CS 4530
Fundamentals of Software Engineering

Module 18: Engineering Software for Equity

Adeel Bhutta, Jan Vitek, and Mitch Wand
Khoury College of Computer Sciences

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

Quote: “Software Engineering at Google: Lessons Learned from Programming Over Time,” Wright, Winters and Manshreck, 2020 (O’Reilly)

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

- One mark of an exceptional engineer is the ability to understand how products can be weaponized to create harms in certain groups.

- Microsoft failed to do this with their chatbot of Tay that learned and picked up the behavior people used.

- People taught Tay to use offensive and racist language attacking jews.

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

- One mark of an exceptional engineer is the ability to understand how products can create harms in certain groups.
- Amazon failed to do this with their AI hiring software that used 10 years worth of resumes that had been submitted to Amazon to learn what candidates should be hired.
- Amazon taught its system to automatically reject the resumes of women.

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%

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 can discriminate against poorer consumers

THE WALL STREET JOURNAL.

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

https://www.wsj.com/articles/SB10001424127887323777204578189391813881534

2017 IEEE European Symposium on Security and Privacy

FairTest: Discovering Unwarranted Associations in Data-Driven Applications*

Florian Tramèr[1], Vaggelis Atlidakis[2], Roxana Geambasu[2], Daniel Hsu[2], Jean-Pierre Hubaux[3], Mathias Humbert[4], Ari Juels[5], Huang Lin[3]

[1]*Stanford,* [2]*Columbia University,* [3]*EPFL,* [4]*Saarland University,* [5]*Cornell 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

Work done while the first author was at EPFL.

DOI 10.1109/EuroSP.2017.29

IEEE computer society

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 drivi satellite and back

Get ready for Energy Star stickers on your

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

Training OpenAI's giant GPT-3 text-generating m car to the Moon and back, computer scientists re

More specifically, they estimated teaching the ne Microsoft data center using Nvidia GPUs require which using the average carbon intensity of America would have produced 85,000 kg of CO_2 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_2e (lbs)
Air travel, 1 passenger, NY↔SF	1984
, 1 year	11,023
vg, 1 year	36,156
el, 1 lifetime	126,000
odel (GPU)	
arsing, SRL)	39
xperimentation	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

