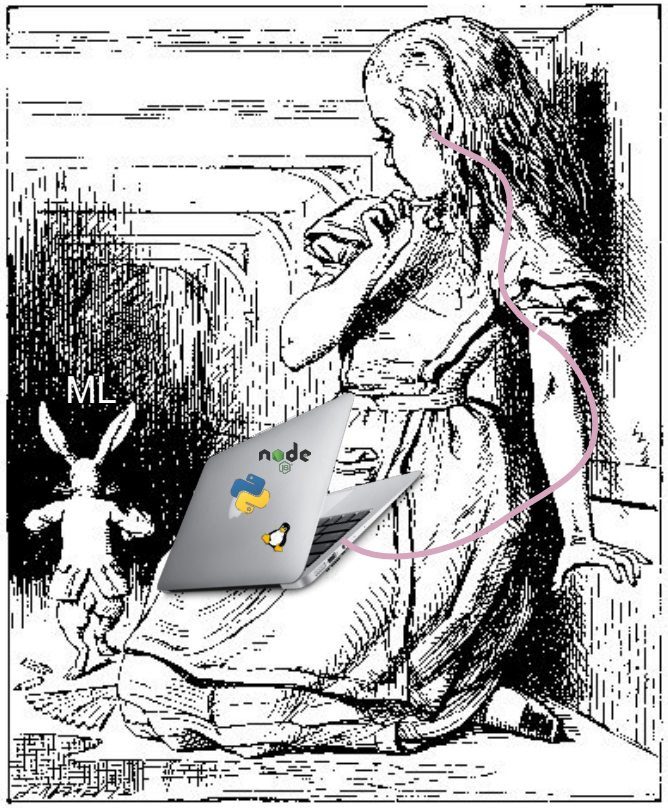
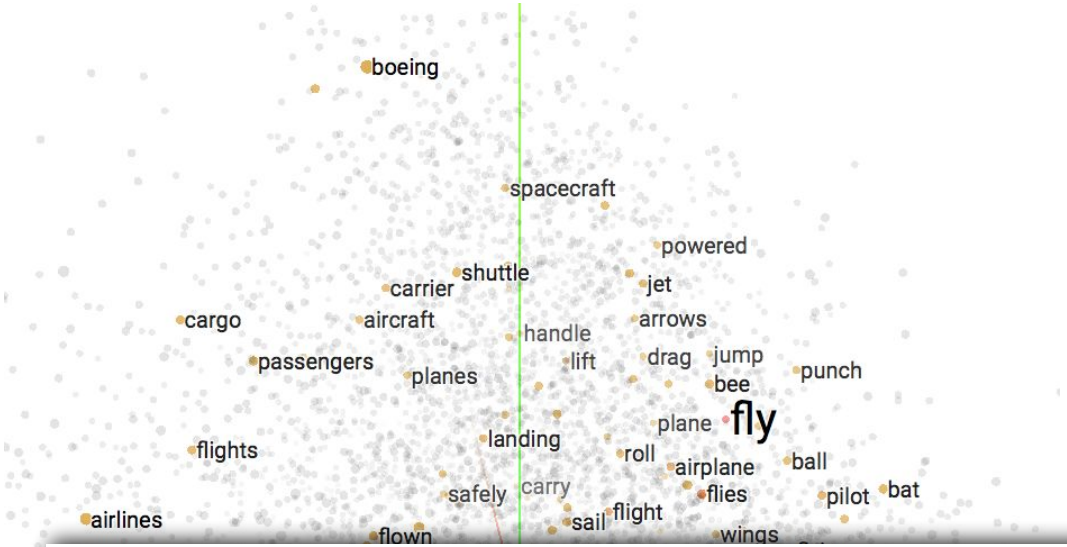
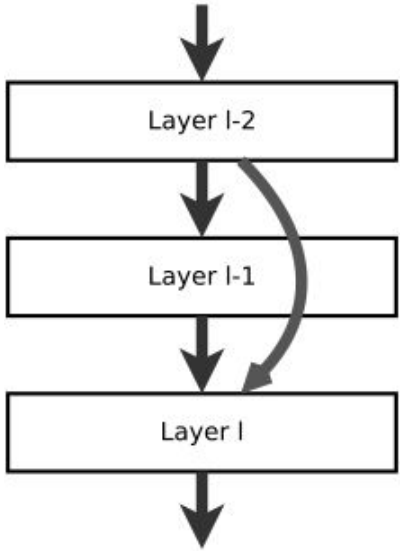

ML
node





TensorFlow

TensorFlow:
Large-Scale Machine Learning on Heterogeneous Distributed Systems
(Preliminary White Paper, November 9, 2015)

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng
Google Research*

Abstract

TensorFlow [1] is an interface for expressing machine learn

sequence prediction [47], move selection for Go [34], pedestrian detection [2], reinforcement learning [38],

Keras

Andrej Karpathy blog

About Hacker's guide to Neural Networks

The Unreasonable Effectiveness of Recurrent Neural Networks

May 21, 2015

There's something magical about Recurrent Neural Networks (RNNs). I still remember when I trained my first recurrent network for Image Captioning. Within a few dozen minutes of training my first baby model (with rather arbitrarily-chosen hyperparameters) started to generate very nice looking descriptions of images that were on the edge of making sense. Sometimes the ratio of how simple your model is to the quality of the results you get out of it blows past your expectations, and this was one of those times. What made this result so shocking at the time was that the common wisdom was that RNNs were supposed to be difficult to train (with more experience I've in fact reached the opposite conclusion). Fast forward about a year: I'm training RNNs all the time and I've witnessed their power and robustness many times, and yet their magical outputs still find ways of amusing me. This post is about sharing some of that magic with you.

> *We'll train RNNs to generate text character by character and ponder the question "how is that even possible?"*

By the way, together with this post I am also releasing code on Github that allows you to train character-level language models based on multi-layer LSTMs. You give it a large chunk of text and it will learn to generate text like it one character at a time. You can also use it to reproduce my experiments below. But we're getting ahead o ourselves; What are RNNs anyway?

Build the LSTM model

```
# build the model: a single LSTM
print('Build model...')
model = Sequential()
model.add(LSTM(128, input_shape=(maxlen, len(chars))))
model.add(Dense(len(chars)))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')

print("Compiling model complete...")
```

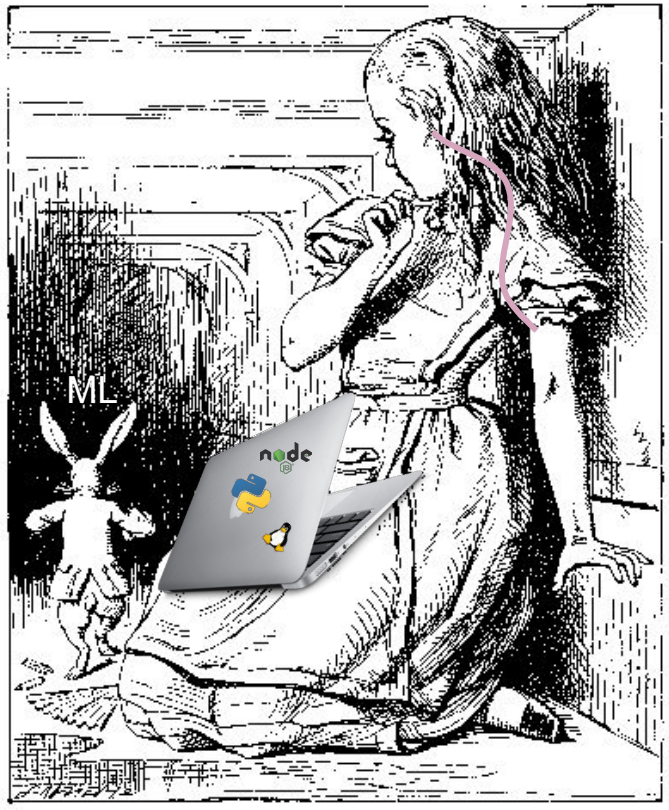
```
Build model...
Compiling model complete...
```

Machine Learning

- Grew out of work in AI
- New capability for computers

Andrew Ng

Alice was excited!



Lots of tutorials

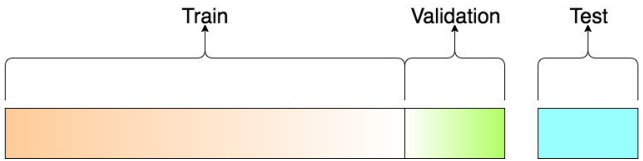
Loads of resources

Endless examples

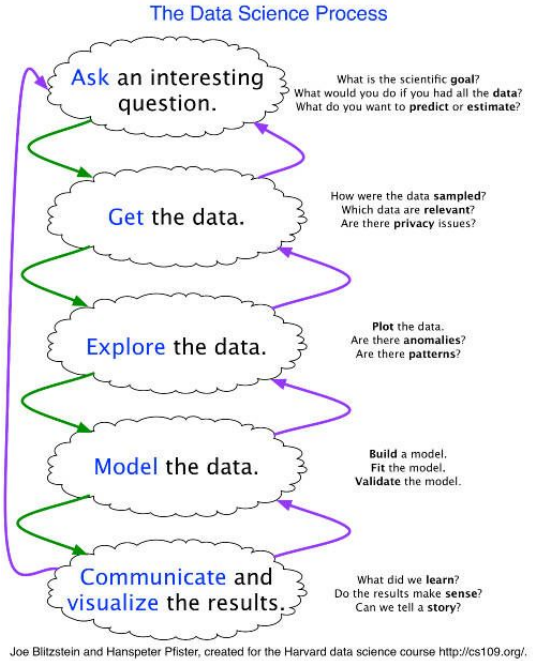
Fast paced research

How to even data science?

