

CS 4530

Fundamentals of Software Engineering

Lecture 12.1: Engineering Software for Equity

Jonathan Bell, Adeel Bhutta, Ferdinand Vesely, Mitch Wand
Khoury College of Computer Sciences

© 2022, released under [CC BY-SA](#)

Learning Objectives for this Lesson

By the end of this lesson, you should be able to...

- Suggest some ways in which software can cause inadvertent harm or amplify inequities, with examples
- Explain why the software engineer has a powerful role to play in avoiding such harms.

From SE @ Google:

As new as the field of software engineering is, we're newer still at understanding the impact it has on underrepresented people and diverse societies. ... [We must recognize] the increasing imbalance of power between those who make development decisions that impact the world and those who simply must accept and live with those decisions that sometimes disadvantage already marginalized communities globally.

A good software engineer will recognize potentials for inequity from their software.



“One mark of an exceptional engineer is the ability to understand how products can advantage and disadvantage different groups of human beings. Engineers are expected to have technical aptitude, but they should also have the discernment to know when to build something and when not to.”

**-Demma Rodriguez,
Head of Equity Engineering @ Google**

Algorithmic sentencing systems can discriminate against Black defendants

Example: the COMPAS Sentencing Tool

	ALL DEFENDANTS	WHITE DEFENDANTS	BLACK DEFENDANTS
Labeled Higher Risk, But Didn't Re-Offend	32.4%	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	37.4%	47.7%	28.0%

Algorithmic bias can discriminate against poorer consumers

Websites Vary Prices, Deals Based on Users' Information



SNAPSAFE; HOME DEPOT; ROSETTA STONE

By Jennifer Valentino-DeVries, Jeremy Singer-Vine and Ashkan Soltani
December 24, 2012

FairTest: Discovering Unwarranted Associations in Data-Driven Applications*

Florian Tramèr¹, Vaggelis Atlidakis², Roxana Geambasu², Daniel Hsu², Jean-Pierre Hubaux³, Mathias Humbert⁴, Ari Juels⁵, Huang Lin³

¹Stanford, ²Columbia University, ³EPFL, ⁴Saarland University, ⁵Cornell Tech, Jacobs Institute

Abstract—In a world where traditional notions of privacy are increasingly challenged by the myriad companies that collect and analyze our data, it is important that decision-making entities are held accountable for unfair treatments arising from irresponsible data usage. Unfortunately, a lack of appropriate methodologies and tools means that even identifying unfair or discriminatory effects can be a challenge in practice.

We introduce the *unwarranted associations (UA) framework*, a principled methodology for the discovery of unfair, discriminatory, or offensive user treatment in data-driven applications. The UA framework unifies and rationalizes a number of prior attempts at formalizing algorithmic fairness. It uniquely combines multiple investigative primitives and fairness metrics with broad applicability, granular exploration of unfair treatment in user subgroups, and incorporation of natural notions of utility that may account for observed disparities.

We instantiate the UA framework in *FairTest*, the first comprehensive tool that helps developers check data-driven applications for unfair user treatment. It enables scalable and statistically rigorous investigation of associations between application outcomes (such as prices or premiums) and sensitive user attributes (such as race or gender). Furthermore, *FairTest* provides *debugging capabilities* that let programmers rule out potential confounders for observed unfair effects.

We report on use of *FairTest* to investigate and in some cases address disparate impact, offensive labeling, and uneven rates of algorithmic error in four data-driven applications. As examples, our results reveal subtle biases against older populations in the distribution of error in a predictive health application and offensive racial labeling in an image tagger.

1. Introduction

Today's applications collect and mine vast quantities of personal information. Such data can boost applications' utility by personalizing content and recommendations, increase business revenue via targeted product placement, and improve a wide range of socially beneficial services, such as healthcare, disaster response, and crime prevention.

The collection and use of such data raise two important challenges. First, massive data collection is perceived by many as a major threat to traditional notions of individual privacy. Second, the use of personal data for algorithmic

decision-making can have unintended and harmful consequences, such as unfair or discriminatory treatment of users.

