRICING. ASTIN Bulletin, 52(1), 55-89. doi:10.1017/asb.2021.23

[3] JOHN A. STENMARK AND CHENG-SHENG PETER WU, SIMPSON'S PARADOX, CONFOUNDING VARIABLES, AND INSURANCE RATEMAKING, Textbook

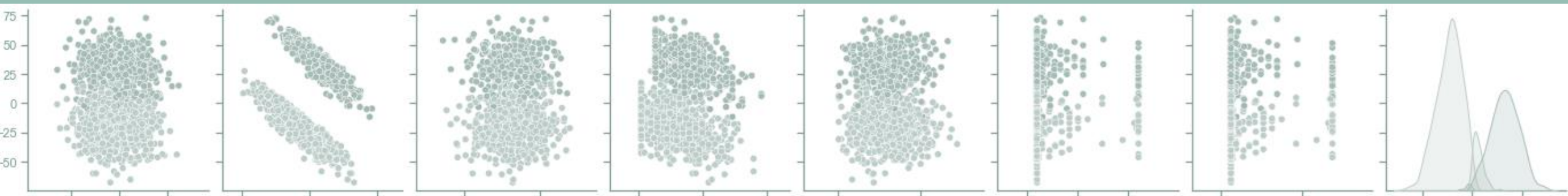
[4] Schwartz et al.: Towards a Standard for Identifying and Managing Bias in Artificial Intelligence (2022), NIST Special Publication 1270, <https://doi.org/10.6028/NIST.SP.1270>

Diskriminierungsfragen im Pricing

Disclaimer

1. Ich spreche mich ausdrücklich gegen jede Form der sachgrundfreien Diskriminierung aus, insbesondere wenn diese auf Grund von geschützten Attributen erfolgt.
2. Der weitere Vortrag hat Use Case-Charakter.
 - a) Es stehen Abschlussberichte verschiedener DAV-Arbeitsgruppen aus, deren Ergebnisse weder vorweggenommen werden sollen noch davon beeinflusst sind.
 - b) Die beschriebenen Zusammenhänge dienen der Illustration der technischen Aspekte des Problemfelds „Diskriminierung“, ausdrücklich **nicht** der politischen Dimension der Implikationen.

Rekurs: Das Simpson-Paradoxon



Diskriminierungsfragen im Pricing

Reality Check

- Welche der beiden folgenden Aussagen ist wahr?

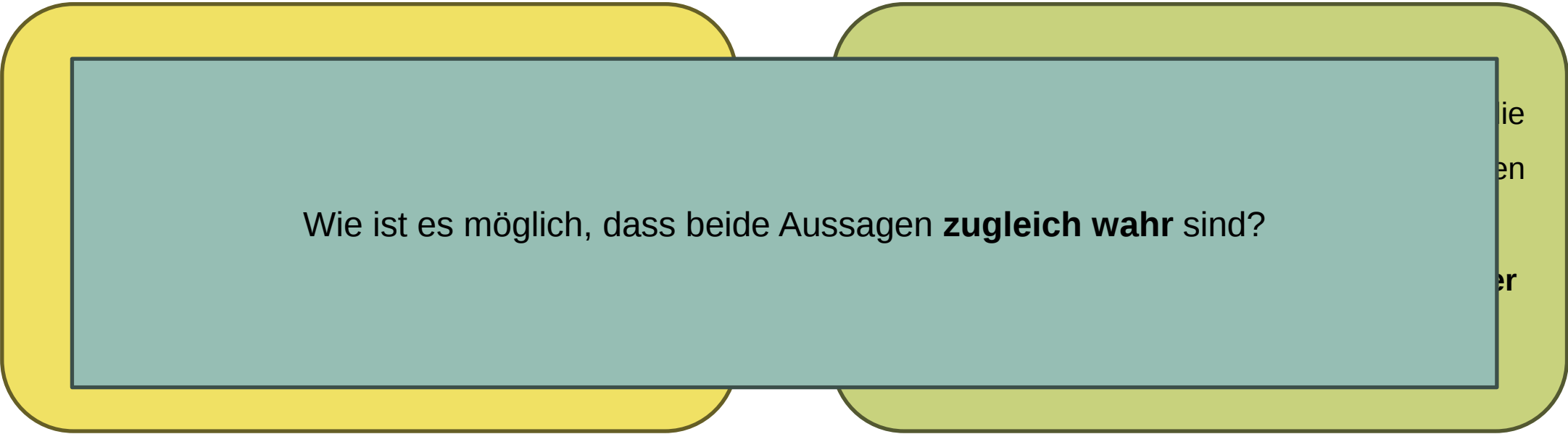
Frauen sind „bessere Autofahrer“ und zahlen in einem durchschnittlichen Portfolio **im Schnitt ~10-15% geringere Beiträge.**

In einem durchschnittlichen Portfolio beträgt die **Wahrscheinlichkeit ~80%**, dass bei beliebigen Vertragspaaren mit den Ausprägungen **männlich <-> weiblich gerade der Mann der „bessere Autofahrer“ ist.**

Diskriminierungsfragen im Pricing

Reality Check

- Welche der beiden folgenden Aussagen ist wahr?



Wie ist es möglich, dass beide Aussagen **zugleich wahr** sind?

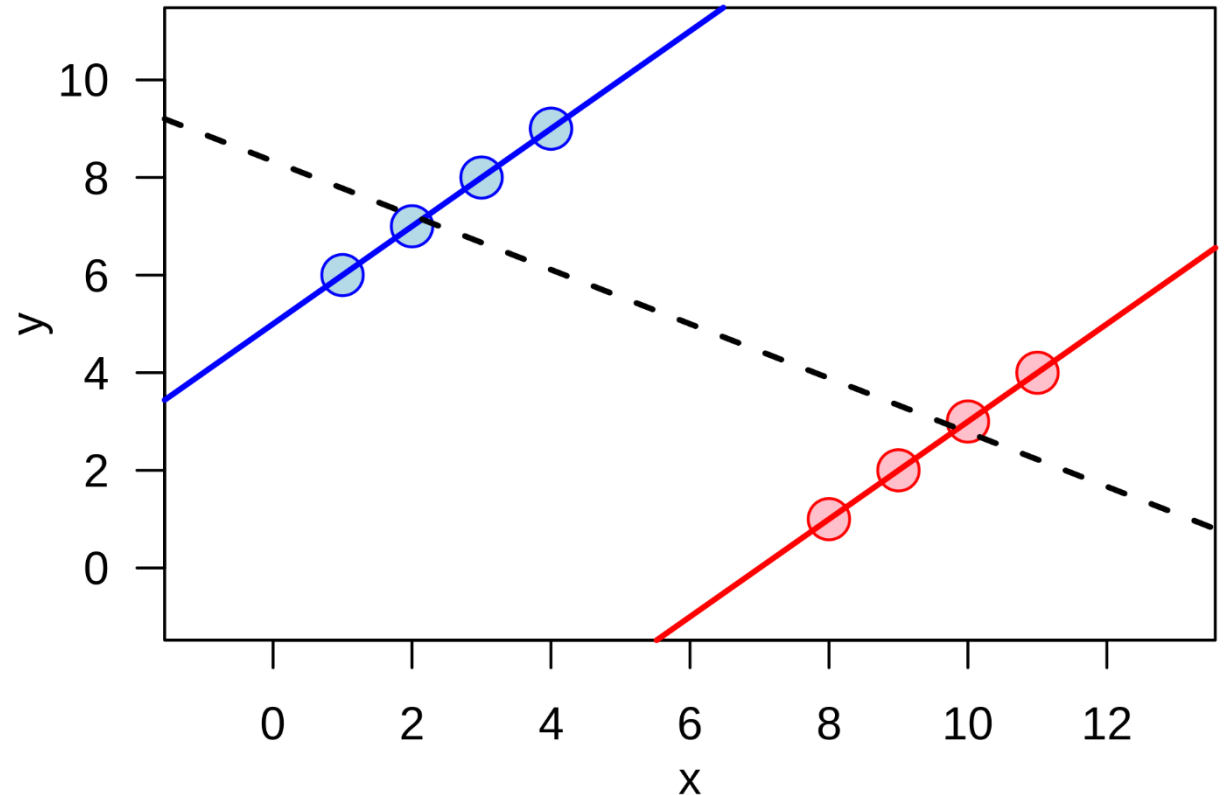
Das Simpson-Paradoxon

Die Mutter aller Missverständnisse

Paradoxon von Simpson^[1]

Bei der Analyse von kategorialen Variablen scheint es manchmal so, als ob die Bewertung verschiedener Gruppen unterschiedlich ausfällt, je nachdem ob man die Ergebnisse der Gruppen kombiniert oder nicht.