How to even data science?



https://miro.medium.com/max/1552/1*Nv2NNALuokZEcV6hYEHdGA.png



Challenge

How to make this work in the real world?

Machine Learning's Surprises

A Checklist for Developers
when Building ML Systems

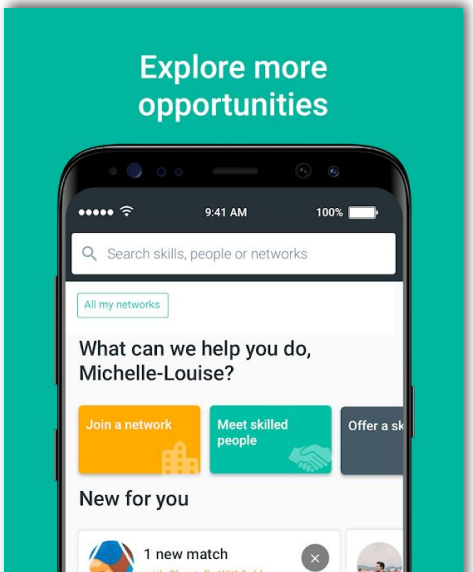
Hi, I'm Jade Abbott

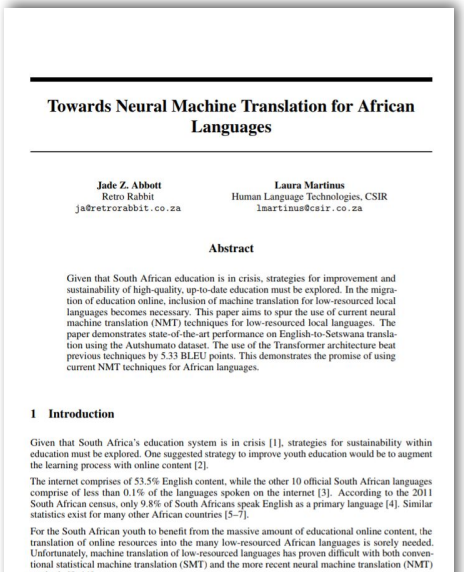
@alienelf

RETRO RABBIT

KALIDO

masakhane.io





Towards Neural Machine Translation for African Languages

Jade Z. Abbott
Retro Rabbit
ja@retrorabbit.co.za

Laura Martinus
Human Language Technologies, CSIR
lmartinus@csir.co.za

Abstract

Given that South African education is in crisis, strategies for improvement and sustainability of high-quality, up-to-date education must be explored. In the migration of education online, inclusion of machine translation for low-resourced local languages becomes necessary. This paper aims to spur the use of current neural machine translation (NMT) techniques for low-resourced local languages. The paper demonstrates state-of-the-art performance on English-to-Setswana translation using the Autshumato dataset. The use of the Transformer architecture beat previous techniques by 5.33 BLEU points. This demonstrates the promise of using current NMT techniques for African languages.

1 Introduction

Given that South Africa's education system is in crisis [1], strategies for sustainability within education must be explored. One suggested strategy to improve youth education would be to augment the learning process with online content [2].

The internet comprises of 53.5% English content, while the other 10 official South African languages comprise of less than 0.1% of the languages spoken on the internet [3]. According to the 2011 South African census, only 9.8% of South Africans speak English as a primary language [4]. Similar statistics exist for many other African countries [5–7].

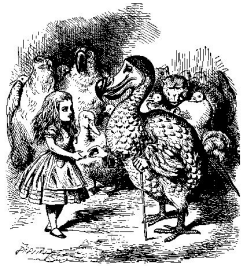
For the South African youth to benefit from the massive amount of educational online content, the translation of online resources into the many low-resourced African languages is sorely needed. Unfortunately, machine translation of low-resourced languages has proven difficult with both conventional statistical machine translation (SMT) and the more recent neural machine translation (NMT)

Hi, I'm Jade Abbott

Surprises while...

Trying to deploy the model

Trying to improve the model



After deployment of model

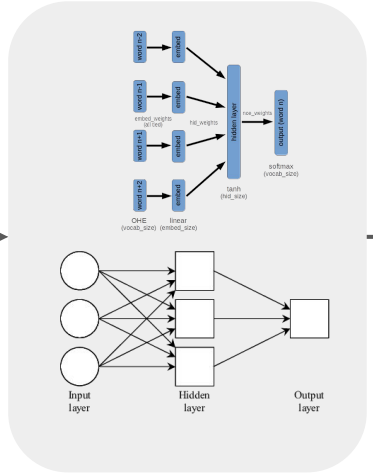
Some context

- I won’t be talking about training machine learning models
- I won’t be talking about which models to chose
- I work primarily in deep learning & NLP
- I am a one person ML team working in a startup context
- I work in a normal world where data is scarce and we need to collect more

The Problem

I want to meet...

I can provide...



Yes, they should meet

No they shouldn't

Embedding + LSTM + Downstream NN

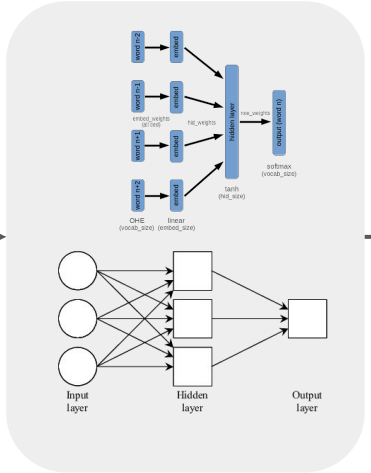
The Problem

I want to meet...

someone to look after my cat

I can provide...

pet sitting
cat breeding
software
development
chef lessons



Yes, they should meet

No they shouldn't

Language Model + Downstream Task

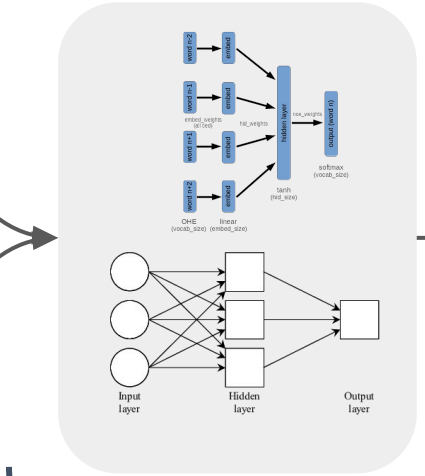
The Problem

I want to meet...

someone to look after my cat

I can provide...

pet sitting
cat breeding
software
development
chef lessons



Yes, they should meet

No they shouldn't

The Problem

I want to meet...

someone to look after my cat

I can provide...

