Simple Machine Learning

Christopher Hoult

Who is this guy?

DATASIFT

Previously...

Time Out

PURE INNOVATION

php BERKSHIRE

I also act

The Challenge

https://flic.kr/p/7DzC2S

How Insular is PHP?

- **Blog post by Larry Garfield (2015-08-24)**
  http://www.garfieldtech.com/blog/php-conference-data
- **Analysis of first time PHP speakers**
- **Used Joind.in events as a proxy**
- **Determined 50.6% first-time speakers**

Joind.in

- **https://joind.in**
- **Open source event site**
- **1084 events with 13871 talks**
- **409 events tagged; 675 untagged**
- **Not all events are about PHP!**

Correct as of 2016-02-13

Solutions

https://flic.kr/p/7vFcRJ

Tag By Hand

- 11 hours' work (1 per min)
- Typos; inconsistency
- What about future events?
- Not fun or cool!

Mechanical Turk

- Scalable
- Typos; inconsistency
- Automatable
- Expensive
- Kinda fun and cool

Machine Learning

- **Fast**
- **Consistent**
- **Automated**
- **Fun and cool!**

How do we learn?

https://flic.kr/p/rHv1bU

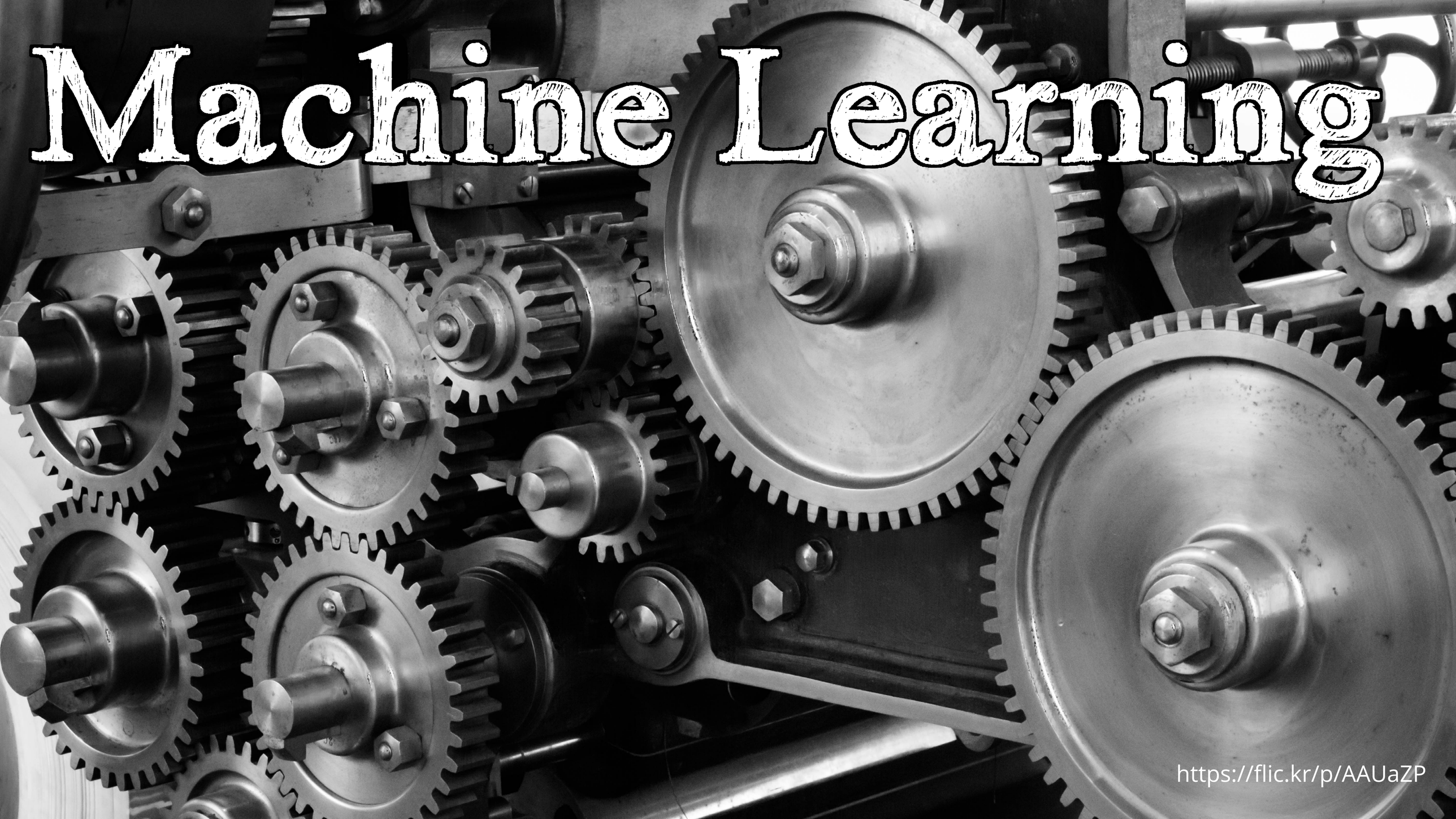
Pattern Recognition

- Brains are pattern recognition machines
- Correlation == Causation (ish)
- Stereotypes
- Baader-Meinhoff

Teaching

- Structured intro to ideas
- Teaching adds authority
- Therefore accuracy
- Chances to fail; be corrected

Machine Learning



https://flic.kr/p/AAUaZP

Unsupervised Learning

- No extra information provided
- Clustering of similar items
- Helps uncover hidden structure
- word2vec is a good example

Supervised Learning

- Existing documents are labelled
- Model build from knowns
- New documents labelled accordingly
- The existing labels are the supervision

Regression

- **Estimate the relationship between values**
- **Essentially line of best fit**
- **Used for continuous values**
  - **eg. predict a value *x* given *n* other variables**

Classification

- Associate labels with documents
- Just like in biology or libraries
- Requires known labels