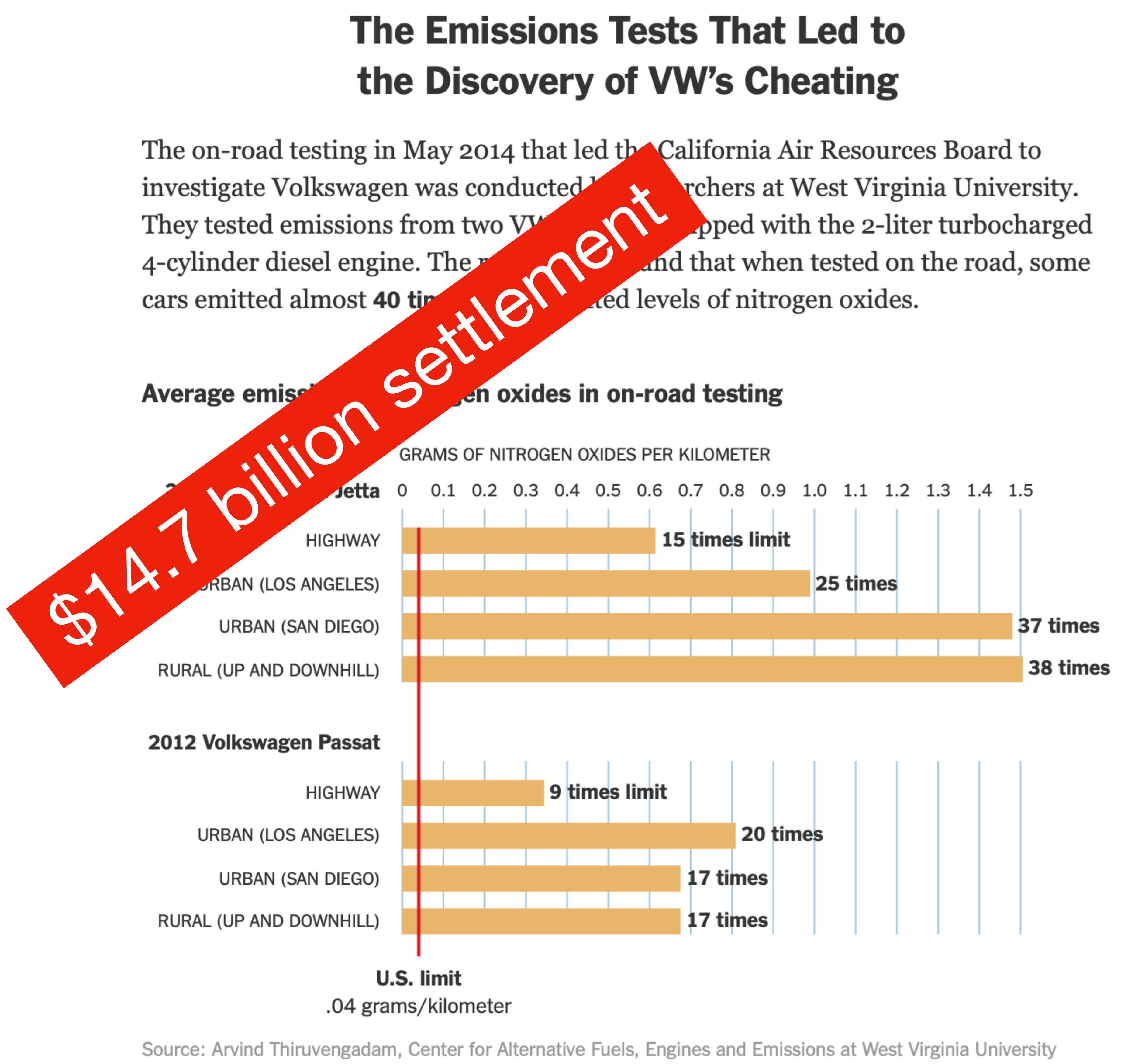
SHARE

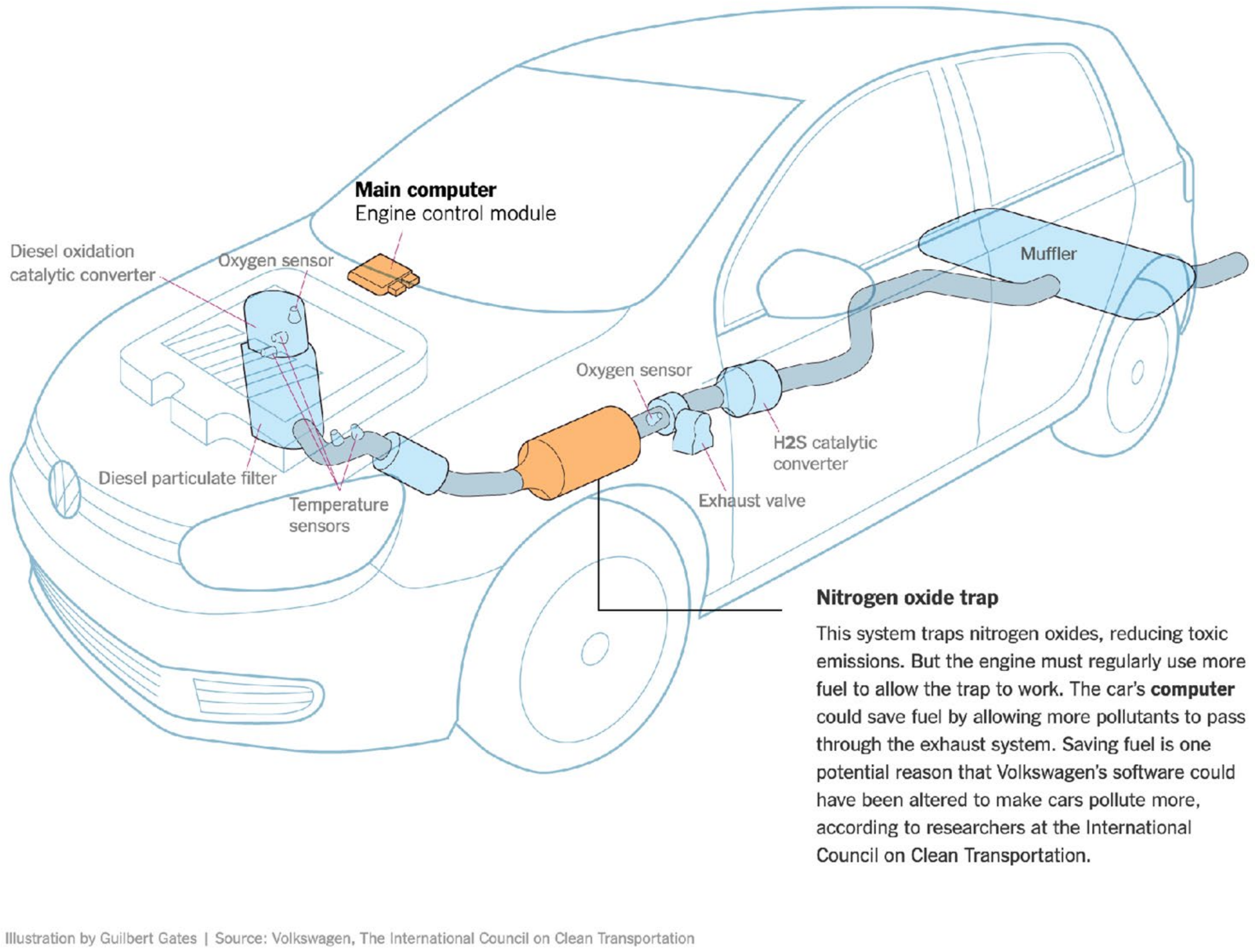
Jul 15 2019	**Brief amicus curiae of Washington Legal Foundation filed.**
Jul 15 2019	**Brief amici curiae of Retail Litigation Center, Inc., et al. filed.**
Jul 15 2019	**Brief amicus curiae of Cato Institute filed.**
Jul 15 2019	**Brief amicus curiae of Restaurant Law Center filed.**