pet sitting
cat breeding
software development
chef lessons

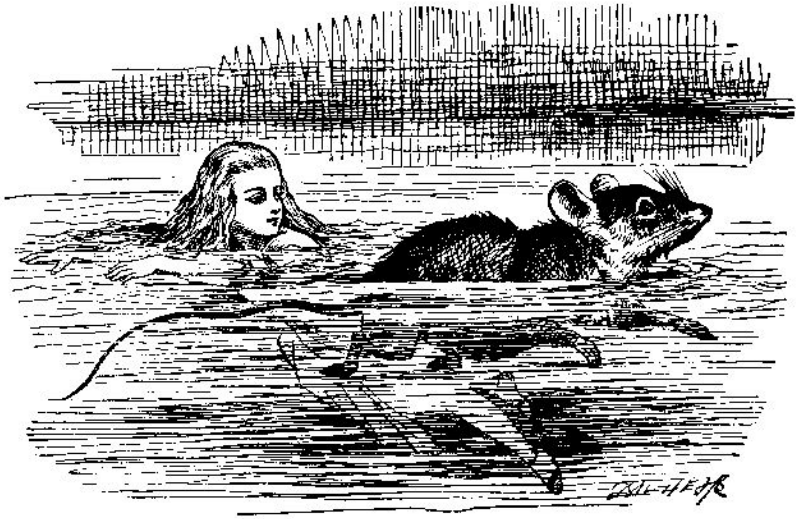
The Model

Yes, they should meet

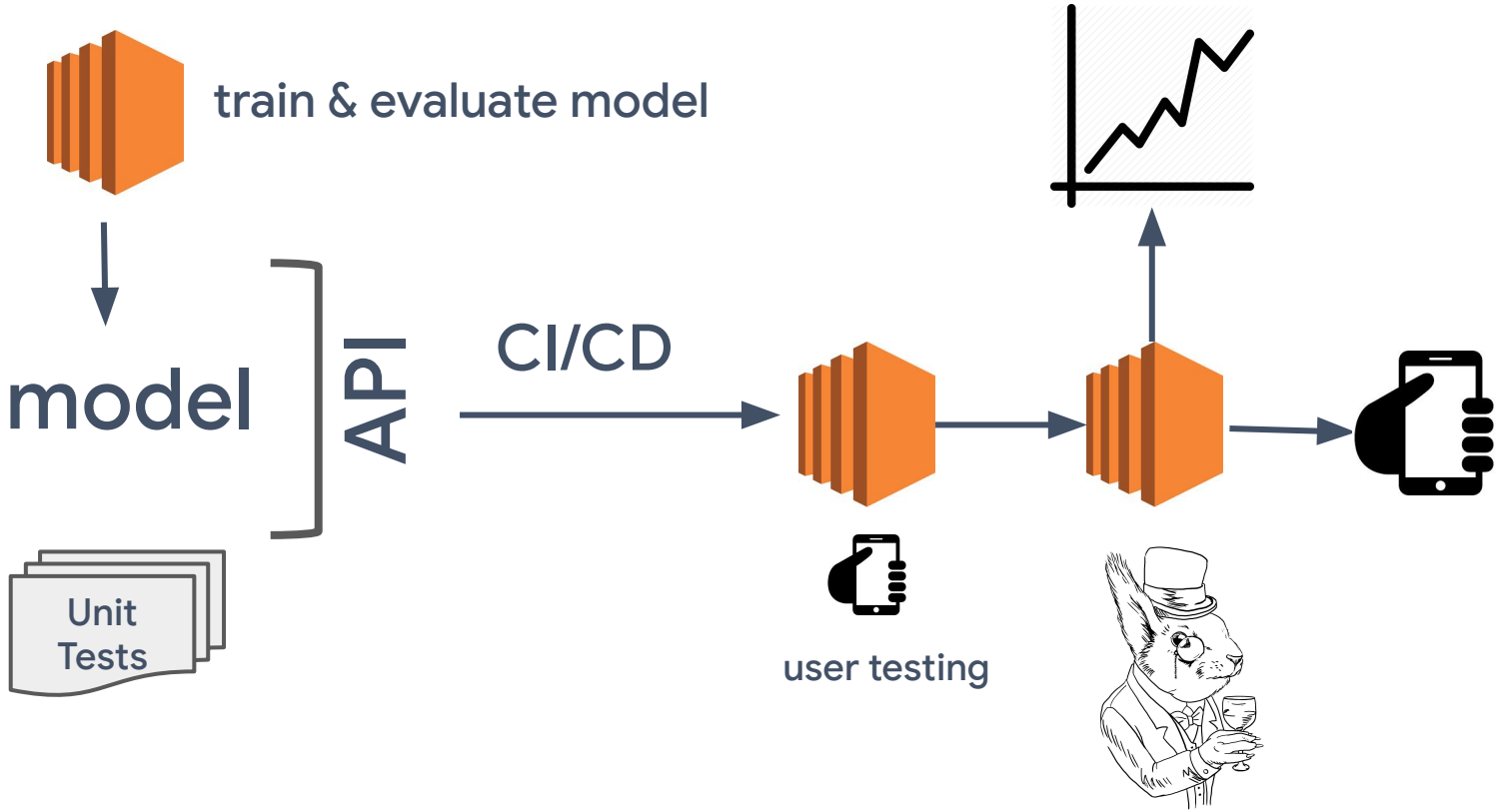
No they shouldn't

Surprises

Surprises trying to deploy the model



Expectations



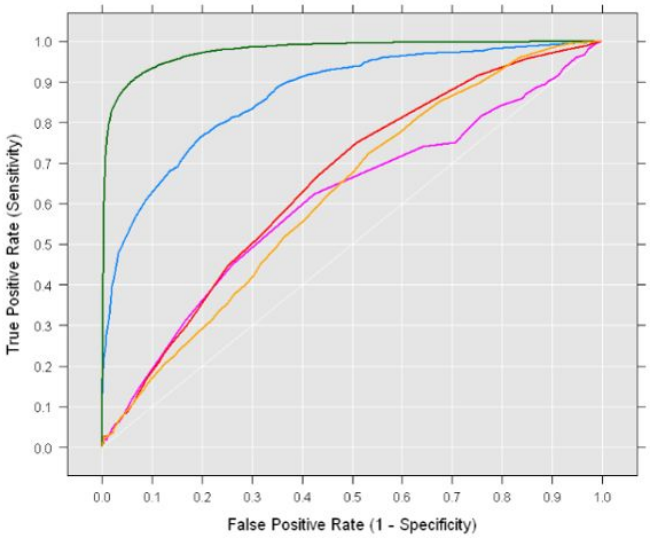
Surprise #1

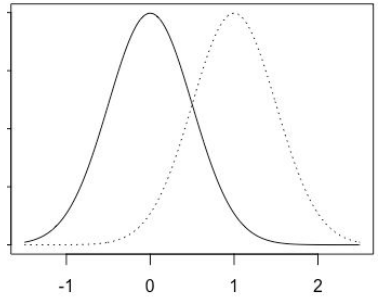
Is the model good enough?

75% Accuracy

Performance Metrics

- Business needs to understand it
- Active discussion about pros & cons
- Get sign off
- Threshold selection strategy



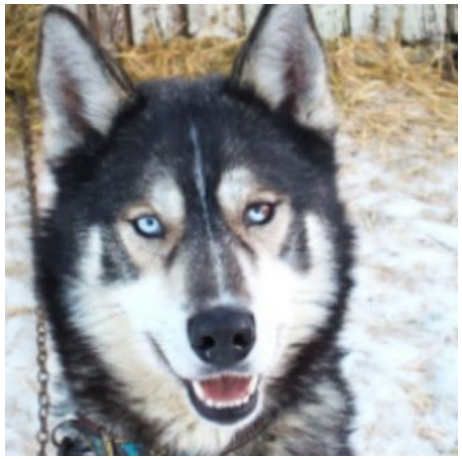


Surprise #2

Can we trust it?

Skin Cancer Detection

Husky/Dog Classifier



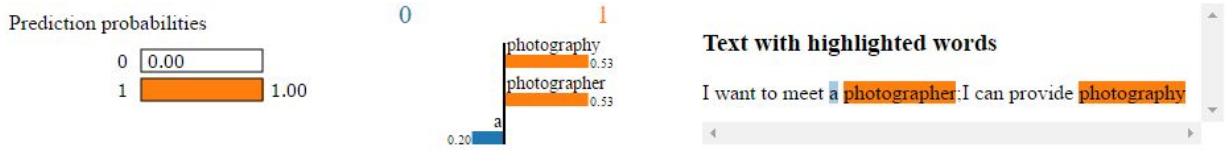
1. https://visualsonline.cancer.gov/details.cfm?imageid=9288
2. https://arxiv.org/pdf/1602.04938.pdf

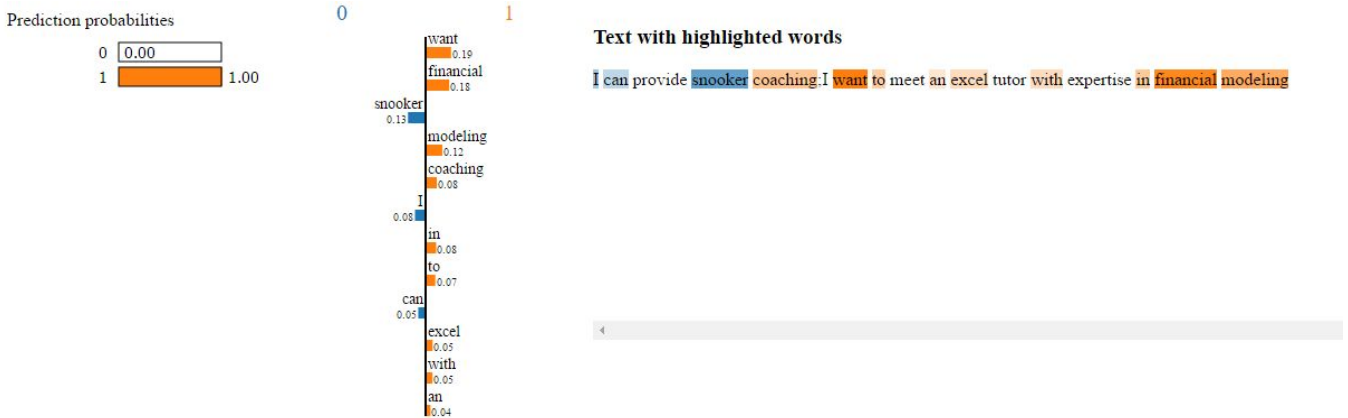
Skin Cancer Detection

Husky/Dog Classifier

1. https://visualsonline.cancer.gov/details.cfm?imageid=9288
2. https://arxiv.org/pdf/1602.04938.pdf

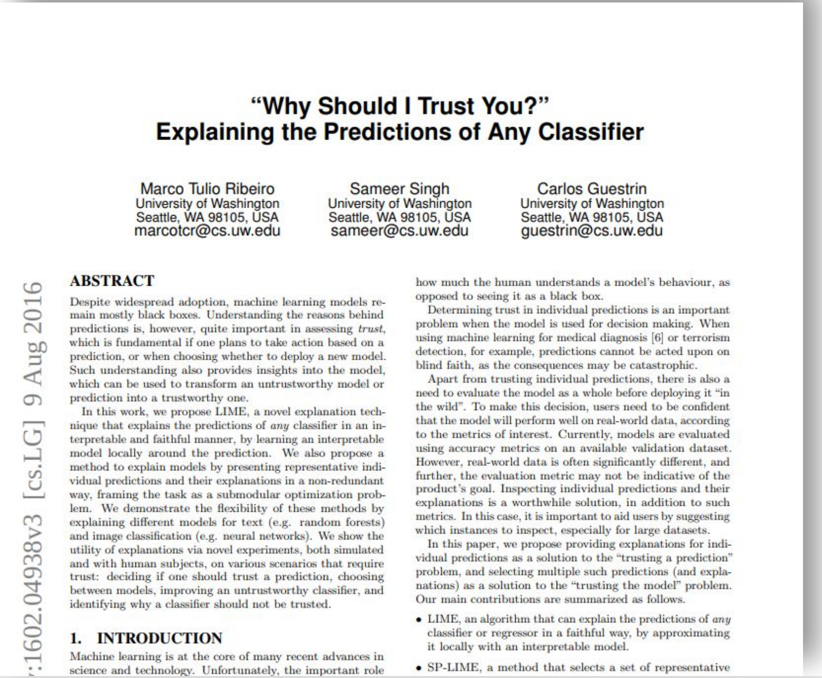
Explanations





https://github.com/marcotcr/lime

https://pair-code.github.io/what-if-tool/



"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

:1602.04938v3 [cs.LG] 9 Aug 2016

ABSTRACT

Despite widespread adoption, machine learning models remain mostly black boxes. Understanding the reasons behind predictions is, however, quite important in assessing *trust*, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model. Such understanding also provides insights into the model, which can be used to transform an untrustworthy model or prediction into a trustworthy one.