In this paper, we deal with the latter challenge. Despite the personal and societal benefits of today's data-driven world, we argue that companies that collect and use our data have a responsibility to ensure equitable user treatment. Indeed, European and U.S. regulators, as well as various policy and legal scholars, have recently called for increased *algorithmic accountability*, and in particular for decision-making tools to be audited and "tested for fairness" [1], [2].

There have been many recent reports of unfair or discriminatory effects in data-driven applications, mostly qualified as unintended consequences of data heuristics or overlooked bugs. For example, Google's image tagger was found to associate racially offensive labels with images of black people [3]; the developers called the situation a bug and promised to remedy it as soon as possible. In another case [4], *Wall Street Journal* investigators showed that Staples' online pricing algorithm discriminated against lower-income people. They referred to the situation as an "unintended consequence" of Staples's seemingly rational decision to adjust online prices based on user proximity to competitors' stores. This led to higher prices for low-income customers, who generally live farther from these stores.

Staples' intentions aside, it is evidently difficult for programmers to foresee all the subtle implications and risks of data-driven heuristics. Moreover, these risks will only increase as data is passed through increasingly complex machine learning (ML) algorithms whose associations and inferences may be impossible to anticipate.

We argue that such algorithmic biases are new kinds of *bugs*, specific to modern, data-driven applications, that programmers should proactively check for, debug, and fix with the same rigor as they apply to other security and privacy bugs. Such bugs can offend and even harm users, and cause programmers and businesses embarrassment, mistrust, and potentially loss of revenue. They may also be symptoms of a malfunction of a data-driven algorithm, such as a ML algorithm exhibiting poor accuracy for minority groups that are underrepresented in its training set [5].

We refer to such bugs generically as *unwarranted associations*. Understanding and identifying unwarranted associations is an important step towards holding automated decision-making entities *accountable* for unfair practices, thus also providing incentive for the adoption of corrective measures [1], [2], [6], [7].

The Unwarranted Associations Framework. In order to

Training AI systems can have serious impacts on climate.

The Register

{* AI + ML *}

AI me to the Moon... Carbon footprint for 'training GPT-3' same as driving a new car to the Moon and back

Get ready for Energy Star stickers on your car

Katyanna Quach Wed 4 Nov 2020 // 07:59 UTC

Training OpenAI's giant GPT-3 text-generating model, a car to the Moon and back, computer scientists revealed today.

More specifically, they estimated teaching the new model, which Microsoft data center using Nvidia GPUs required, which using the average carbon intensity of America would have produced 85,000 kg of CO₂ equivalents, the same amount produced by a new car in Europe driving 700,000 km, or 435,000 miles, which is about twice the distance between Earth and the Moon, some 480,000 miles. Phew.

Not to mention bitcoin mining!

Consumption	CO ₂ e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Driving a car, 1 year	11,023
Driving a car, avg, 1 year	36,156
Driving a car, 1 lifetime	126,000
Model (GPU)	
Training, SRL)	39
Experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

[“Energy and Policy Considerations for Deep Learning in NLP” Emma Strubell, Ananya Ganesh, Andrew McCallum, in Proceedings of ACL 2019](#)

https://www.theregister.com/2020/11/04/gpt3_carbon_footprint_estimate/

Poor user interfaces can discriminate against differently-abled people.

Inclusivity and Accessibility: Domino's Pizza LLC v. Robles

Domino's Would Rather Go to the Supreme Court Than Make Its Website Accessible to the Blind

Rather than developing technology to support users with disabilities, the pizza chain is taking its fight to the top

by Brenna Houck | @EaterDetroit | Jul 25, 2019, 6:00pm EDT

f   SHARE



Jul 15 2019	Brief amicus curiae of Washington Legal Foundation filed.
Jul 15 2019	Brief amici curiae of Retail Litigation Center, Inc., et al. filed.
Jul 15 2019	Brief amicus curiae of Cato Institute filed.
Jul 15 2019	Brief amicus curiae of Restaurant Law Center filed.
Jul 15 2019	Brief amici curiae of Chamber of Commerce of the United States of America, et al. filed.