Jul 15 2019	**Brief 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





"How Volkswagen's 'Defeat Devices' Worked" By Guilbert Gates, Jack Ewing, Karl Russell and Derek Watkins

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

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.

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

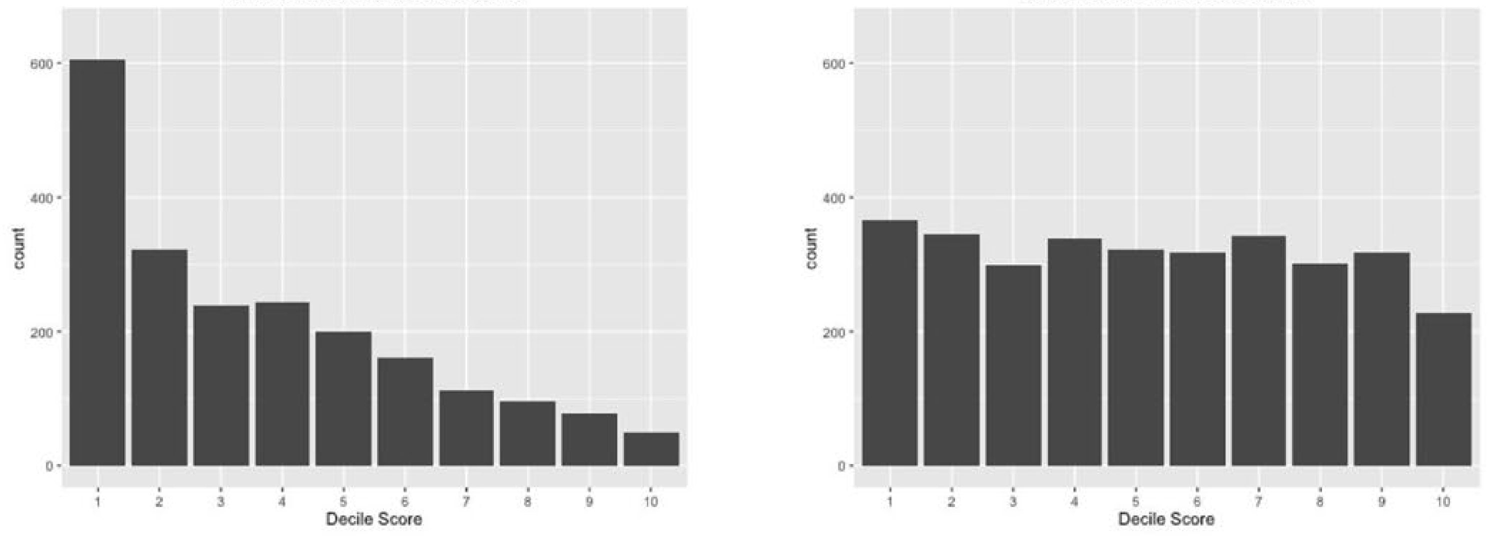
https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/