In this work, we propose LIME, a novel explanation technique that explains the predictions of *any* classifier in an interpretable and faithful manner, by learning an interpretable model locally around the prediction. We also propose a method to explain models by presenting representative individual predictions and their explanations in a non-redundant way, framing the task as a submodular optimization problem. We demonstrate the flexibility of these methods by explaining different models for text (e.g. random forests) and image classification (e.g. neural networks). We show the utility of explanations via novel experiments, both simulated and with human subjects, on various scenarios that require trust: deciding if one should trust a prediction, choosing between models, improving an untrustworthy classifier, and identifying why a classifier should not be trusted.

1. INTRODUCTION

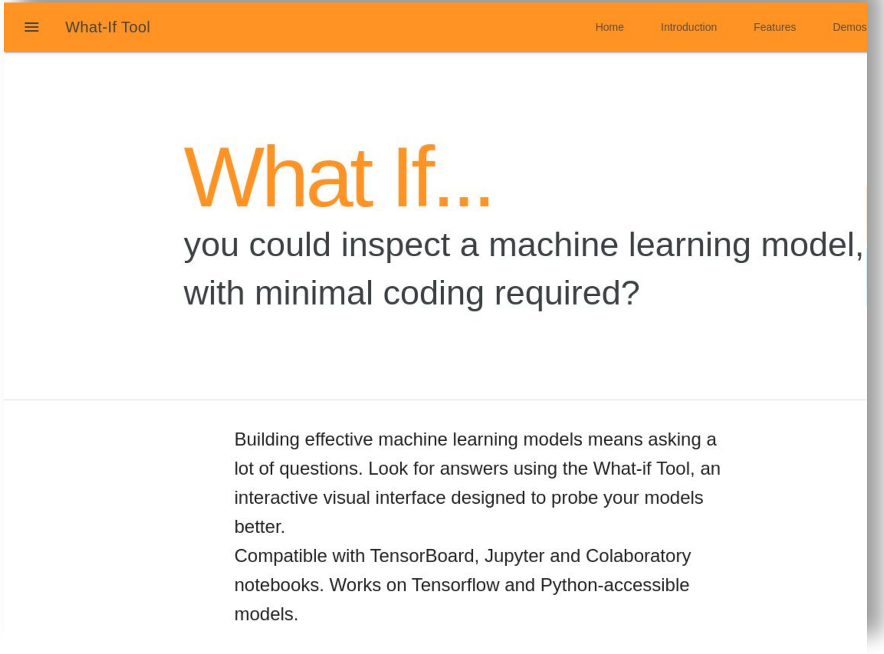
Machine learning is at the core of many recent advances in science and technology. Unfortunately, the important role how much the human understands a model's behaviour, as opposed to seeing it as a black box.

Determining trust in individual predictions is an important problem when the model is used for decision making. When using machine learning for medical diagnosis [6] or terrorism detection, for example, predictions cannot be acted upon on blind faith, as the consequences may be catastrophic.

Apart from trusting individual predictions, there is also a need to evaluate the model as a whole before deploying it "in the wild". To make this decision, users need to be confident that the model will perform well on real-world data, according to the metrics of interest. Currently, models are evaluated using accuracy metrics on an available validation dataset. However, real-world data is often significantly different, and further, the evaluation metric may not be indicative of the product's goal. Inspecting individual predictions and their explanations is a worthwhile solution, in addition to such metrics. In this case, it is important to aid users by suggesting which instances to inspect, especially for large datasets.

In this paper, we propose providing explanations for individual predictions as a solution to the "trusting a prediction" problem, and selecting multiple such predictions (and explanations) as a solution to the "trusting the model" problem. Our main contributions are summarized as follows.

- LIME, an algorithm that can explain the predictions of *any* classifier or regressor in a faithful way, by approximating it locally with an interpretable model.
- SP-LIME, a method that selects a set of representative



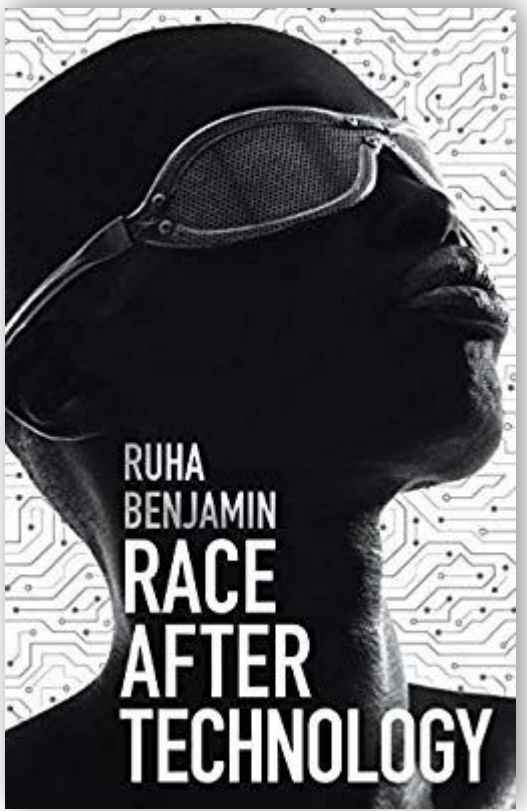
Surprise #3

Will this model harm users?


“Racial bias in a medical algorithm favors white patients over sicker black patients”
Washington Post

“Racist robots, as I invoke them here, represent a much broader process: social bias embedded in technical artifacts, the allure of objectivity without public accountability”

~ Ruha Benjamin @ruha9



“What are the unintended consequences of designing systems at scale on the basis of existing patterns of society?”

~ M.C. Eilish & Danah Boyd,
Don’t Believe Every AI You See
@m_c_elish @zephoria

- Word2Vec has known gender and race biases
- It’s in English
- Is it robust to spelling errors?
- How does it perform with malicious data?

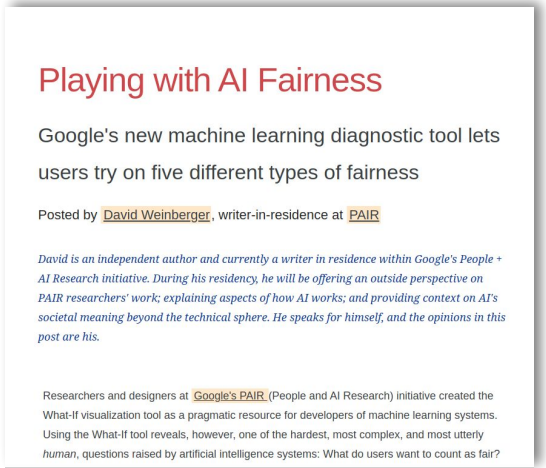
- Word2Vec has known gender and race biases
- It’s in English
- Is it robust to spelling errors?
- How does it perform with malicious data?

Make it measurable!

https://pair-code.github.io

http://aif360.mybluemix.net

https://github.com/fairlearn/fairlearn



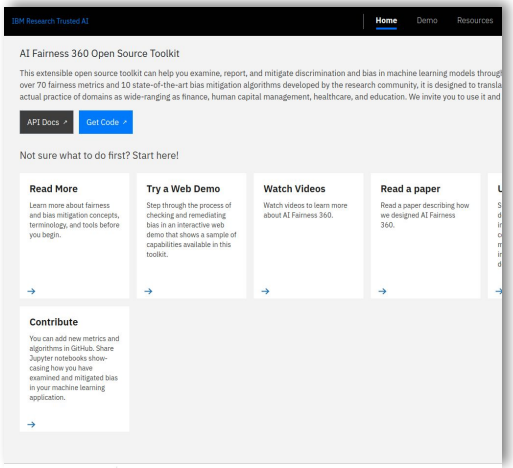
Playing with AI Fairness

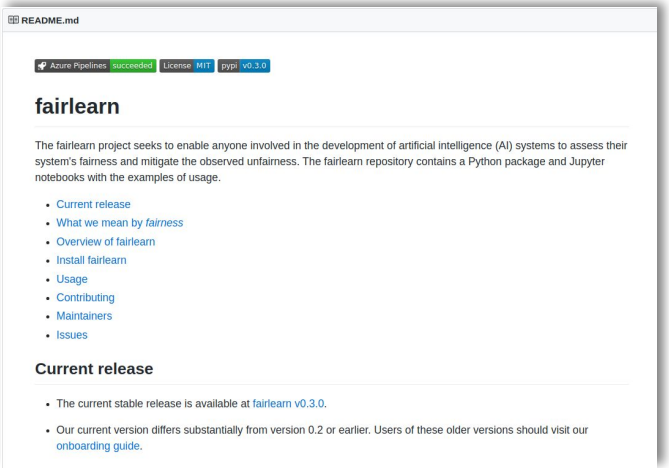
Google's new machine learning diagnostic tool lets users try on five different types of fairness

Posted by David Weinberger, writer-in-residence at PAIR

David is an independent author and currently a writer in residence within Google's People + AI Research initiative. During his residency, he will be offering an outside perspective on PAIR researchers' work; explaining aspects of how AI works; and providing context on AI's societal meaning beyond the technical sphere. He speaks for himself, and the opinions in this post are his.