- **Störvariablen** wurden in der statistischen Analyse nicht betrachtet.
- Durch die Nichtbeachtung der Gruppen kommt es zu einer **Scheinkorrelation**.
- Analoges Phänomen kontinuierlich: Lord's Paradox



[1] Edward Hugh Simpson: *The Interpretation of Interaction in Contingency Tables*. In: *Journal of the Royal Statistical Society, Ser. B*. Band 13, 1951, S. 238–241, doi:[10.1111/j.2517-6161.1951.tb00088.x](https://doi.org/10.1111/j.2517-6161.1951.tb00088.x), JSTOR:2984065.
Bildquelle: https://de.wikipedia.org/wiki/Simpson-Paradoxon#/media/Datei:Simpson's_paradox_continuous.svg, CC-BY 4.0, <https://commons.wikimedia.org/wiki/User:Schutz>

Das Simpson-Paradoxon

Die Mutter aller Missverständnisse

Paradoxen von Simpson^[1]

Bei der
ob die B
nachde

→ Stö

→ Dur
eine

→ Ana

Warum ist das relevant im Kontext von **Diskriminierung im Pricing?**

→ Die Festlegung des „richtigen“ Vergleichsmaßstab ist kritisch und zugleich nicht kanonisch. ←

[1] Edward Hugh Simpson: *The Interpretation of Interaction in Contingency Tables*. In: *Journal of the Royal Statistical Society, Ser. B.* Band 13, 1951, S. 238–241, doi:[10.1111/j.2517-6161.1951.tb00088.x](https://doi.org/10.1111/j.2517-6161.1951.tb00088.x), JSTOR:2984065.
Bildquelle: https://de.wikipedia.org/wiki/Simpson-Paradoxon#/media/Datei:Simpson's_paradox_continuous.svg, CC-BY 4.0, <https://commons.wikimedia.org/wiki/User:Schutz>

Use Case: Kfz-Pricing



Bildquelle: Pixabay

Use Case

Ein diskriminierungsfreundlicher(?) Datensatz

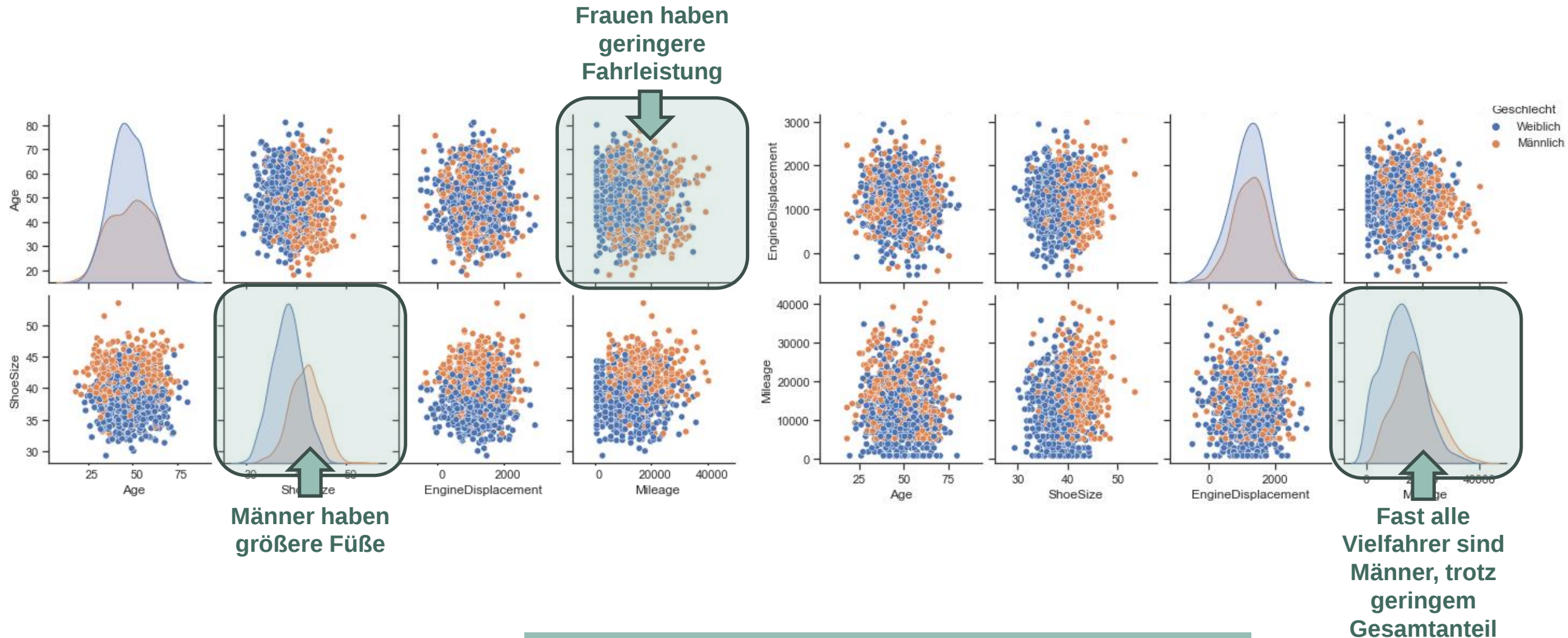
- Zielsetzung des Use Case: Illustration von Simpson-Paradoxon und Disparate Impact
- Ansatz: Kfz-Pricing-Tabelle mit typischen und untypischen Attributen

	Gender	Age	ShoeSize	EngineDisplacement	Mileage	VehicleValue	freq	Response	AgeGroup
0	1	45.5	38.840481	1772.958706	7222.0	4418.959431	0.05500	550.0	[46 - 65]
1	0	51.5	36.346661	1383.202597	15778.0	14412.902857	0.04875	487.5	[46 - 65]
2	0	58.0	39.226578	406.564692	7778.0	8423.109328	0.03625	362.5	[56 - 65]
3	0	56.0	37.788013	1141.595642	19778.0	14116.257354	0.04875	487.5	[56 - 65]
4	0	42.0	31.635306	1415.485005	19778.0	489.006994	0.04875	487.5	[36 - 45]
...	...	...	...	...	...	...	...	...	...
995	0	26.0	38.823284	1997.613745	14778.0	15837.689417	0.09600	960.0	[26 - 35]
996	0	56.0	35.057895	565.051625	15778.0	5760.171811	0.03625	362.5	[56 - 65]
997	0	37.0	33.919225	1377.627448	5778.0	10230.480476	0.04875	487.5	[36 - 45]
998	1	58.5	39.431886	916.046621	27222.0	5773.967912	0.03625	362.5	[56 - 65]
999	0	41.5	41.925698	691.586150	5778.0	8013.263449	0.03625	362.5	[36 - 45]