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


2017 IEEE European Symposium on Security and Privacy

FairTest: Discovering Unwarranted Associations in Data-Driven Applications

https://www.wsj.com/articles/SB10001424127887323777204578189391813881534

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

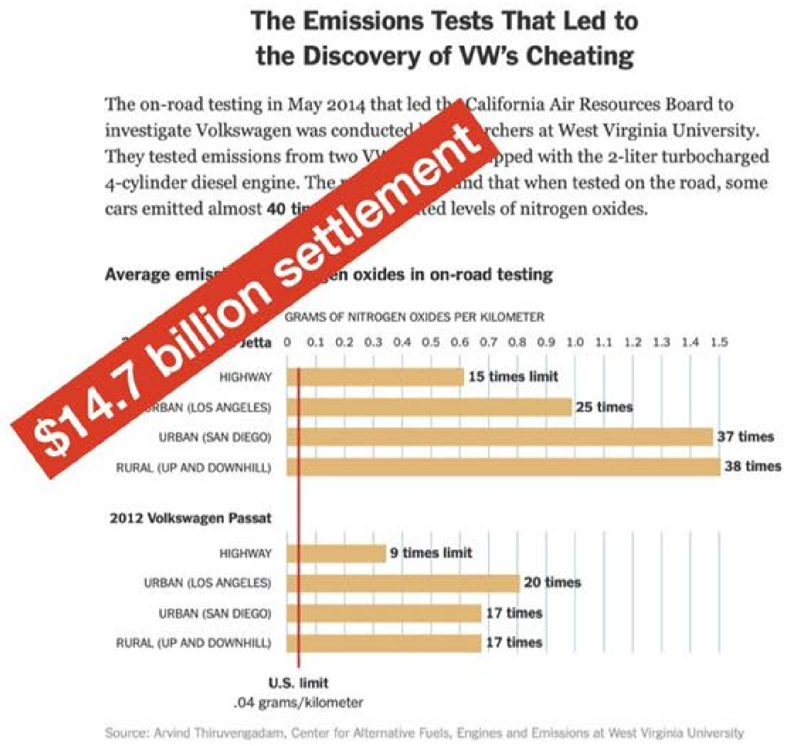
Rather than developing technology to support users with disabilities, the pizza chain is taking its fight to the top