[“Domino’s Would Rather Go to the Supreme Court Than Make Its Website Accessible to the Blind” by Brenna Houck, Eater Detroit](#)

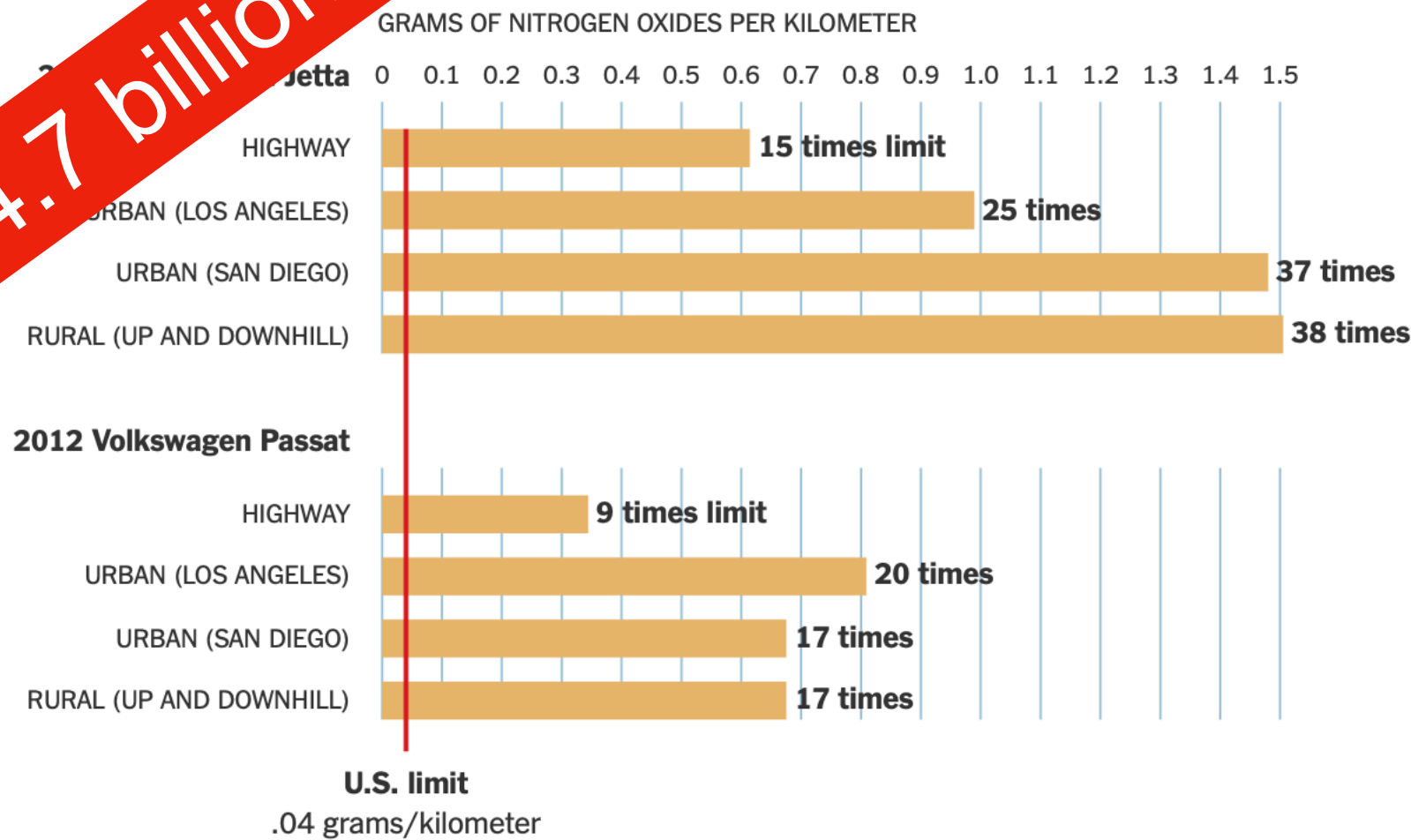
Software Systems can be used to evade regulation.