**Datenerzeugung
im Anhang
dokumentiert!**

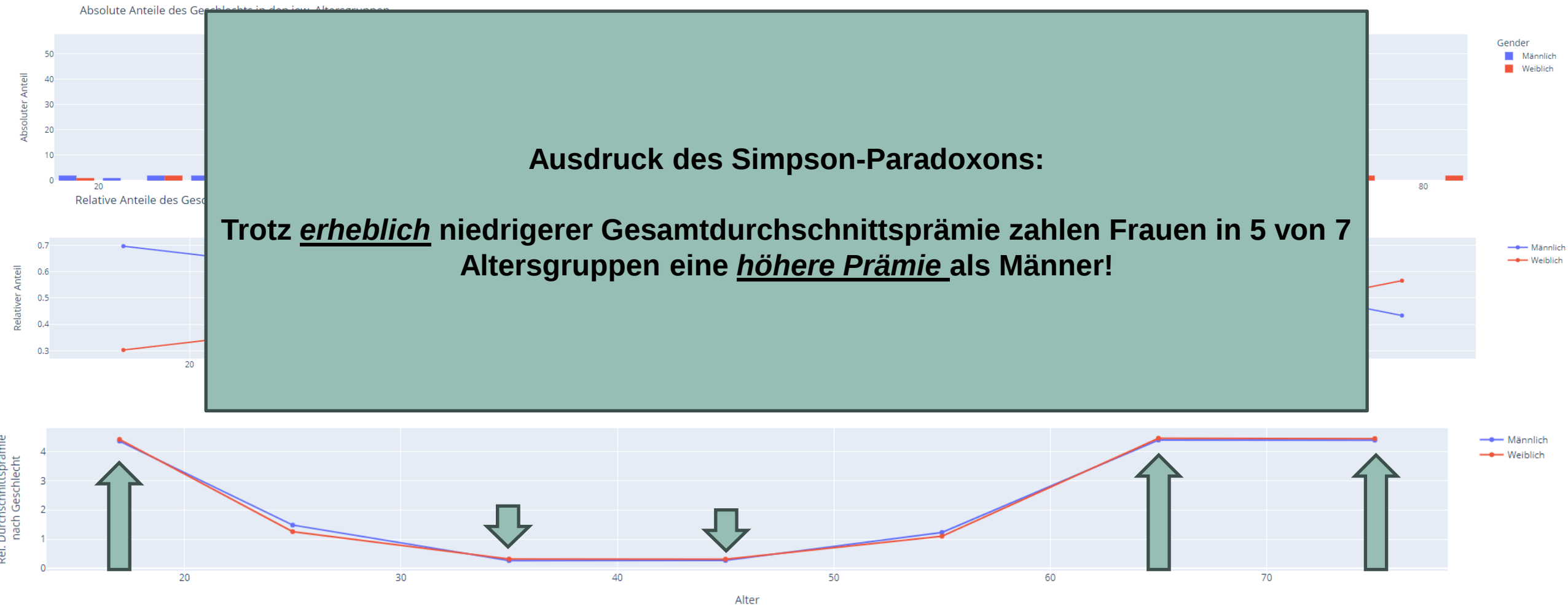
Use Case

Explorative Datenanalyse: Scattermatrix



Use Case

Explorative Datenanalyse: Geschlechterverteilung



Use Case

Preis

```
X = data.copy().drop(['AgeGroup', 'freq', 'Response'], axis=1)
y = data['Response'].copy()

print(X)
lasso = Lasso(alpha=0.1, fit_intercept=True)
lasso.fit(X,y)
print("Ermittelte Koeffizienten:", list(zip(X.columns, lasso.coef_)))
print("Intercept:", lasso.intercept_)

# Berechne mittleren Preis

lasso_pred = lasso.predict(X)
prediction_w_gender = lasso_pred
lasso_mean_price = np.mean(lasso_pred)
print("Mittlere Prämie: ", lasso_mean_price)

# Berechnen Preis für Männer

lasso_index_gender_male = data[data.Gender > 0].index
lasso_mean_price_male = np.mean(lasso_pred[lasso_index_gender_male])
print("Prämie (Männer): ", np.round(lasso_mean_price_male, 2))

# Berechne Preis für Frauen

lasso_index_gender_female = data[data.Gender < 1].index
lasso_mean_price_female = np.mean(lasso_pred[lasso_index_gender_female])
print("Prämie (Frauen): ", np.round(lasso_mean_price_female, 2))
```

	Gender	Age	ShoeSize	EngineDisplacement	Mileage	VehicleValue
0	1	45.5	38.840481	1772.958706	7222.0	4418.959431
1	0	51.5	36.346661	1383.202597	15778.0	14412.902857
2	0	58.0	39.226578	406.564692	7778.0	8423.00328
3	0	56.0	37.788013	1141.595642	19778.0	16716.257354
4	0	42.0	31.635306	1415.485005	19778.0	489.006994
...	...	...	...	...	...	...
995	0	26.0	38.823284	1997.63945	14778.0	15837.689417
996	0	56.0	35.057895	65.051625	15778.0	5760.171811
997	0	37.0	33.919225	1377.627448	5778.0	10230.480476
998	1	58.5	39.431896	916.046621	27222.0	5773.967912
999	0	41.5	41.915698	691.586150	5778.0	8013.263449

[1000 rows x 6 columns]

```
Ermittelte Koeffizienten: [('Gender', 27.45078745142572), ('Age', 1.8734642118792628), ('ShoeSize', -1.0766862546448255), ('EngineDisplacement', 0.14270790712303688), ('Mileage', -0.001425783131184414), ('VehicleValue', 0.0002589522221712431)]
Intercept: 308.0761470610472
Mittlere Prämie: 524.47625
Prämie (Männer): 539.94
Prämie (Frauen): 514.42
```

Mittlere Prämie: 524.47625
Prämie (Männer): 539.94
Prämie (Frauen): 514.42

Use Case

Preis ohne Einbezug des Merkmals „Geschlecht“

```
X_wo_gender = X.drop(['Gender'],axis=1)
#y = data['Response']

lasso_wo_gender = Lasso(alpha=0.2, fit_intercept=True)
lasso_wo_gender.fit(X_wo_gender,y)
print(lasso.coef_)
print(lasso.intercept_)

# Berechne mittleren Preis

lasso_pred_wo_gender = lasso_wo_gender.predict(X_wo_gender)
prediction_wo_gender = lasso_pred_wo_gender
lasso_mean_price_wo_gender = np.mean(lasso_pred_wo_gender)
print("Mittlere Prämie: ",lasso_mean_price_wo_gender)

# Berechnen Preis für Männer

lasso_index_gender_male = data[data.Gender > 0].index
lasso_mean_price_male_wo_gender = np.mean(lasso_pred_wo_gender[lasso_index_gender_male])
print("Prämie (Männer): ",np.round(lasso_mean_price_male_wo_gender,2))

# Berechne Preis für Frauen

lasso_index_gender_female = data[data.Gender < 1].index
lasso_mean_price_female_wo_gender = np.mean(lasso_pred_wo_gender[lasso_index_gender_female])
print("Prämie (Frauen): ",np.round(lasso_mean_price_female_wo_gender,2))
```

```
[ 2.74507675e+01  1.87346421e+00 -1.07668625e+00  1.42707907e-01
 -1.42578313e-03  2.58952222e-04]
308.0761470610472
```

```
Mittlere Prämie: 524.4762500000002
Prämie (Männer): 528.55
Prämie (Frauen): 521.83
```

Der merkmalspezifische Spreiz
wird also reduziert, aber nicht
vollständig ausgeglichen

→ Es besteht indirekte
Diskriminierung über
Ersatzmerkmale.

Use Case

Preis ohne Einbezug des Merkmals „Geschlecht“

Beobachtung: Auch ohne Verwendung des geschützten Attributs verbleibt ein erheblicher (und signifikanter) Unterschied zwischen den Ausprägungen.

→ Entfernen von geschützten Attributen allein garantiert keine „Fairness“.

→ ES besteht indirekte Diskriminierung über Ersatzmerkmale.

```
X_wo_gender = X.drop(['Gender'], axis=1)
#y = d

lasso_
lasso_
print(
print(

# Bere

lasso_
predic
lasso_
print(

# Berechnen Preis für Männer

lasso_index_gender_male = data[data.Gender > 0].index
lasso_mean_price_male_wo_gender = np.mean(lasso_pred_wo_gender[lasso_in
print("Prämie (Männer): ", np.round(lasso_mean_price_male_wo_gender, 2))

# Berechne Preis für Frauen

lasso_index_gender_female = data[data.Gender < 1].index
lasso_mean_price_female_wo_gender = np.mean(lasso_pred_wo_gender[lasso_
print("Prämie (Frauen): ", np.round(lasso_mean_price_female_wo_gender, 2))

[ 2.74507675e+01  1.87346421e+00 -1.07668625e+00  1.42707907e-01
 -1.42578313e-03  2.58952222e-04]
308.0761470610472

Mittlere Prämie: 524.4762500000002
Prämie (Männer): 528.55
Prämie (Frauen): 521.83
```

Mittlere Prämie: 524.476
Prämie (Männer): 528.55
Prämie (Frauen): 521.83

Use Case

Möglichkeiten zur Entfernung von Diskriminierung

[1]

DISCRIMINATION-FREE INSURANCE PRICING

BY

M. LINDHOLM, R. RICHMAN, A. TSANAKAS AND M.V. WÜTHRICH

ABSTRACT

We consider the following question: given information on individual policyholder characteristics, how can we ensure that insurance prices do not discriminate with respect to protected characteristics, such as gender? We address the issues of direct and indirect discrimination, the latter resulting from implicit learning of protected characteristics from nonprotected ones. We provide rigorous mathematical definitions for direct and indirect discrimination, and we introduce a simple formula for discrimination-free pricing, that avoids both direct and indirect discrimination. Our formula works in any statistical model. We demonstrate its application on a health insurance example, using a state-of-the-art generalized linear model and a neural network regression model. An important conclusion is that discrimination-free pricing in general requires collection of policyholders' discriminatory characteristics, posing potential challenges in relation to policyholder's privacy concerns.

KEYWORDS

Causal inference, differentiation, direct discrimination, discriminatory covariates, indirect discrimination, individual policy characteristics, insurance pricing, proxy discrimination.

Der Standardansatz zur Berechnung von diskriminierungsfreien Preisen erfolgt heute praktisch immer durch Angleichung des mittleren Preises.