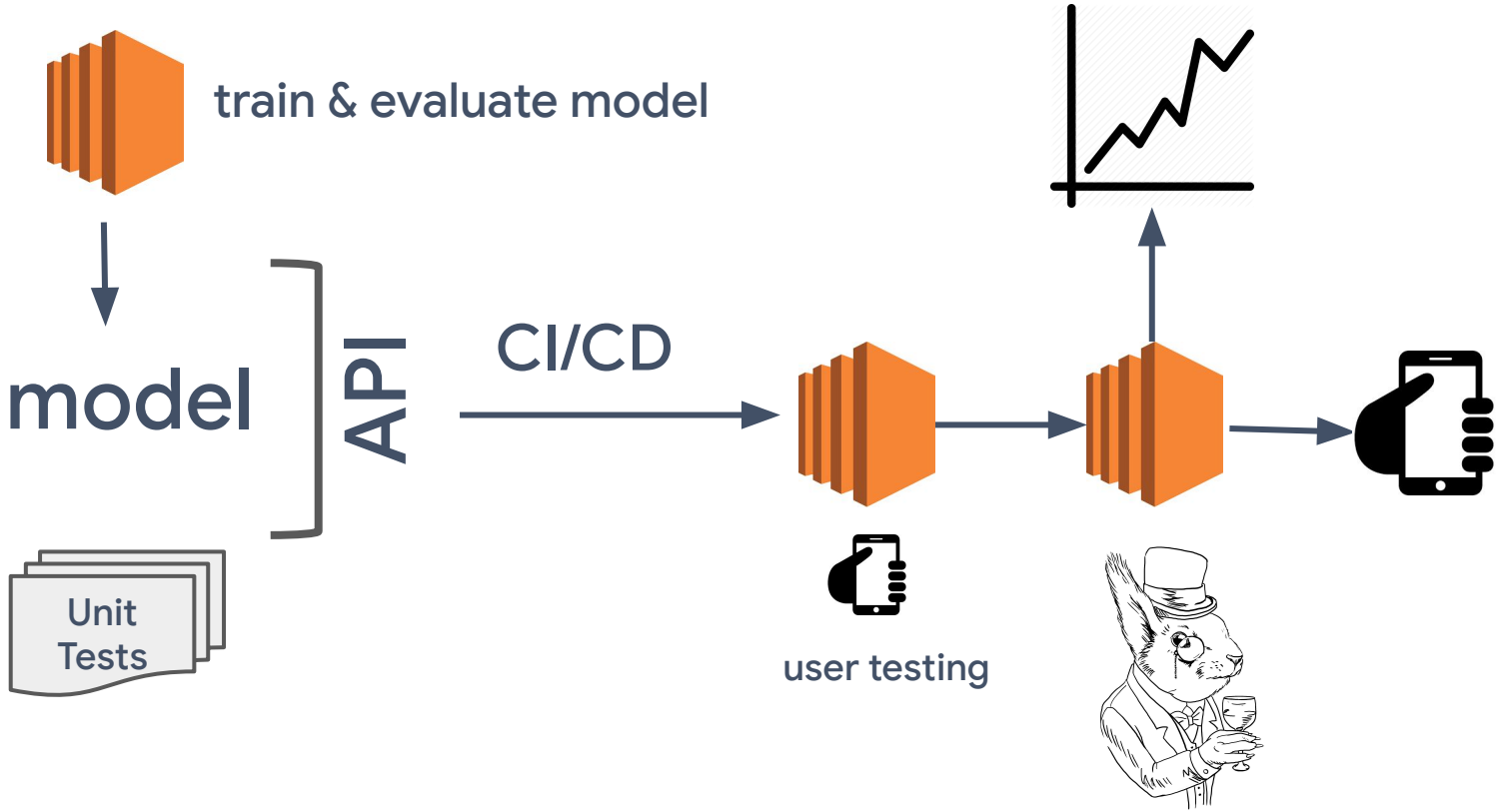
Researchers and designers at Google's PAIR (People and AI Research) initiative created the What-If visualization tool as a pragmatic resource for developers of machine learning systems. Using the What-If tool reveals, however, one of the hardest, most complex, and most utterly *human*, questions raised by artificial intelligence systems: What do users want to count as fair?





https://github.com/jphall663/awesome-machine-learning-interpretability

Expectations



Reality

choose
a useful metric

Evaluate model

Choose threshold

Explain predictions

Fairness Framework

model API

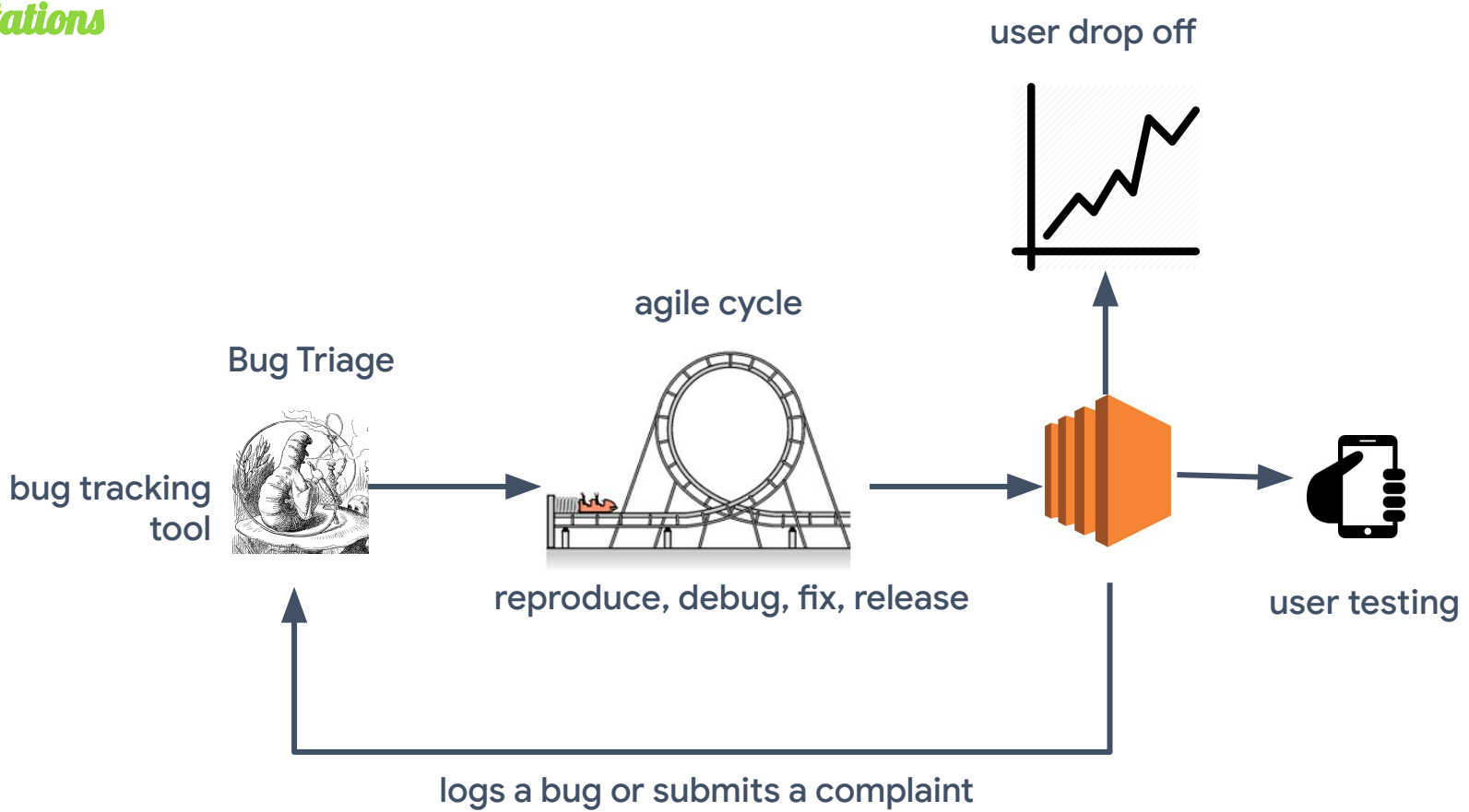
Unit Tests

user testing

Surprises

Surprises after deploying the model

Expectations



Surprise #5


I want to meet a doctor
I can provide marijuana and other drugs which improves health

Surprise #5

The model has some “bugs”

Surprise #5 continued...

- What is a model “bug”
- How to fix the bug?
- When is the “bug” fixed?
- How do I ensure test regression?
- “Bug” priority?

Surprise #5


I want to meet a doctor
I can provide marijuana and other drugs which improves health

Describing the "bugs"

Add to your test set

		Prediction	Target	
I can provide marijuana and other drugs which improves health	I want to meet a doctor	YES	NO	False Positive
I can provide marijuana	I want to meet a doctor	NO	NO	True Negative
I can provide drugs for cancer patients	I want to meet a doctor	YES	NO	
I can provide general practitioner services	I want to meet a doctor	NO	YES	False Negative
I can provide medicine	I want to meet a drug addiction sponsor	YES	YES	
I can provide medicine	I want to meet a pharmacist	YES	YES	True Positives
I can provide illegal drugs	I want to meet a drug dealer	YES	NO	

Match Problems

Showing 65 match problems

ID	Name	Description
1	same-day-problem	If tangibles have the same day but everything is different, they will match when they shouldnt
2	same-place-problem	If tangibles have the same place but everything is different, they will match when they shouldnt
3	for-me	Tangibles with for me in them get less matches
4	1-services	If 1 tangible has services, then it experiences a bit negative
5	2-services	If both tangibles has service in it, then it experiences a big positive
6	commas	If 1 tangible has a comma in it, then prediction has a big negative

View match problem

< Match problems

ID
1

Created at
2017-02-14

Name *
same-day-problem

Regular expression *
select * from tangibles where (description ~ 'Monday' or description ~ 'Tuesday' or description ~ 'Wednesday' or description ~ 'Thursday' or description ~ 'Friday' or description ~ 'Saturday' or