Example: Volkswagen diesel emissions

The Emissions Tests That Led to the Discovery of VW's Cheating

The on-road testing in May 2014 that led the California Air Resources Board to investigate Volkswagen was conducted by researchers at West Virginia University. They tested emissions from two VW Jettas equipped with the 2-liter turbocharged 4-cylinder diesel engine. The researchers found that when tested on the road, some cars emitted almost 40 times the allowed levels of nitrogen oxides.

Average emissions of nitrogen oxides in on-road testing



Source: Arvind Thiruvengadam, Center for Alternative Fuels, Engines and Emissions at West Virginia University

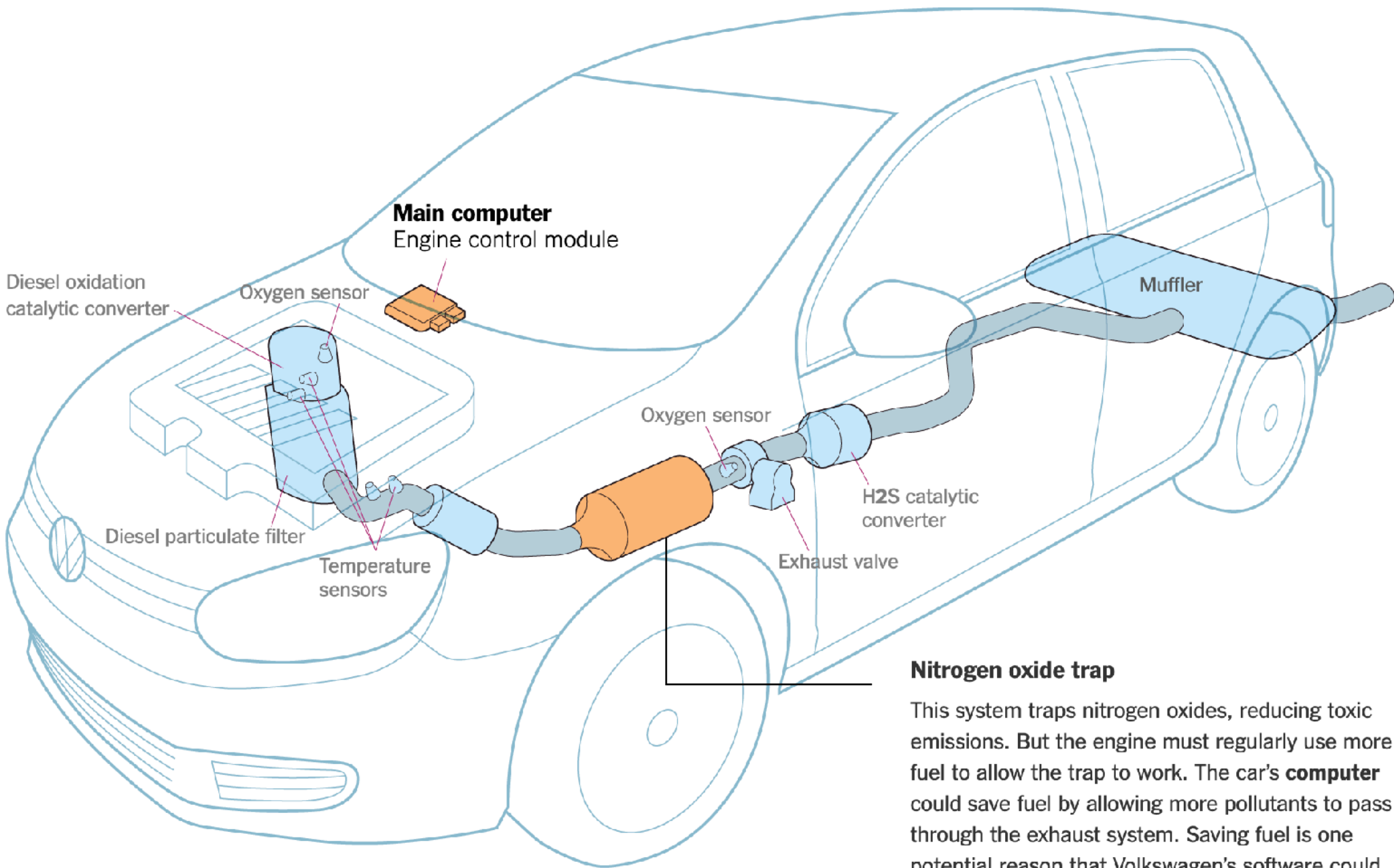


Illustration by Guilbert Gates | Source: Volkswagen, The International Council on Clean Transportation

Bias is the Default

Example: Google Photos auto-tagging (2015)



THE WALL STREET JOURNAL.