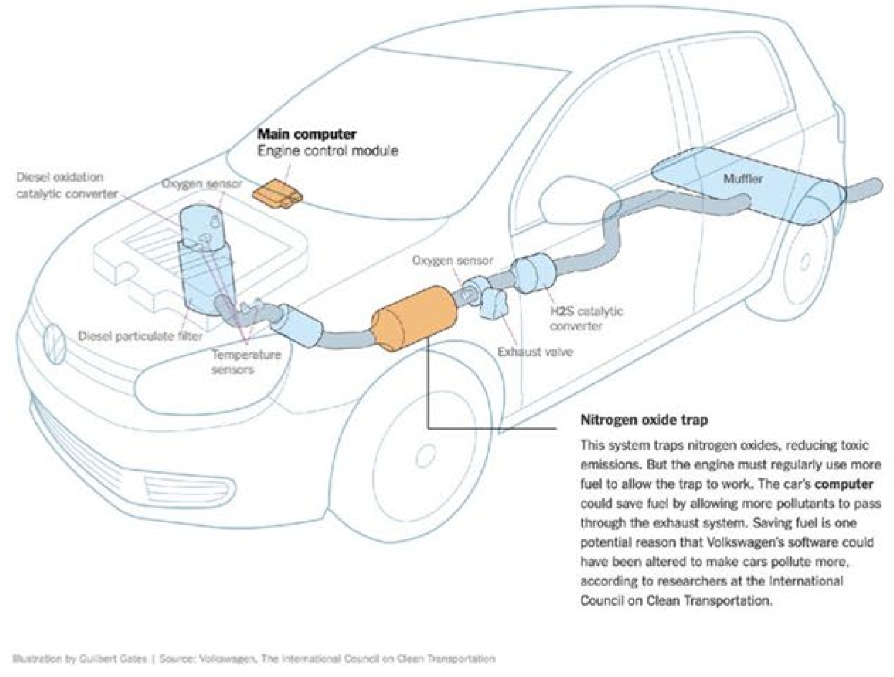
by Brenna Houck | @EaterDetroit | Jul 25, 2019, 6:00pm EDT

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

Evading regulation: Volkswagen





"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?

How might we mitigate harms in Software?

Everything can and should be iterated on, including the problem itself … what are you trying to solve?

- For every piece of software you create, you should iterate on it and include a wide range of people to use your software.
- By including more people you can better detect biases and harm that your software might create on certain populations.
- You want to iterate your software throughout its entire life cycle.

Guidelines from Microsoft on how to create software for people that mitigates harm.

Microsoft

Link: https://www.microsoft.com/en-us/research/uploads/prod/2020/03/Guidelines_summary_image@2x.png

1

INITIALLY

Make clear what the system can do

Help the users understand what the AI system is capable of doing.

2

INITIALLY

Make clear how well the system can do what it can do.

Help the user understand how often the AI system may make mistakes.

INITIALLY

Microsoft

3

DURING INTERACTION

Time services based on context.

Time when to act or interrupt based on the user's current task and environment.

4

DURING INTERACTION

Show contextually relevant information.

Display information relevant to the users' current task and environment.

5

DURING INTERACTION

Match relevant social norms.

Ensure the experience is delivered in a way that users would expect, given their social and cultural context.

6

DURING INTERACTION

Mitigate social biases.

Ensure the AI system's language and behaviors do not reinforce undesirable and unfair stereotypes and biases.

DURING INTERACTION

Microsoft

7

WHEN WRONG

Support efficient invocation.

Make it easy to invoke or request the AI system's services when needed.

8

WHEN WRONG

Support efficient dismissal.

Make it easy to dismiss or ignore undesired system services.

9

WHEN WRONG

Support efficient correction.

Make it easy to edit, refine, or recover when the AI system is wrong.

10

WHEN WRONG

Scope services when in doubt.

Engage in disambiguation or gracefully degrade the AI system's services when uncertain about a user's goals.

11

WHEN WRONG

Make clear why the system did what it did.

Enable the user to access an explanation of why the AI system behaved as it did.

⚠ WHEN WRONG

Microsoft

12
OVER TIME

Remember recent interactions.

Maintain short-term memory and allow the user to make efficient references to that memory.

13
OVER TIME

Learn from user behavior.

Personalize the user's experience by learning from their actions over time.

14
OVER TIME

Update and adapt cautiously.

Limit disruptive changes when updating and adapting the AI system's behaviors.

15
OVER TIME

Encourage granular feedback.

Enable the user to provide feedback indicating their preferences during regular interaction with the AI system.

16
OVER TIME

Convey the consequences of user actions.

Immediately update or convey how user actions will impact future behaviors of the AI system.

17
OVER TIME

Provide global controls.

Allow the user to globally customize what the AI system monitors and how it behaves.

18
OVER TIME

Notify users about changes.

Inform the user when the AI system adds or updates its capabilities.

OVER TIME

Microsoft

Class Exercise

- For Amazon’s Hiring Software, define with a partner how you would re -design the system using these guidelines to mitigate harm from the software.

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.