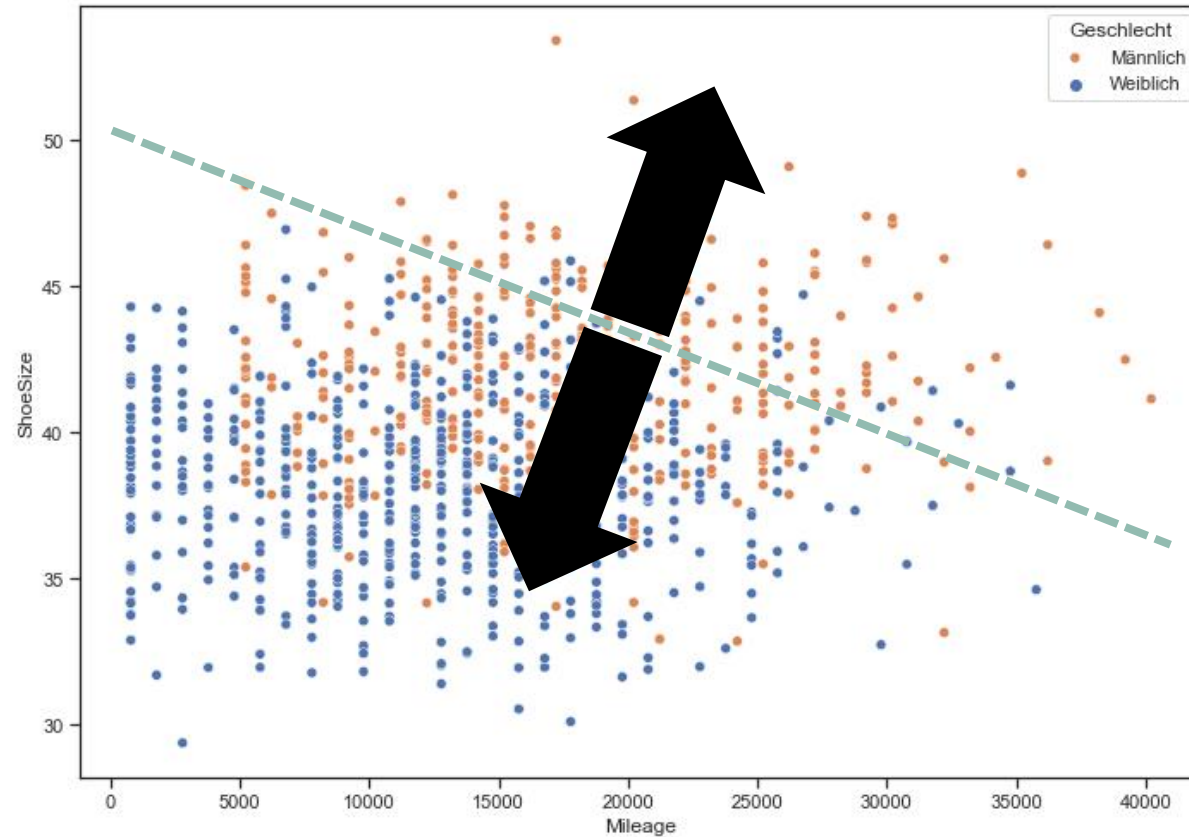
→ Besonderheit des Ansatzes von Lindholm et al. besteht darin, auch indirekte Diskriminierung über Verwendung der jeweiligen Kovarianzen zu minimieren.

Vorsicht: Minimieren bedeutet *nicht*, dass sich indirekte Diskriminierung auch *vollständig* nivellieren lässt.

[1] Lindholm, M., Richman, R., Tsanakas, A., & Wüthrich, M. (2022). DISCRIMINATION-FREE INSURANCE PRICING. ASTIN Bulletin, 52(1), 55-89. doi:10.1017/asb.2021.23

Use Case

Preis mit Ausgleichung indirekter Diskriminierung (1)



Es gibt offensichtlich *unendlich viele Möglichkeiten*, einen fairen Preis unter dem Constraint

$$P_{\text{male}} + b_{\text{male}} = P_{\text{female}} + b_{\text{female}}$$

zu bestimmen:

- a) Jeder Preis wird um konstanten Faktor verändert.
- b) Genau ein Vertrag wird um den Fehlbetrag verteuert.
- c) Irgendetwas dazwischen.

Use Case

Preis mit Ausgleichung indirekter Diskriminierung (2): Beispiel

```
class UnbiasedWeightedLassoEstimator(BaseEstimator):
    def __init__(self, column_name, correction_bias):
        # expected format for bias_dict: {column_name:bias_amount}
        self.biased_column = column_name
        self.correction_bias = correction_bias
        self.subclf = Lasso(alpha=0.1, fit_intercept=True)
        self.lastpred = 0
        self._estimator_type = "regressor"
    def __bias_corrector(self, biased_instance):
        #print(biased_instance)
        bias_correction = 0
        if biased_instance == 1.0: #i.e. male
            bias_correction = self.correction_bias[0]
        if biased_instance == 0.0:
            bias_correction = self.correction_bias[1]
        return bias_correction
    def __sklearn_is_fitted__(self):
        return True
    def fit(self, X, y):
        # Check that X and y have correct shape
        X, y = check_X_y(X, y)
        # Store the classes seen during fit
        self.X_ = X
        self.y_ = y
        self.subclf.fit(self.X[:,1:],self.y_)
        # Return the classifier
        return self
    def predict(self, X):
        # Check if fit has been called
        check_is_fitted(self)
        # Input validation
        X = check_array(X)
        biased_instance = X[:,0]
        bias_correction = 0
        bias_correction = np.array(list(map(self.__bias_corrector,
        return self.subclf.predict(X[:,1:]) + bias_correction - ma
```

```
male_avg_price = np.mean(lasso_wo_gender.predict(partition_male.values[:,1:]))
female_avg_price = np.mean(lasso_wo_gender.predict(partition_female.values[:,1:]))
overall_avg_price = np.mean(lasso_wo_gender.predict(X_wo_gender))
print("Mittl. male price",male_avg_price)
print("Mittl. female price",female_avg_price)
print("Mittl. overall price",overall_avg_price)

male_diff = male_avg_price - overall_avg_price
female_diff = female_avg_price - overall_avg_price

male_prop = num_males/(1000)
female_prop = num_females/(1000)

print(" ")
print("male proportion",male_prop)
print("female proportion",female_prop)
print(" ")
print("Differenz (male):",male_diff)
print("Differenz (female):",female_diff)
print(" ")
print("Hypothese: Mittl. male price",male_avg_price - male_diff)
```

```
Mittl. male price 528.5471551535495
Mittl. female price 521.829489883666
Mittl. overall price 524.4762500000002

male proportion 0.394
female proportion 0.606

Differenz (male): 4.070905153549347
Differenz (female): -2.6467601163341214

Hypothese: Mittl. male price 524.4762500000002
Hypothese: Mittl. female price 524.4762500000002

Gew. Differenz (male): 4.070905153549347
Gew. Differenz (female): -2.6467601163341214

Tatsächl. Mittl. male price 524.3130302310404
Tatsächl. Mittl. female price 524.2856839862822
Tatsächl. Mittl. overall price 524.2964584067169
```

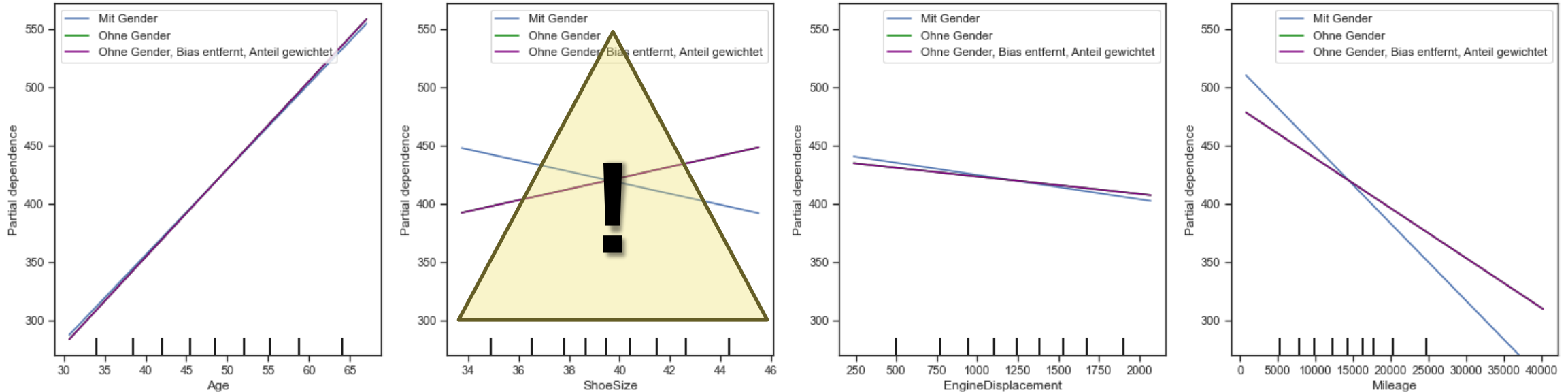
Warum der Aufwand mit überladener BaseEstimator-Klasse?

→ Damit wir die **scikit**-Syntax verwenden können! (z.B. für PDP)

Use Case

Preisfindung: Partial Dependence Plot

Code für
Abbildung im
Anhang
dokumentiert!