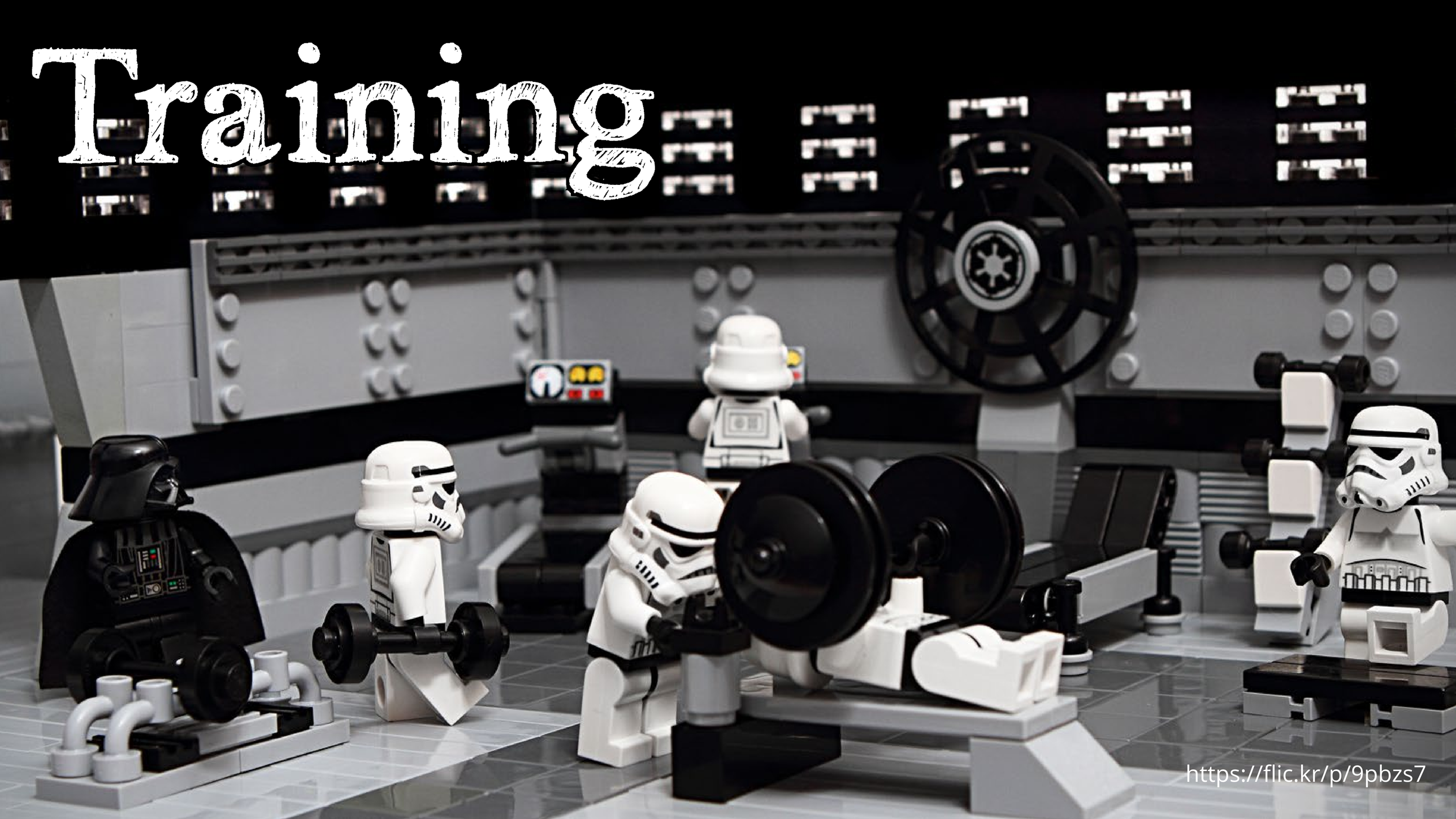
Our Approach

- Supervised classification
  - using Joind.in tags
  - using interactive tagging
- Multivariate Naive Bayesian Classification
- Bernoulli Multivariate NB Classification

Steps

- Training
  - Feature Extraction
  - Model building
- Prediction
  - Feature Extraction
  - Model application

Training



https://flic.kr/p/9pbzs7

Feature Extraction

- Machines can read
- Machines cannot comprehend
- Objectivity vs subjectivity
- Machines understand numbers
- So convert items to numbers!

Feature Extraction

- Numerous approaches:
  - Term frequency
  - tf*idf
  - N-gram presence
  - Can be constructed
  - Anything that creates a number
- ML is about identifying correlation

Feature Vectors

- Each feature is a dimension
- Matrix manipulation
- Also cool: cosine similarity
  - Modelled in n-dimension space
  - Requires positive values (eg. tf*idf)
  - Dot product over magnitude

Joind.in Example

Event: PHP UK Conference 2016

Talk: Simple Machine Learning

Description: *Want to* separate *the* signal *from the* noise, *but have too much* input *to* deal *with?* Fed *up with* reading everything *yourself?* Mechanical Turk *got you down? Then* perhaps *you need to* apply *some* machine learning*! In this* talk, Christopher *will* cover *some* basic approaches *to* machine-learned classification *as well as* demonstrate *a* real-life application *of it in* PHP.

Feature*	Value
event.title.php	1
event.title.uk	1
event.title.conference	1
talk.title.simple	1
talk.title.machine	1
talk.title.learning	1
description.word.separate	1
description.word.signal	1
description.word.noise	1
description.word.input	1
description.word.deal	1
description.word.fed	1