Description *
If tangibles have the same day but everything is different, they will match when they shouldnt

Test patterns

2 positive, 6 negative samples

Type *
Provide

Description *
I can provide iOS development

29 / 120

Type *
Meet

Description *
I want to meet mobile app dev

29 / 120

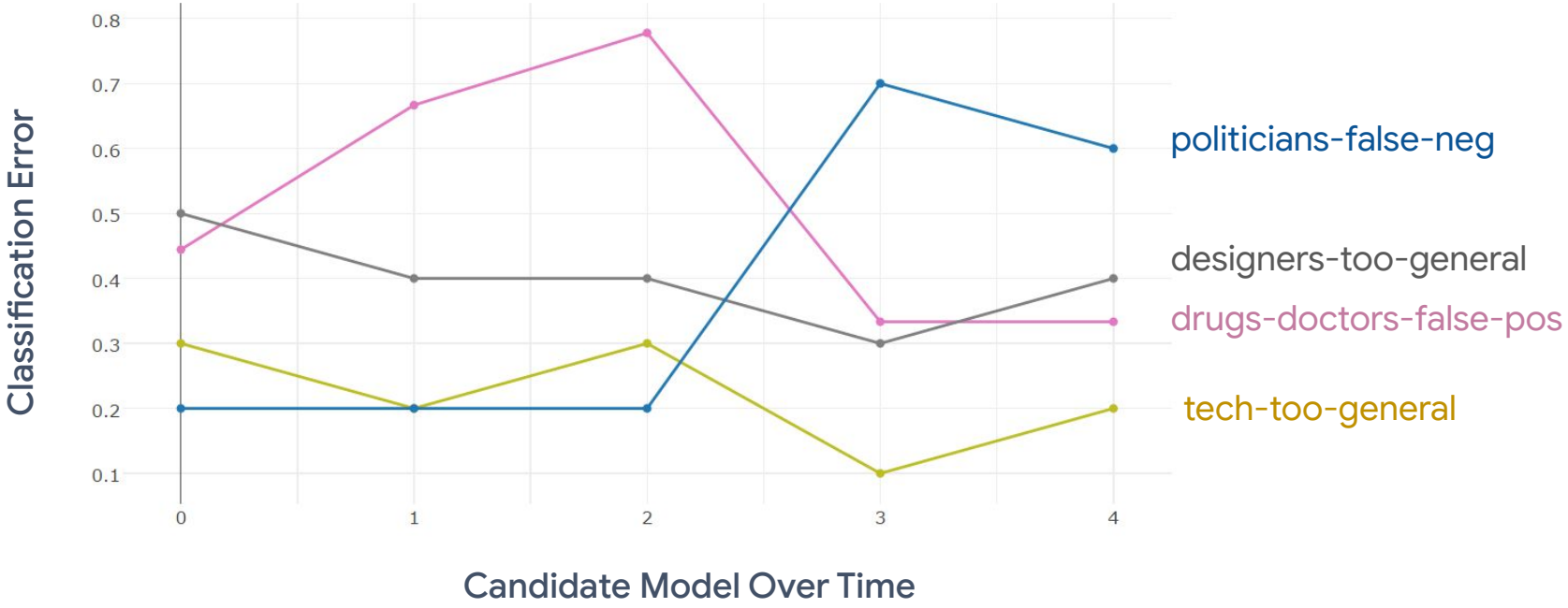
Should match

Type *
Provide

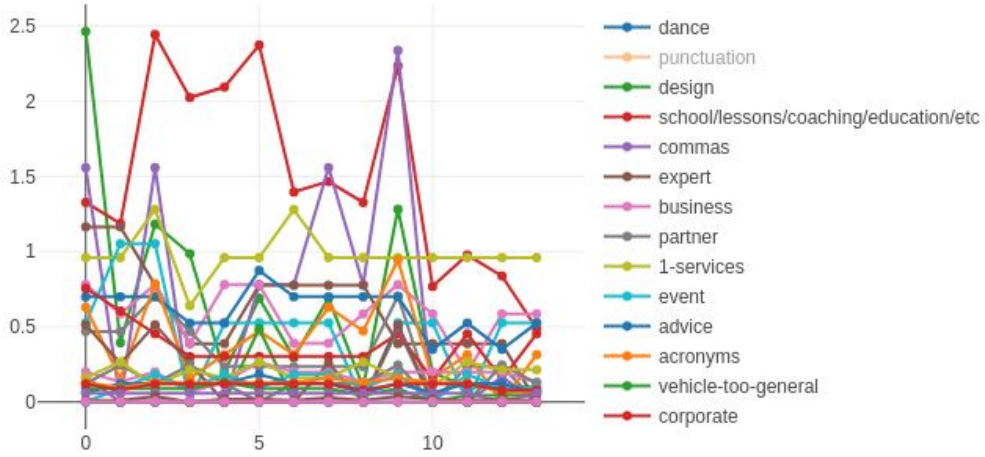
Description *
I can provide social media management and setup in Safety Harbor

64 / 120

Is my "bug" fixed?



How do we triage these “bugs”?



How do we triage these “bugs”?

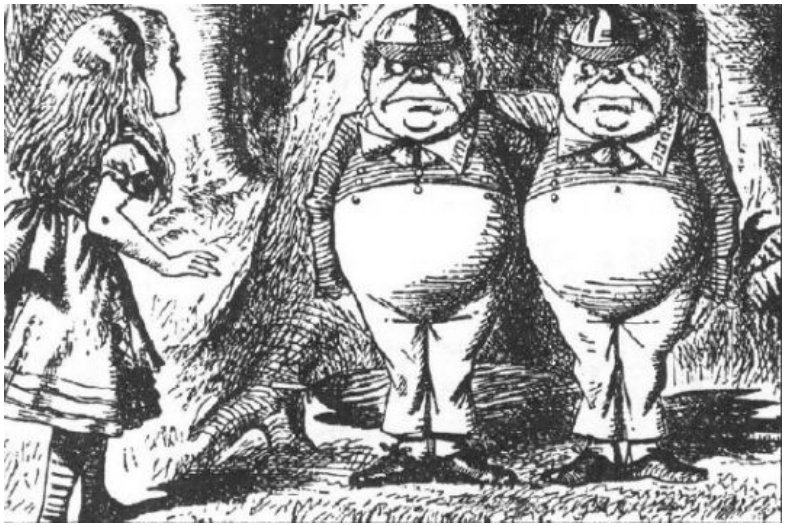
% Users Affected
x
Normalized Error
x
Harm

How do we triage these “bugs”?

Problem	Impact Error
the-arts-too-general	2.931529
health-more-specific	1.53985
brand-marketing-social-media	1.285735
developer	1.054248
1-services	0.960129

Surprise #6

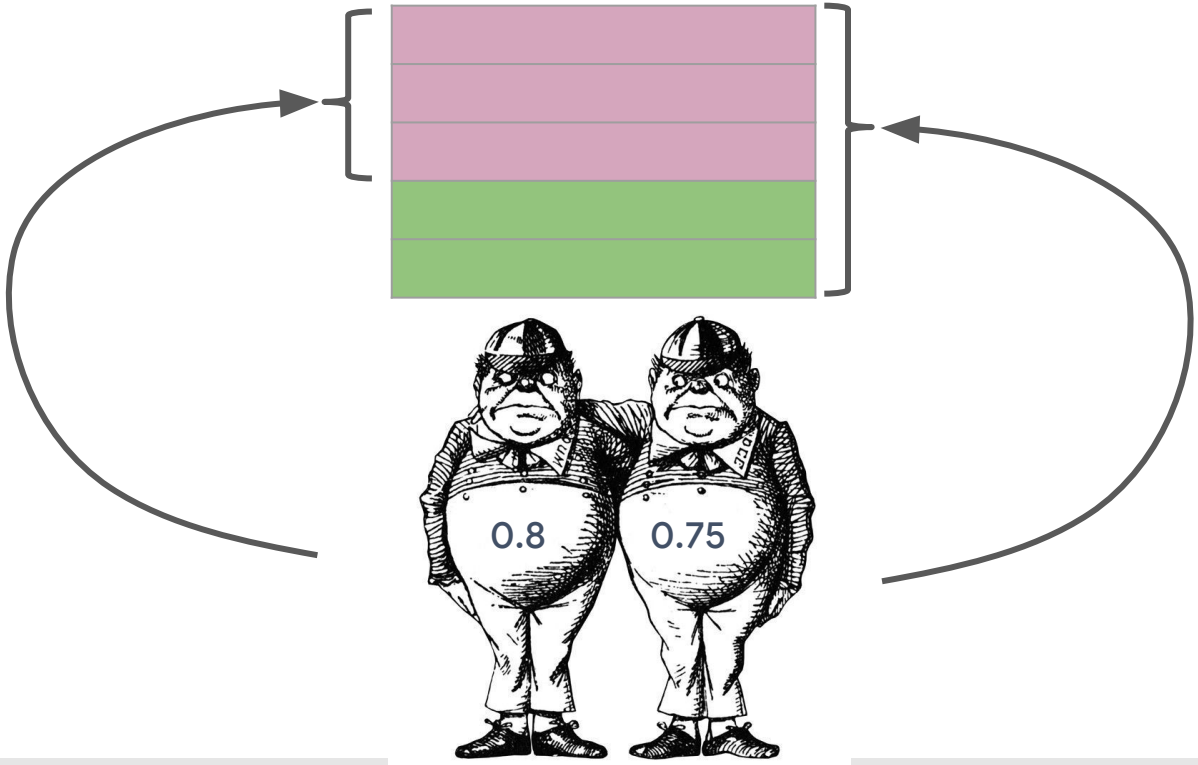
Is this new model better than my old model?