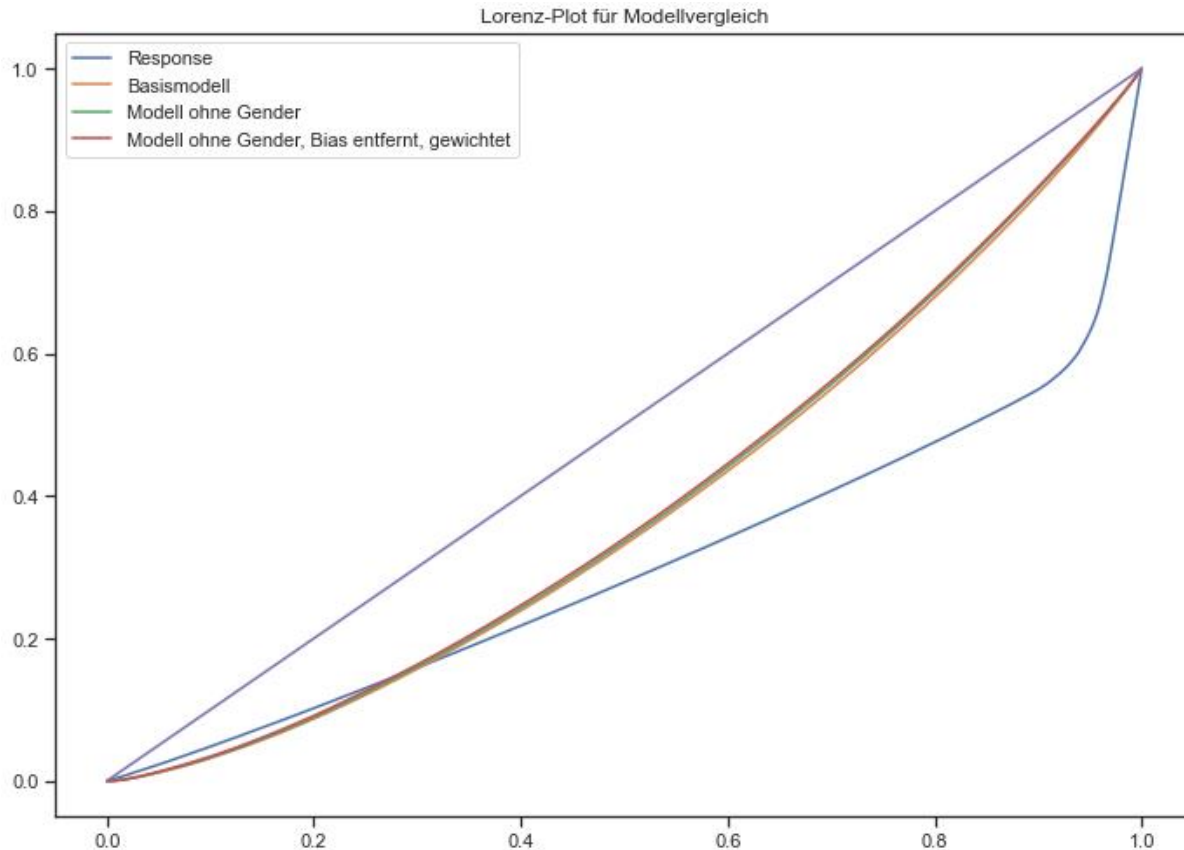


Wesentliche Beobachtungen

- Unterschiede in der Wirkung der „fairen“ Modelle in Bezug auf andere Merkmale identisch.
- Koeffizienten für Alter, Hubraum und Fahrleistung ändern sich leicht.
- Entfernung des Merkmals Geschlecht sorgt für Umkehrung der Wirkung des Merkmals Schuhgröße
→ Rechtfertigungsdilemma in Bezug auf Kausalität!

Use Case

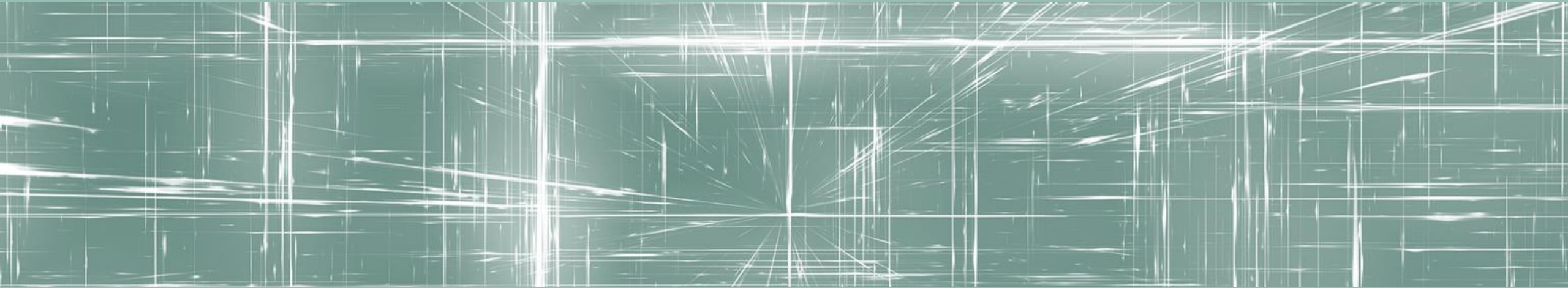
Preisfindung: Lorenz-Plot



Beobachtungen

- Datensatz ohnehin nicht besonders variabel, eher geringe Ungleichheit
- Basismodell und Modelle ohne Gender zeigen gleiche Ungleichheit
 - Diskriminierung nicht effektiv reduziert
- Entfernen des Bias von indirekter Diskriminierung reduziert die Ungleichheit nicht weiter
 - Beträge werden de facto nur umgeschichtet

(Neue) Kennzahlen zur Evaluation von Diskriminierung im Pricing



Bildquelle: Pixabay

Use Case

Disparate Impact

Erinnerung:

Disparate Impact[1] beschreibt adverse Effekte für Personen mit vermeintlich nachteiligen geschützten Attribute, die erst durch die Anwendung von Antidiskriminierungsmaßnahmen Nachteile erleiden[2].

→ In diesem Use Case wären dies gerade diejenigen Verträge, die vom Diskriminierungsausgleich übermäßig stark betroffen wären.

→ Könnte bspw. folgendermaßen formalisiert werden:

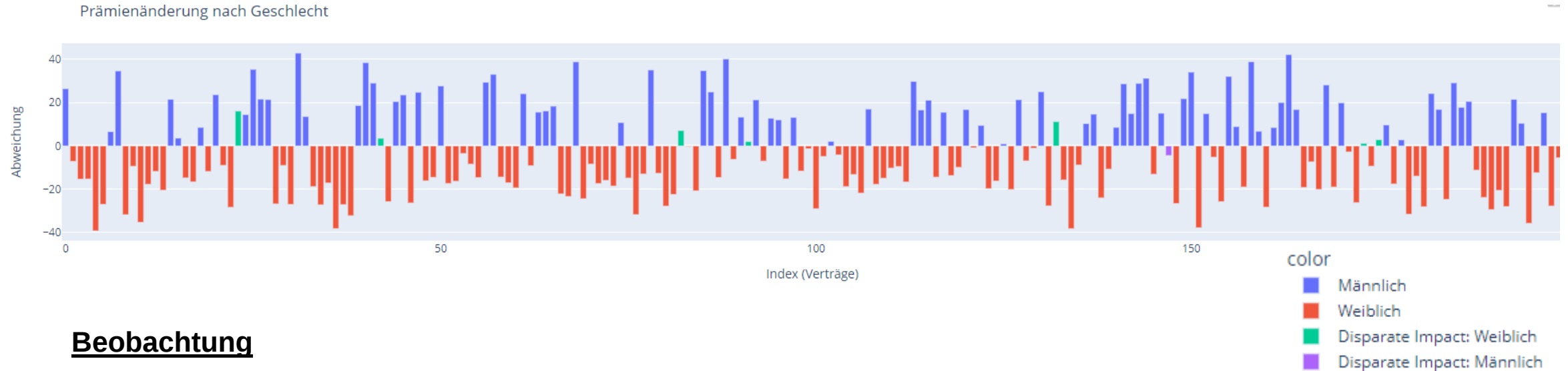
Für Verträge liegt DI vor, sofern die Änderung der Prämie durch Diskriminierungsnivellierung erstens negativ (also höher ausfallend) und zweitens größer als die durchschnittliche Prämienänderung wäre.

[1] https://en.wikipedia.org/wiki/Disparate_impact

[2] z.B. H. Wang, B. Ustun and F. P. Calmon, "On the Direction of Discrimination: An Information-Theoretic Analysis of Disparate Impact in Machine Learning," 2018 ISIT, Vail, CO, USA, 2018, pp. 126-130, doi: 10.1109/ISIT.2018.8437661.