DIGITS

Google Mistakenly Tags Black People as ‘Gorillas,’ Showing Limits of Algorithms

By [Alistair Barr](#)

Updated July 1, 2015 3:41 pm ET

 SHARE  TEXT

Google is a leader in artificial intelligence and machine learning. But the company’s computers still have a lot to learn, judging by a major blunder by its Photos app this week.

The app tagged two black people as “Gorillas,” according to Jacky Alciné, a Web developer who spotted the error and tweeted a photo of it.

“Google Photos, y’all f**ked up. My friend’s not a gorilla,” [he wrote on Twitter](#).

Google apologized and said it’s tweaking its algorithms to fix the problem.

“We’re appalled and genuinely sorry that this happened,” a company

<https://www.wsj.com/articles/BL-DGB-42522>

<https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/>



WIRED

BACKCHANNEL

BUSINESS

CULTURE

GEAR

IDEAS

MORE ▾

SIGN IN



TOM SIMONITE

BUSINESS

01.11.2018 07:00 AM

When It Comes to Gorillas, Google Photos Remains Blind

Google promised a fix after its photo-categorization software labeled black people as gorillas in 2015. More than two years later, it hasn't found one.

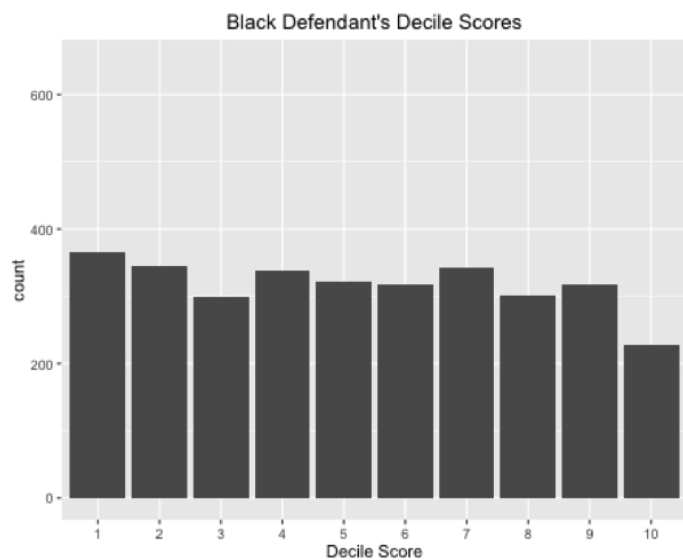
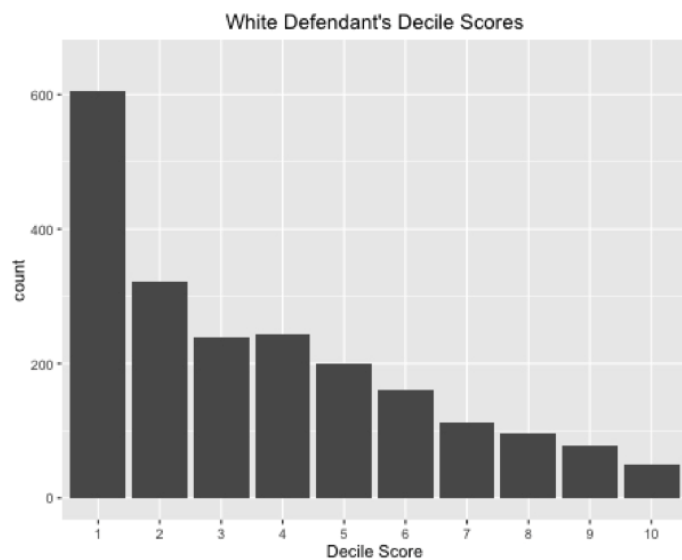