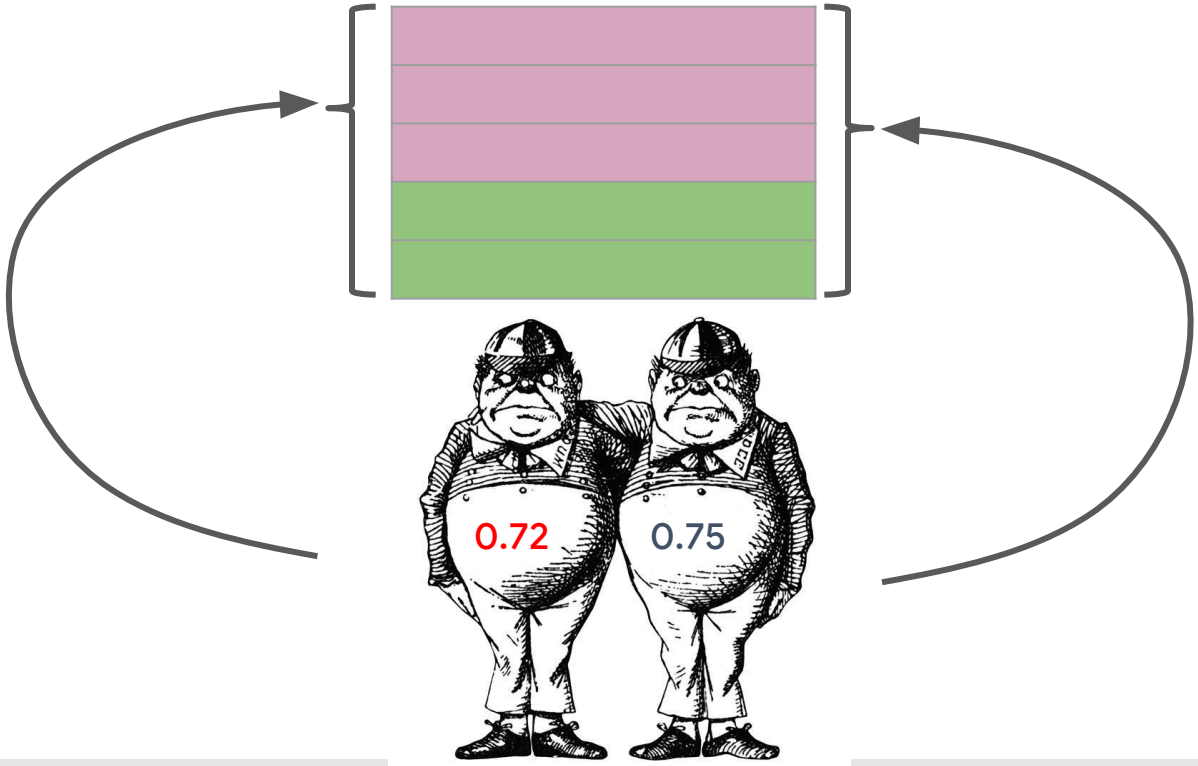
Alice replied, rather shyly, “I—I hardly know, sir, just at present—at least I know who I was when I got up this morning, but I think I must have changed several times since then.”

Why is model comparison hard?

Living Test Set



Re-evaluate ALL models



Surprise #7

I demoed the model yesterday and it went off-script!
What changed?

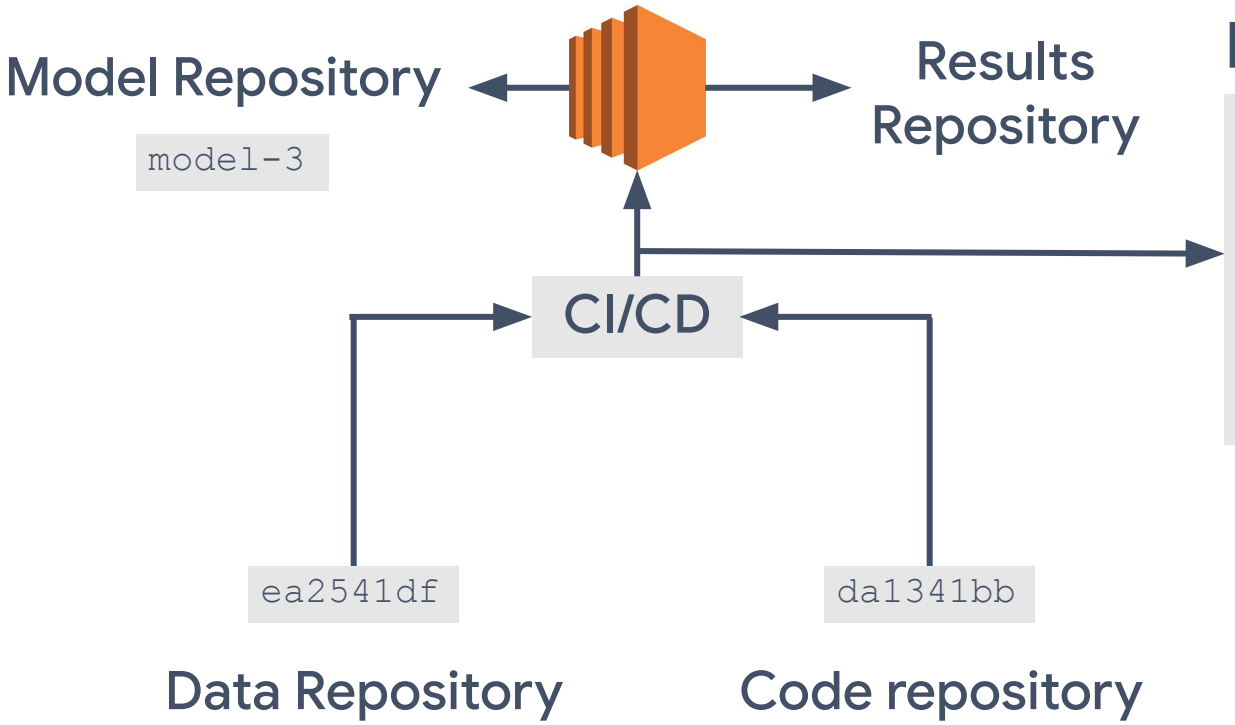
Surprise #7

Why is the model doing something differently today?

What changed?

- My data?
- My model?
- My preprocessing?

How to figure out what changed?

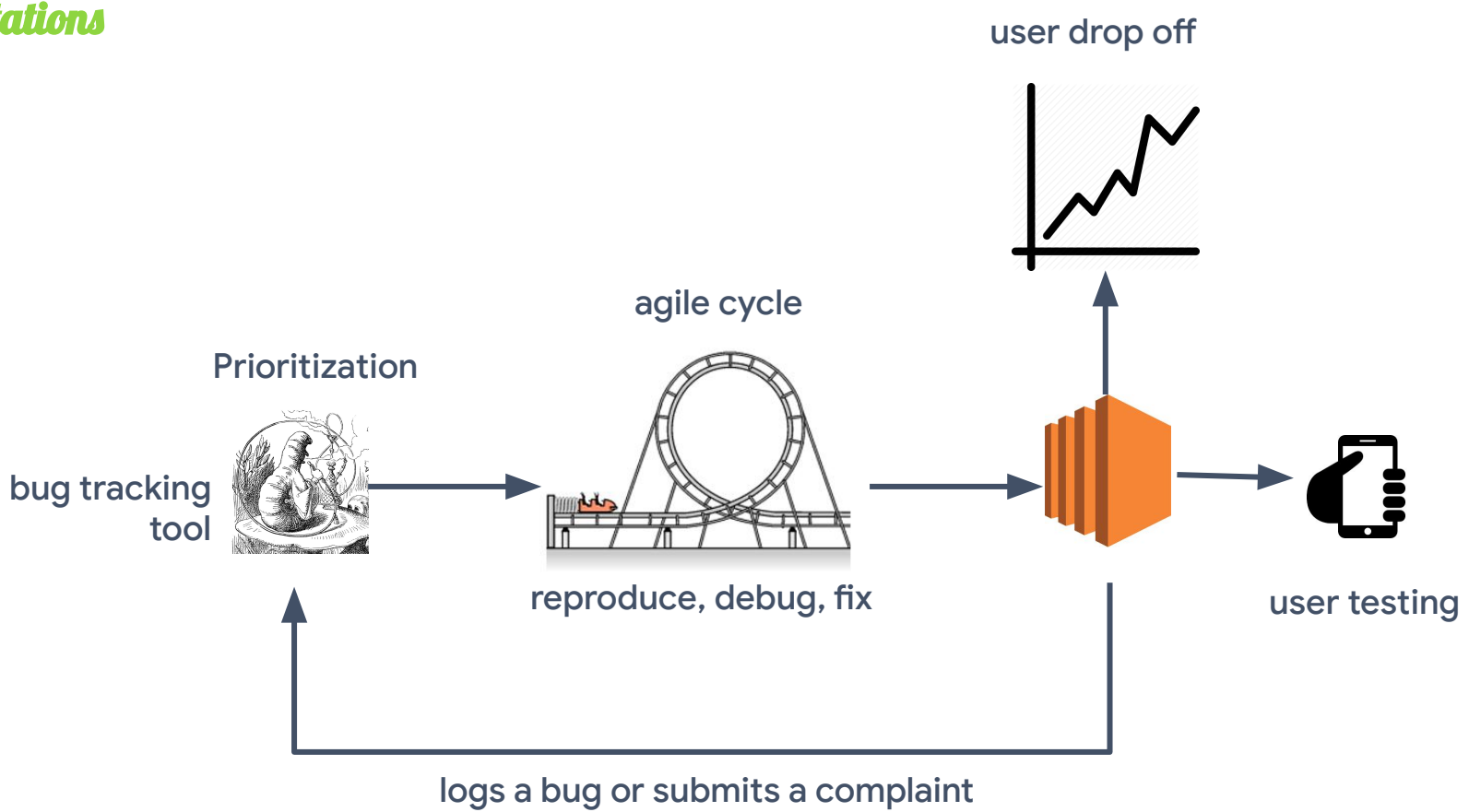


Experiment

Metadata Store

```
experiment: 3
data: ea2541df
code: da1341bb
desc: "Added feature to
training pipeline"
run_on: 10-10-2019
completed_on:11-10-2019
model: model-3
results: 3
```