Use Case

Disparate Impact: Basis vs. Modell ohne Geschlecht



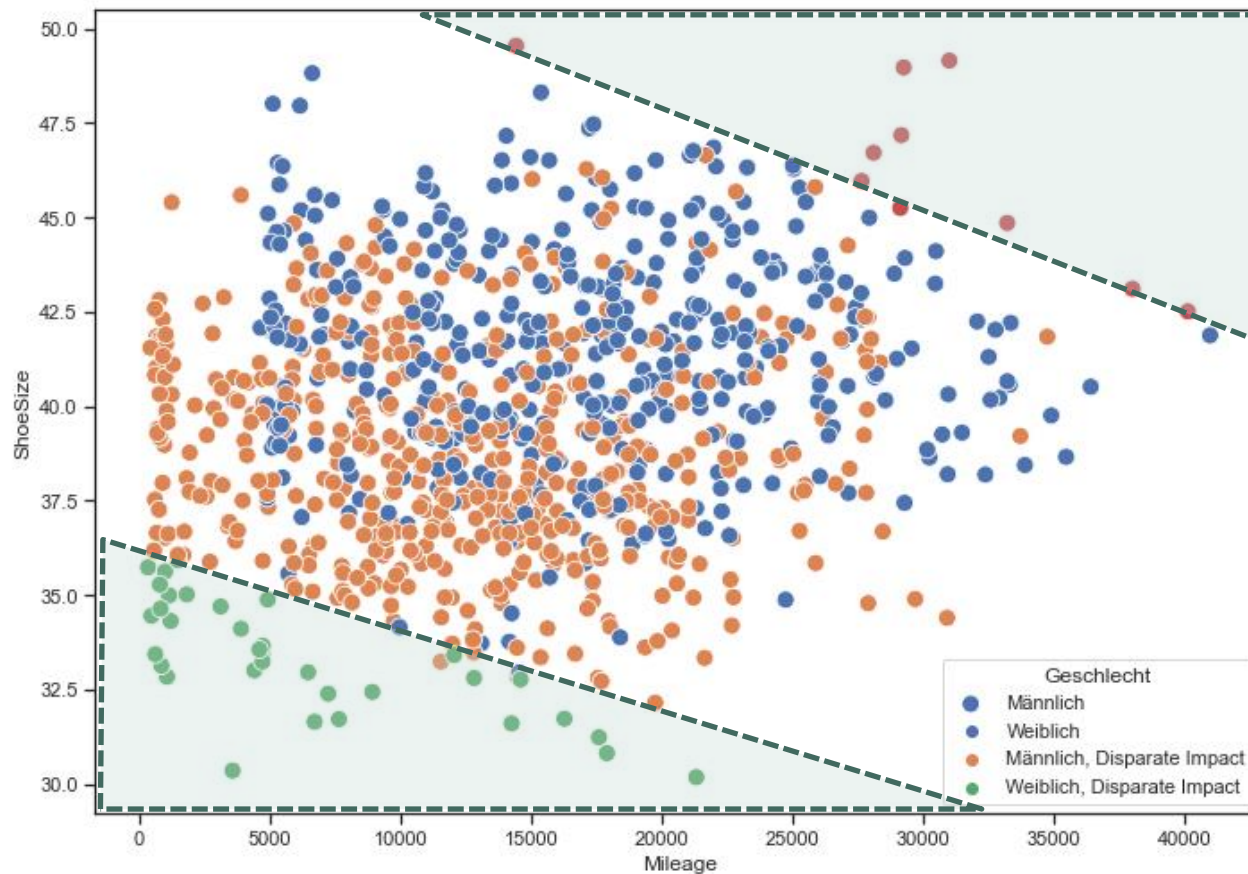
Beobachtung

Beim Modell ohne das Merkmal Geschlecht lassen sich Verträge identifizieren, die stärker als der Durchschnitt betroffen sind:

- *Männer mit sehr großen Füßen* und *Frauen mit sehr geringer Fahrleistung* sind betroffen.
- Anmerkung: „Disparate“ im wortwörtlichen Sinne sind nur Verträge, deren Prämie durch den Effekt **steigt**.

Use Case

Disparate Impact: Basis vs. Modell ohne Geschlecht



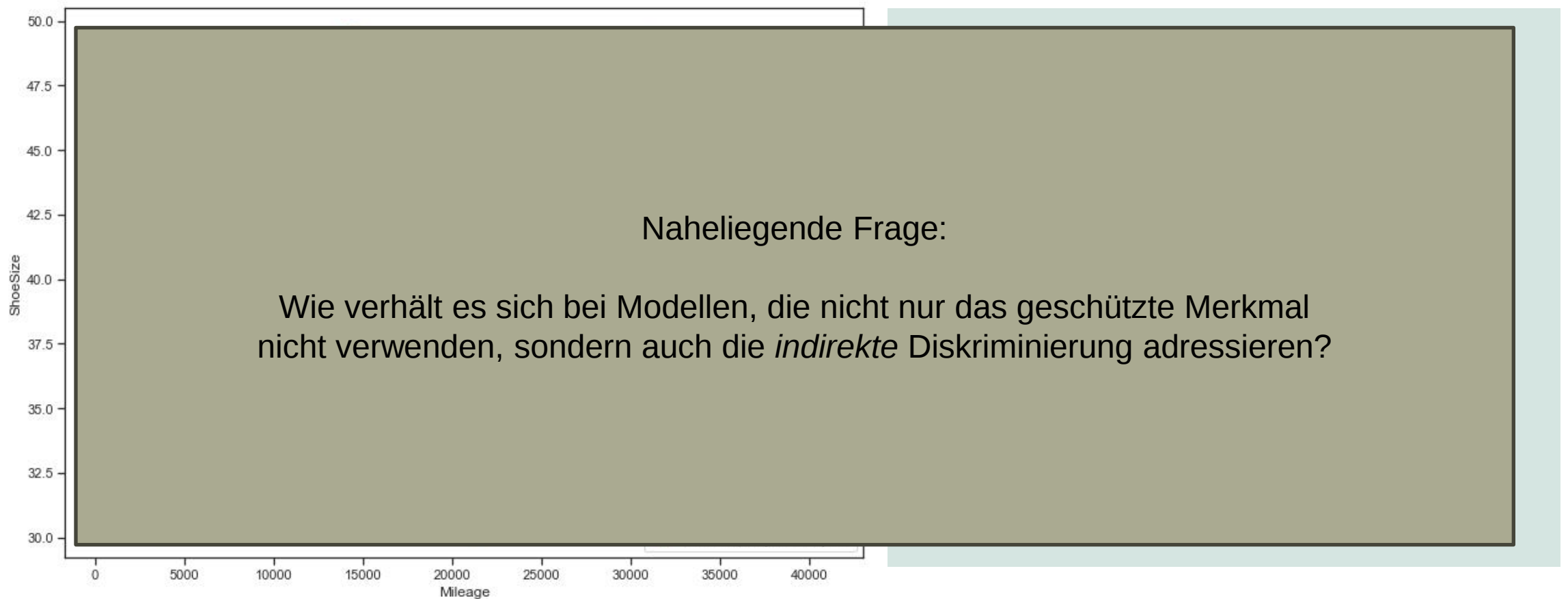
Beobachtung

Verträge, deren Ausprägungen besonders weit vom Durchschnitt entfernt liegen, sind besonders betroffen, wenn auf geschützte Merkmale verzichtet wird.

→ Das ist mglw. nicht das Ziel des Ansinnens! ←

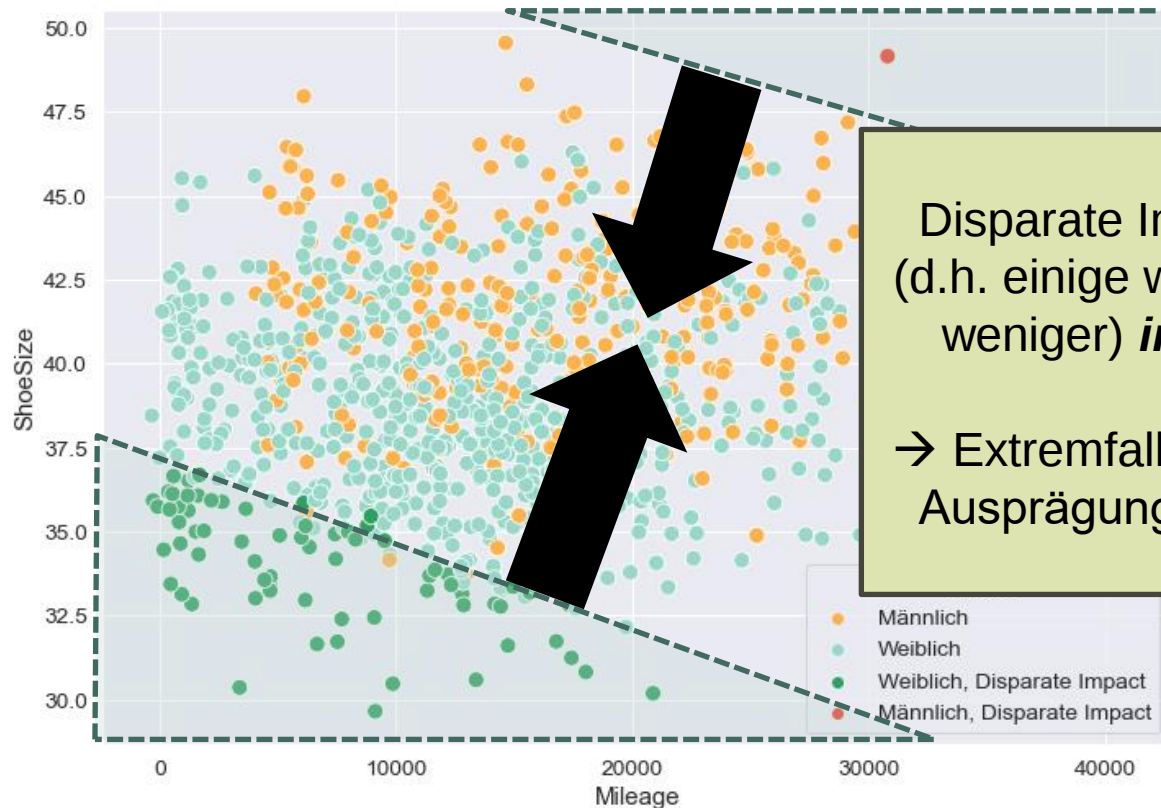
Use Case

Disparate Impact: Basis vs. Modell ohne Geschlecht



Use Case

Disparate Impact: Modell ohne Geschlecht, Bias entfernt



Beobachtung

Disparate Impact kann durch **disproportionale Steuerung** (d.h. einige wenige Verträge zahlen besonders viel mehr oder weniger) **in Richtung der Kovarianz** angepasst werden.