description.word.read	1
description.word.everything	1
description.word.mechanical	1
description.word.turk	1
description.word.perhaps	1
description.word.apply	2
description.word.machine	2
description.word.learn	2
description.word.talk	1
description.word.christopher	1
description.word.cover	1
description.word.basic	1
description.word.approach	1
description.word.classify	1
description.word.demonstrate	1
description.word.real	1
description.word.life	1
description.word.php	1

* Stop words removed and partially stemmed

Supervised Learning

- Start with a Training Set
  - Already tagged/classified
  - Considered representative of tag occurrence
- Extract features
- Correlate with tags

Probabilistic Correlation

- **Many methods**
- **Assume TS is a representative sample**

$$\text{Probability that a tag predicts a feature} = \frac{\text{Number of items with feature and tag}}{\text{Number of items with feature}}$$

Probabilistic Correlation

- **Also written as:**

$$P(\text{Feature} \mid \text{Tag}) = \frac{\text{Number of items with feature and tag}}{\text{Number of items with feature}}$$

Probabilistic Correlation

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

```php
public function generateModel()
{
    $this->model->reset();
    $cnt = 0;
    foreach ($this->tagFeatureList as $label => $labelFeatures) {
        $model = array_fill_keys(array_keys($this->featureList), 0);
        $model = array_merge($model, $labelFeatures);
        foreach ($this->featureList as $feature => $count) {
            $model[$feature] = ($model[$feature] + 1) / ($count + 1);
        }
        $this->model->setLabelModel($label, $this->tagCounts[$label], $model);
    }
    $this->model->setDocCount($this->docCount);
}
```