Reflecting on these examples

Personal philosophies and business cases

Algorithmic Bias: COMPAS Sentencing Tool

	ALL DEFENDANTS	WHITE DEFENDANTS	BLACK DEFENDANTS
Labeled Higher Risk, But Didn't Re-Offend	32.4%	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	37.4%	47.7%	28.0%



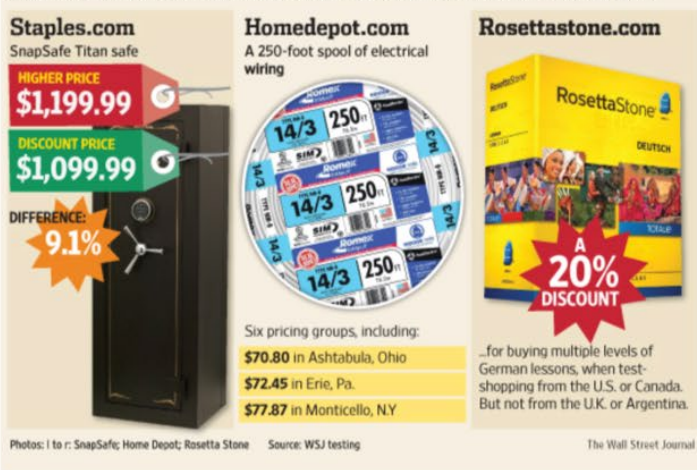
Analysis of Broward County, FL data: “How We Analyzed the COMPAS Recidivism Algorithm” by Jeff Larson, Surya Mattu, Lauren Kirchner and Julia Angwin

Algorithmic Bias: Price Discrimination

Websites Vary Prices, Deals Based on Users' Information

Getting Different Deals Online

A Journal examination found online retailers adjusted prices by a shopper's location, among other factors



PHOTOS: i to r: SnapSafe; Home Depot; Rosetta Stone

By Jennifer Valentino-DeVries, Jeremy Singer-Vine and Ashkan Soltani
December 24, 2012

<https://www.wsj.com/articles/SB1000142412788732377204578189391813881534>

2017 IEEE European Symposium on Security and Privacy

FairTest: Discovering Unwarranted Associations in Data-Driven Applications*

Florian Tramèr¹, Vaggelis Athidakis¹, Roxana Geambasu², Daniel Hsu³,
Jean-Pierre Hubaux¹, Mathias Humbert⁴, Ari Juels⁵, Huang Lin⁶

¹Stanford, ²Columbia University, ³EPFL, ⁴Saarland University, ⁵Cornell Tech, Jacobs Institute

Abstract—In a world where traditional notions of privacy are increasingly challenged by the myriad companies that collect and analyze our data, it is important that decision-making entities are held accountable for unfair treatments arising from irresponsible data usage. Unfortunately, a lack of appropriate methodologies and tools means that even identifying unfair or discriminatory effects can be a challenge in practice.

We introduce the *unwarranted associations* (UA) framework, a principled methodology for the discovery of unfair, discriminatory, or offensive user treatment in data-driven applications.

The UA framework unifies and rationalizes a number of prior attempts at formalizing algorithmic fairness. It uniquely combines multiple investigative primitives and fairness metrics with broad applicability, granular exploration of unfair treatment in user subgroups, and incorporation of natural notions of utility that may account for observed disparities.

We instantiate the UA framework in *FairTest*, the first comprehensive tool that helps developers check data-driven applications for unfair user treatment. It enables scalable and statistically rigorous investigation of associations between application outcomes (such as prices or premiums) and sensitive user attributes (such as race or gender). Furthermore, *FairTest* provides debugging capabilities that let programmers rule out potential confounders for observed unfair effects.

We report on use of *FairTest* to investigate and in some cases address disparate impact, offensive labeling, and uneven rates of algorithmic error in four data-driven applications. As examples, our results reveal subtle biases against older populations in the distribution of error in a predictive health application and offensive racial labeling in an image tagger.