→ Extremfall: Es gibt nur einen einzigen Vertrag mit fraglicher Ausprägung und dieser trägt gesamten Ausgleichsbetrag...

Belastung derjenigen Gruppe, die die Angleichung zu tragen hat.

Use Case

Disparate Impact: Gütemaße

- **Was wird (bisher) gemessen?**

- Gibt es Personen (d.h. Verträge), die Nachteile haben, obwohl sie Vorteile (i.S. eines Lastenausgleichs) erhalten sollten?

- **Was sollte gemessen werden?**

- Gibt es Personen (d.h. Verträge), die um mehr als den mittleren Ausgleichsfaktor be- bzw. entlastet werden?

Use Case

Disparate Impact: Gütemaße

```
# Berechnung von Disparate Impact: Linear u. Quadratisch

gender_data_names_df = pd.DataFrame(gender_data_names, columns=['DF_TYPE'])
DI_values_female = residuum_mit_ohne_gender[gender_data_names_df[gender_data_names_df['DF_TYPE'] == 'Weiblich', Disparate Impact'].index]
print("Ohne Gender (weiblich) | Mittlerer DI\t\t\t", np.mean(DI_values))
print("Ohne Gender (weiblich) | Median DI\t\t\t", np.median(DI_values))
print("Ohne Gender (weiblich) | Mittlerer quadratischer DI\t\t", np.sqrt(np.mean(DI_values_female**2)))
DI_values_male = residuum_mit_ohne_gender[gender_data_names_df[gender_data_names_df['DF_TYPE'] == 'Männlich', Disparate Impact'].index]
print("Ohne Gender (männlich) | Mittlerer DI\t\t\t", np.mean(DI_values_male))
print("Ohne Gender (männlich) | Median DI\t\t\t", np.median(DI_values_male))
print("Ohne Gender (männlich) | Mittlerer quadratischer DI\t\t", np.sqrt(np.mean(DI_values_male**2)))
print(" ")
print("Ohne Gender | Differenz Mittlerer DI:", np.mean(DI_values_female) - np.mean(DI_values_male))
print("Ohne Gender | Differenz Median DI:", np.median(DI_values_female) - np.median(DI_values_male))
print("Ohne Gender | Differenz quadratischer mittlerer DI:", np.sqrt(np.mean(DI_values_female**2)) - np.sqrt(np.mean(DI_values_male**2)))
print(" ")
DI_unbiased_weighted_values_female = residuum_mit_ohne_gender_unbiased_weighted[gender_data_names_df[gender_data_names_df['DF_TYPE'] == 'Weiblich', Disparate Impact'].index]
print("Unbiased Weighted (weiblich) | Mittlerer DI\t\t\t", np.mean(DI_unbiased_weighted_values_female))
print("Unbiased Weighted (weiblich) | Median DI\t\t\t", np.median(DI_unbiased_weighted_values_female))
print("Unbiased Weighted (weiblich) | Mittlerer quadratischer DI\t\t", np.sqrt(np.mean(DI_unbiased_weighted_values_female**2)))
DI_unbiased_weighted_values_male = residuum_mit_ohne_gender_unbiased_weighted[gender_data_names_df[gender_data_names_df['DF_TYPE'] == 'Männlich', Disparate Impact'].index]
print("Unbiased Weighted (männlich) | Mittlerer DI\t\t\t", np.mean(DI_unbiased_weighted_values_male))
print("Unbiased Weighted (männlich) | Median DI\t\t\t", np.median(DI_unbiased_weighted_values_male))
print("Unbiased Weighted (männlich) | Mittlerer quadratischer DI\t\t", np.sqrt(np.mean(DI_unbiased_weighted_values_male**2)))
print(" ")
print("Unbiased Weighted | Differenz Mittlerer DI:", np.mean(DI_unbiased_weighted_values_female) - np.mean(DI_unbiased_weighted_values_male))
print("Unbiased Weighted | Differenz Median DI:", np.median(DI_unbiased_weighted_values_female) - np.median(DI_unbiased_weighted_values_male))
print("Unbiased Weighted | Differenz quadratischer mittlerer DI:", np.sqrt(np.mean(DI_unbiased_weighted_values_female**2)) - np.sqrt(np.mean(DI_unbiased_weighted_values_male**2)))
```

Ohne Gender (weiblich) Mittlerer DI	22.410374488219563
Ohne Gender (weiblich) Median DI	18.582395841707523
Ohne Gender (weiblich) Mittlerer quadratischer DI	26.470634689983704
Ohne Gender (männlich) Mittlerer DI	-18.749593180818238
Ohne Gender (männlich) Median DI	-18.749593180818238
Ohne Gender (männlich) Mittlerer quadratischer DI	18.749593180818238

Ohne Gender Differenz Mittlerer DI:	41.1599676690378
Ohne Gender Differenz Median DI:	37.33198902252576
Ohne Gender Differenz quadratischer mittlerer DI:	7.721041509165467

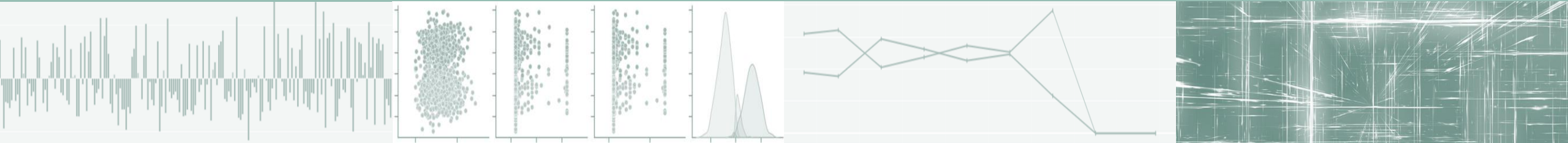
Unbiased Weighted (weiblich) Mittlerer DI	19.01010718377841
Unbiased Weighted (weiblich) Median DI	15.182690141406937
Unbiased Weighted (weiblich) Mittlerer quadratischer DI	23.65659110277824
Unbiased Weighted (männlich) Mittlerer DI	-15.277050632755731
Unbiased Weighted (männlich) Median DI	-15.277050632755731
Unbiased Weighted (männlich) Mittlerer quadratischer DI	15.277050632755731

Unbiased Weighted Differenz Mittlerer DI:	34.28715781653414
Unbiased Weighted Differenz Median DI:	30.45974077416267
Unbiased Weighted Differenz quadratischer mittlerer DI:	8.379540470022508

Bei deskriptiver Betrachtung zeigt das Modell mit Entfernung indirekter Diskriminierung ein besseres Bild.

→ Unklar dabei: Ob ethisch/moralisch eine Gewichtung von besonders problematischen Fällen (→ quadratischer Disparate Impact) sinnvoll und möglich wäre.