Expectations



Actual

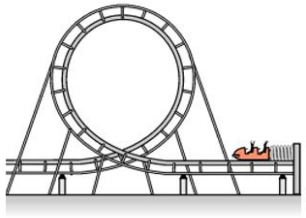
user reports bug → Identify problem → Describe problem with test patterns → Add to model bug tracking tool → Calculate Priority → Triage

"Agile Sprint"

Pick Problem

- Gather More Data for Problem
- Change Model
- Create Features

Retrain



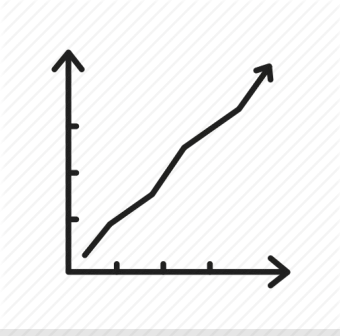
- Evaluate model against other models
- Evaluate individual problems
- Select model

Surprises

Surprises maintaining and improving the model over time

Expectation

Pick an issue → Generate/select unlabelled patterns → Get them labelled → Add to data set → Retrain



Surprise #8

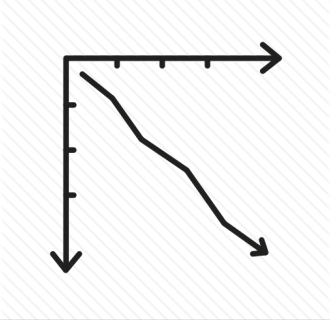
User behaviour drifts

Now what?

- Regularly sample data from production for training
- Regularly refresh your test set

Surprise #9

Data labellers are rarely experts



Surprise #10

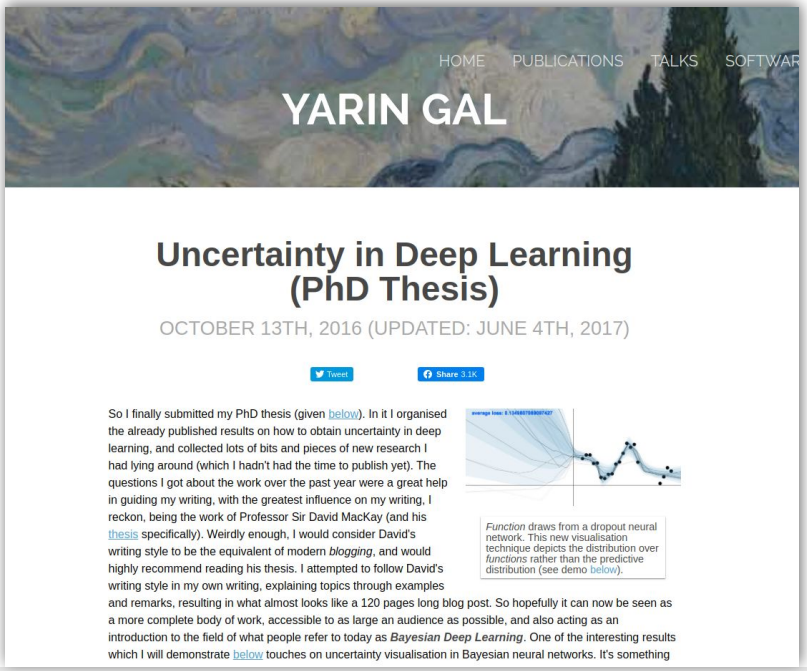
The model is not robust

Surprise #10

The model knows when it's uncertain

Techniques for detecting robustness & uncertainty

- Softmax predictions that are uncertain
- Dropout at Inference
- Add noise to data and see how much output changes



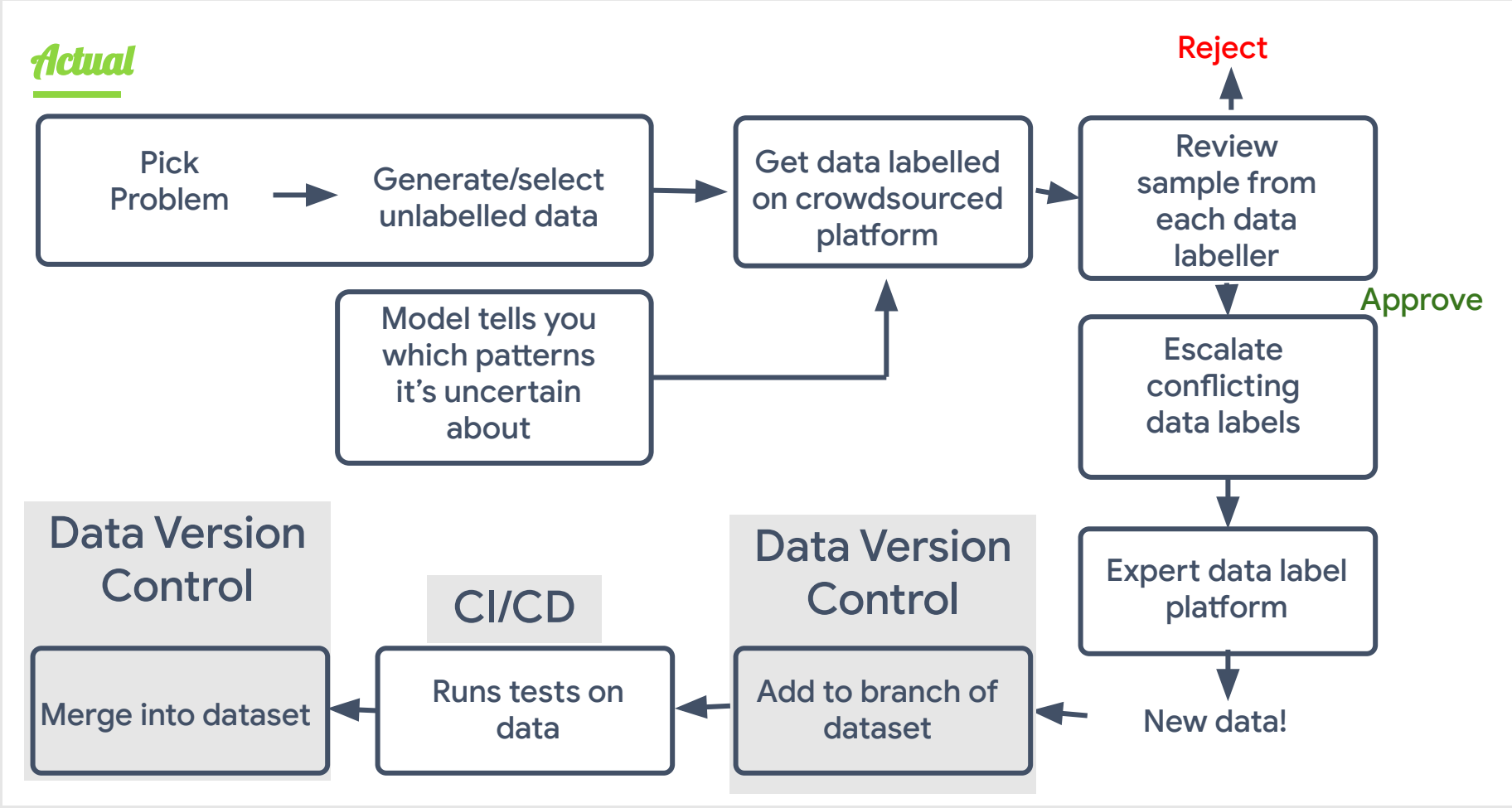
Surprise #11

Changing and updating the data so often gets messy

Needed to check the following

- Data Leakage
- Duplicates
- Distributions

Expectation


Actual
Pick Problem
Generate/select unlabelled data
Get data labelled on crowdsourced platform
Review sample from each data labeller
Reject
Approve
Escalate conflicting data labels
Expert data label platform
New data!
Model tells you which patterns it's uncertain about
Data Version Control
Add to branch of dataset
CI/CD
Runs tests on data
Data Version Control
Merge into dataset

The Checklist

First Release

Careful metric selection	✓
Threshold selection strategy	✓
Explain Predictions	✓
Fairness Framework	✓

The Checklist

After First Release

ML Problem Tracker	✓
Problem Triage Strategy	✓
Reproducible Training	✓
Comparable Results	✓
Result Management	✓
Be able to answer why	✓

The Checklist

Long term improvements & maintenance

Data refresh strategy	✓
Data Version Control	✓
CI/CD or Metrics for Data	✓
Data Labeller Platform + Strategy	✓
Robustness & Uncertainty	✓

Things I didn't cover

Pipelines & Orchestration	Kubeflow, MLFlow
End-to-end Products	TFX, Sage Maker, Azure ML
Unit Testing ML systems	"Testing your ML pipelines" by Kristina Georgieva
Debugging ML models	A field guide to fixing your neural network model by Josh Tobin.
Privacy	Google's Federated Learning
Hyper parameter optimization	So many!

The End

@alienelf

ja@retrorabbit.co.za

https://retrorabbit.co

https://kalido.me

https://masakhane.io