We argue that such algorithmic biases are new kinds of bugs, specific to modern, data-driven applications, that programmers should proactively check for, debug, and fix with the same rigor as they apply to other security and privacy bugs. Such bugs can offend and even harm users, and cause programmers and businesses embarrassment, mistrust, and potentially loss of revenue. They may also be symptoms of a malfunction of a data-driven algorithm, such as a ML algorithm exhibiting poor accuracy for minority groups that are underrepresented in its training set [5].

We refer to such bugs generally as *unwarranted associations*. Understanding and identifying unwarranted associations is an important step towards holding automated decision-making entities accountable for unfair practices, thus also providing incentive for the adoption of corrective measures [1], [2], [6], [7].

*Work done while the first author was at EPFL.

1. Introduction

Today's applications collect and mine vast quantities of personal information. Such data can boost applications' utility by personalizing content and recommendations, increase business revenue via targeted product placement, and improve a wide range of socially beneficial services, such as healthcare, disaster response, and crime prevention. The collection and use of such data raise two important challenges. First, massive data collection is perceived by many as a major threat to traditional notions of individual privacy. Second, the use of personal data for algorithmic

decision-making can have unintended and harmful consequences, such as unfair or discriminatory treatment of users.

In this paper, we deal with the latter challenge. Despite the personal and societal benefits of today's data-driven world, we argue that companies that collect and use our data have a responsibility to ensure equitable user treatment. Indeed, European and U.S. regulators, as well as various policy and legal scholars, have recently called for increased algorithmic accountability, and in particular for decision-making tools to be audited and “tested for fairness” [1], [2].

There have been many recent reports of unfair or discriminatory effects in data-driven applications, mostly qualified as unintended consequences of data heuristics or overlooked bugs. For example, Google's image tagger was found to associate racially offensive labels with images of black people [3]; the developers called the situation a bug and promised to remedy it as soon as possible. In another case [4], *Wall Street Journal* investigators showed that Staples' online pricing algorithm discriminated against lower-income people. They referred to the situation as an “unintended consequence” of Staples' seemingly rational decision to adjust online prices based on user proximity to competitors' stores. This led to higher prices for low-income customers, who generally live farther from these stores.

Staples' intentions aside, it is evidently difficult for programmers to foresee all the subtle implications and risks of data-driven heuristics. Moreover, these risks will only increase as data is passed through increasingly complex machine learning (ML) algorithms whose associations and inferences may be impossible to anticipate.

We argue that such algorithmic biases are new kinds of bugs, specific to modern, data-driven applications, that programmers should proactively check for, debug, and fix with the same rigor as they apply to other security and privacy bugs. Such bugs can offend and even harm users, and cause programmers and businesses embarrassment, mistrust, and potentially loss of revenue. They may also be symptoms of a malfunction of a data-driven algorithm, such as a ML algorithm exhibiting poor accuracy for minority groups that are underrepresented in its training set [5].

We refer to such bugs generally as *unwarranted associations*. Understanding and identifying unwarranted associations is an important step towards holding automated decision-making entities accountable for unfair practices, thus also providing incentive for the adoption of corrective measures [1], [2], [6], [7].

The Unwarranted Association Framework. In order to

© 2017, Florian Tramèr. Under license to IEEE.
DOI 10.1109/EuroSP.2017.29

401

IEEE
COMPUTER
SOCIETY

Inclusivity and Accessibility: Domino's Pizza LLC v. Robles

Domino's Would Rather Go to the Supreme Court Than Make Its Website Accessible to the Blind

Rather than developing technology to support users with disabilities, the pizza chain is taking its fight to the top

by Brenna Houck | @EaterDetroit | Jul 25, 2019, 6:00pm EDT

f SHARE



“Domino's Would Rather Go to the Supreme Court Than Make Its Website Accessible to the Blind” by Brenna Houck, Eater Detroit