Fazit & Ausblick



Bildquelle: Pixabay

Diskriminierungsfreies Pricing

Fazit & Ausblick

- **Zentrales Problem am Vorgehen: Unklare Vergleichsbasis**
 - **Wer legt die relevante Vergleichsgruppe für Prämienausgleich fest?**
 - Versicherer auf Basis des Portfolios? → Selection Bias
 - GDV-Empfehlung? → Nur für Kfz-Versicherte repräsentativ, nicht Gesellschaft
 - Auf Basis des Zensus? → Benachteiligung von Fahrzeughaltern ggü. Nicht-Haltern
 - **Erhebungsmethode selbst muss das Simpson-Paradoxon berücksichtigen**
 - Welche (Ersatz-)Merkmale zählen als indirekte Diskriminierung?
 - **Anwender von Diskriminierungsfreiem Pricing hat Wettbewerbsnachteile**
 - Antiselektion durch Fehlanreize im Preis, erhebliche Gefahr von Unterdeckung

Diskriminierungsfreies Pricing

Fazit & Ausblick

- Technisch stehen verschiedene Ansätze zur Beseitigung von ungewollten diskriminierenden Effekten zur Verfügung, die Preise bezüglich geschützter Attribute nivellieren können
- Diskutable Aspekte bleiben:
 - Der Referenzrahmen zur Beseitigung von Diskriminierung hängt kritisch von der Betrachtungsweise ab:
 - Je nach Aggregierungsebene ergeben sich unterschiedlich stratifizierte Vergleichspreise
→ Abwägung ist politische, nicht aber aktuarielle Frage
 - Beseitigung von Diskriminierung erzeugt zwingendermaßen neue Voreingenommenheiten
 - Diskriminierungsfreie Bepreisung ohne Erzeugung von neuem Disparate Impact ist nicht möglich
→ Auch hier ist die Abwägung politisch zu beantworten
→ In jedem Falle ist „voraussetzender Gehorsam“ eher nicht angezeigt

Vielen Dank!

Prof. Dr. Fabian Transchel

Stiftungsprofessur Data Science der E+S Rückversicherung AG,
Hochschule Harz

Telefon +49 3943 – 305

Telefax +49 3943 – 5305

E-Mail ftranschel@hs-harz.de

Friedrichstraße 57 – 59

38855 Wernigerode

Bildquelle: Pixabay

Anhang: Datenerzeugung

Use Case

Ein diskriminierungsfreundlicher(?) Datensatz

- Zielsetzung des Use Case: Illustration von Simpson-Paradoxon und Disparate Impact
- Ansatz: Kfz-Pricing-Tabelle mit typischen und untypischen Attributen

	Gender	Age	ShoeSize	EngineDisplacement	Mileage	VehicleValue	freq	Response	AgeGroup
0	1	45.5	38.840481	1772.958706	7222.0	4418.959431	0.05500	550.0	[46 - 65]
1	0	51.5	36.346661	1383.202597	15778.0	14412.902857	0.04875	487.5	[46 - 65]
2	0	58.0	39.226578	406.564692	7778.0	8423.109328	0.03625	362.5	[56 - 65]
3	0	56.0	37.788013	1141.595642	19778.0	14116.257354	0.04875	487.5	[56 - 65]
4	0	42.0	31.635306	1415.485005	19778.0	489.006994	0.04875	487.5	[36 - 45]
...	...	...	...	...	...	...	...	...	...
995	0	26.0	38.823284	1997.613745	14778.0	15837.689417	0.09600	960.0	[26 - 35]
996	0	56.0	35.057895	565.051625	15778.0	5760.171811	0.03625	362.5	[56 - 65]
997	0	37.0	33.919225	1377.627448	5778.0	10230.480476	0.04875	487.5	[36 - 45]
998	1	58.5	39.431886	916.046621	27222.0	5773.967912	0.03625	362.5	[56 - 65]
999	0	41.5	41.925698	691.586150	5778.0	8013.263449	0.03625	362.5	[36 - 45]

Use Case

Datenerzeugung (1)

- Libraries

```
# Base tools

import pandas as pd
import numpy as np

# Modelling

from sklearn.linear_model import Lasso
from sklearn.base import BaseEstimator
from sklearn.utils.validation import check_X_y, check_array, check_is_fitted
from sklearn.utils.multiclass import unique_labels
from sklearn.metrics import euclidean_distances

# Plotting

import plotly.express as px
import plotly.figure_factory as ff
import plotly.graph_objects as go
import seaborn as sns
import matplotlib.pyplot as plt

# Inspection

from sklearn.inspection import partial_dependence
from sklearn.inspection import PartialDependenceDisplay
```

Use Case

Datenerzeugung (2)

```
# Set a random seed for reproducibility
np.random.seed(42)

# Generate synthetic data for features
num_samples = 1000

# Generate age data (assuming a normal distribution centered around 40 with standard deviation 10)
#age = np.clip(np.random.normal(loc=49, scale=15, size=num_samples).astype(int),17,99)
age = 0.5 * np.random.uniform(18,79,size=num_samples).astype(int) + 0.5 * np.clip(np.random.normal(loc=49, scale=15, size=num_samples).astype(int),17,99)

# Generate gender data (assuming binary genders)
gender = num_samples * [0] # Female = 0 Male = 1
for i in range(num_samples):
    if age[i] < 35 and np.random.choice([0,1],p=[0.25,0.75],size=1)[0] < 1:
        gender[i] = 1
    elif age[i] > 55 and np.random.choice([0,1],p=[0.1,0.9],size=1)[0] < 1:
        gender[i] = 1
    else:
        gender[i] = np.random.choice([0,1],p=[0.65,0.35],size=1)[0]

# Generate shoe size data (assuming a normal distribution centered around 8 with standard deviation 1.5)
shoe_size = np.random.normal(loc=40, scale=3, size=num_samples)

# Generate engine displacement data (assuming a normal distribution centered around 2000 with standard deviation 500)
engine_displacement = np.random.normal(loc=1250, scale=550, size=num_samples)

# Generate driving distance data (assuming a normal distribution centered around 15000 with standard deviation 5000)
mileage = np.round(np.clip(np.random.normal(loc=15000, scale=7500, size=num_samples),3000,100000),decimals=-3)

# Generate vehicle value data (assuming a normal distribution centered around 12500 with standard deviation 5000)
vehicle_value = np.clip(np.random.normal(loc=12500, scale=5000, size=num_samples),250,1000000)

# Create a DataFrame with the generated data
data = pd.DataFrame({
    'Gender': gender,
    'Age': age,
    'ShoeSize': shoe_size,
    'EngineDisplacement': engine_displacement,
    'Mileage': mileage,
    'VehicleValue': vehicle_value
})
```

Use Case

Datenerzeugung (3)

```
# Create desired correlations
data['ShoeSize'] += 1.65 * data['Gender'].apply(lambda x: 1 if x == 1 else -1)
data['EngineDisplacement'] += 0.45 * data['Gender'].apply(lambda x: 1 if x == 1 else -1)
data['Mileage'] += 2222 * data['Gender'].apply(lambda x: 1 if x == 1 else -1)
data['VehicleValue'] += 0.175 * data['Age']

def age_interaction(age,gender):
    age_factor = 0
    if(age) < 35:
        age_factor = 0.2 * ((age-30)**2) + 0.5 * gender
    if(age) > 55:
        age_factor = 0.15 * ((age-65)**3) + 0.5 * gender
    return age_factor

def mileage_interaction(mileage):
    mileage_factor = 0
    mileage_factor = 1 / (1-np.exp(-mileage*20))
    return mileage_factor

data['freq'] = np.array([0.005]*num_samples) * (
    np.array(list((np.clip(np.array(list(map(lambda x,y:age_interaction(x,y),age,gender))),0,10))))
    + np.array(list(map(mileage_interaction,mileage)))
data['Response'] = data['freq'] * 10000
```

Use Case

Code für Partial Dependence Plot

```
fig, ax = plt.subplots(figsize=(12, 6))
ax.set_title("Mit Gender")
disp_wi_gender = PartialDependenceDisplay.from_estimator(lasso, X, [1,2,3,4], ax=ax, line_kw={"color": "blue"})
fig, ax = plt.subplots(figsize=(12, 6))
ax.set_title("Ohne Gender")
disp_wo_gender = PartialDependenceDisplay.from_estimator(lasso_wo_gender, X_wo_gender, features = [0, 1, 2, 3], ax = ax, line_kw={"color": "green"})
#fig, ax = plt.subplots(figsize=(12, 6))
#ax.set_title("Ohne Gender, Bias entfernt")
#disp_wo_gender_bias_removal = PartialDependenceDisplay.from_estimator(lasso_wo_gender_unbiased, X, features = [1,2,3,4], ax = ax, line_kw={"color": "red"})
fig, ax = plt.subplots(figsize=(12, 6))
ax.set_title("Ohne Gender, Bias entfernt, Anteil gewichtet")
disp_wo_gender_bias_removal_weighted = PartialDependenceDisplay.from_estimator(lasso_wo_gender_unbiased_weighted, X, features = [1,2,3,4], ax = ax, line_kw={"color": "purple"})
```

```
fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize=(25, 6))
disp_wi_gender.plot(ax=[ax1, ax2, ax3, ax4], line_kw={"label": "Mit Gender"})
disp_wo_gender.plot(ax=[ax1, ax2, ax3, ax4], line_kw={"label": "Ohne Gender", "color": "green"})
#disp_wo_gender_bias_removal.plot(ax=[ax1, ax2, ax3, ax4], line_kw={"label": "Ohne Gender, Bias entfernt", "color": "red"})
disp_wo_gender_bias_removal_weighted.plot(ax=[ax1, ax2, ax3, ax4], line_kw={"label": "Ohne Gender, Bias entfernt, Anteil gewichtet", "color": "purple"})
ax1.legend()
ax2.legend()
ax3.legend()
ax4.legend()
```