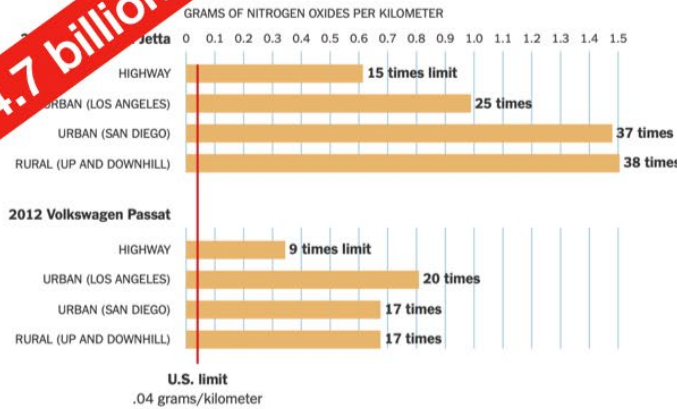
Jul 15 2019	Brief amicus curiae of Washington Legal Foundation filed.
Jul 15 2019	Brief amici curiae of Retail Litigation Center, Inc., et al. filed.
Jul 15 2019	Brief amicus curiae of Cato Institute filed.
Jul 15 2019	Brief amicus curiae of Restaurant Law Center filed.
Jul 15 2019	Brief amici curiae of Chamber of Commerce of the United States of America, et al. filed.

Evading regulation: Volkswagen

The Emissions Tests That Led to the Discovery of VW's Cheating

The on-road testing in May 2014 that led the California Air Resources Board to investigate Volkswagen was conducted by researchers at West Virginia University. They tested emissions from two VW Passats equipped with the 2-liter turbocharged 4-cylinder diesel engine. The results showed that when tested on the road, some cars emitted almost 40 times the allowed levels of nitrogen oxides.

Average emissions of nitrogen oxides in on-road testing



Source: Arvind Thiruvengadam, Center for Alternative Fuels, Engines and Emissions at West Virginia University

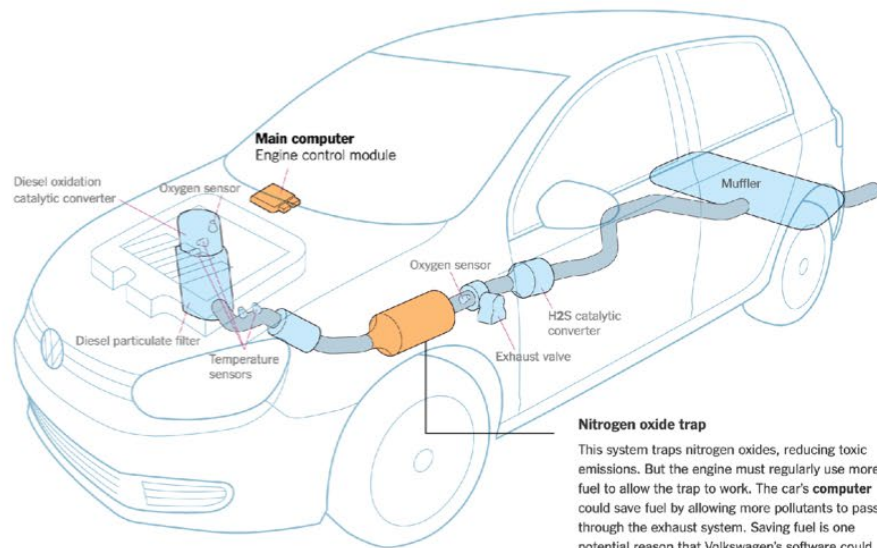


Illustration by Guilbert Gates | Source: Volkswagen, The International Council on Clean Transportation

“How Volkswagen’s ‘Defeat Devices’ Worked” By Guilbert Gates, Jack Ewing, Karl Russell and Derek Watkins

More than “don’t be evil”

Engineering equitable software requires conscious effort

- How do we determine what “the right thing” is?
- How do we convince our investors/managers to take this action?

This lesson was about the harms that software can inflict

You should now be able to...

- Suggest some ways in which software can cause inadvertent harm or amplify inequities, with examples
- Explain why the software engineer has a powerful role to play in avoiding such harms.