Probabilistic Correlation

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

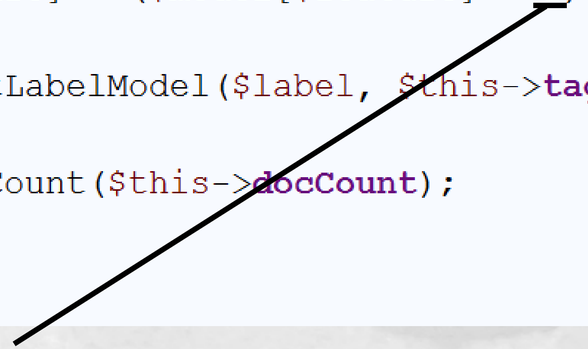
```php
public function generateModel()
{
    $this->model->reset();
    $cnt = 0;
    foreach ($this->tagFeatureList as $label => $labelFeatures) {
        $model = array_fill_keys(array_keys($this->featureList), 0);
        $model = array_merge($model, $labelFeatures);
        foreach ($this->featureList as $feature => $count) {
            $model[$feature] = ($model[$feature] + 1) / ($count + 1);
        }
        $this->model->setLabelModel($label, $this->tagCounts[$label], $model);
    }
    $this->model->setDocCount($this->docCount);
}
```

- **A list of all features encountered and their occurrence count**

Probabilistic Correlation

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

```php
public function generateModel()
{
    $this->model->reset();
    $cnt = 0;
    foreach ($this->tagFeatureList as $label => $labelFeatures) {
        $model = array_fill_keys(array_keys($this->featureList), 0);
        $model = array_merge($model, $labelFeatures);
        foreach ($this->featureList as $feature => $count) {
            $model[$feature] = ($model[$feature] + 1) / ($count + 1);
        }
        $this->model->setLabelModel($label, $this->tagCounts[$label], $model);
    }
    $this->model->setDocCount($this->docCount);
}
```



- **Laplace Smoothing reduces zeros in our model**

Probabilistic Correlation

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

```php
public function generateModel()
{
    $this->model->reset();
    $cnt = 0;
    foreach ($this->tagFeatureList as $label => $labelFeatures) {
        $model = array_fill_keys(array_keys($this->featureList), 0);
        $model = array_merge($model, $labelFeatures);
        foreach ($this->featureList as $feature => $count) {
            $model[$feature] = ($model[$feature] + 1) / ($count + 1);
        }
        $this->model->setLabelModel($label, $this->tagCounts[$label], $model);
    }
    $this->model->setDocCount($this->docCount);
}
```

- **Cheating (ssh!)**

Bayes' Theorem

$$P(T \mid F) = \frac{P(T)P(F \mid T)}{P(F)}$$

https://upload.wikimedia.org/wikipedia/commons/7/77/Sneeze.JPG

Multivariate Bayes'

$$P(T|F_1,F_2,F_3) = \frac{P(T)P(F_1|T)(F_2|T)(F_3|T)}{P(F_1,F_2,F_3)}$$

(See http://choult.com/blog/bayes-theorem-machine-learning)

Multivariate Bayes'

$$P(T \mid F_1, F_2, F_3) = \frac{\boxed{P(T)} P(F_1 \mid T)(F_2 \mid T)(F_3 \mid T)}{Z}$$

Also known as our *prior*
We'll assume this to be
docs tagged / # docs

Multivariate Bayes'

$$P(T|F_1,F_2,F_3) = \frac{P(T)P(F_1|T)(F_2|T)(F_3|T)}{Z}$$

We can use this number
as a lever for our model

Or In Code...

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

```php
private function predictLabel($label, array $features)
{
    $score = $this->model->getLabelCount($label) / $this->model->getDocCount();
    foreach (array_keys($features) as $feature) {
        if ($this->model->labelModelsFeature($label, $feature)) {
            $probability = $this->model->getLabelFeatureModel($label, $feature);
            $score *= $probability;
        }
    }
    return $score;
}
```

Or In Code...

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

```
private function predictLabel($label, array $features)
{
    $score = $this->model->getLabelCount($label) / $this->model->getDocCount();
    foreach (array_keys($features) as $feature) {
        if ($this->model->labelModelsFeature($label, $feature)) {
            $probability = $this->model->getLabelFeatureModel($label, $feature);
            $score *= $probability;
        }
    }
    return $score;
}
```

- Our *prior*

Or In Code...

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

```
private function predictLabel($label, array $features)
{
    $score = $this->model->getLabelCount($label) / $this->model->getDocCount();
    foreach (array_keys($features) as $feature) {
        if ($this->model->labelModelsFeature($label, $feature)) {
            $probability = $this->model->getLabelFeatureModel($label, $feature);
            $score *= $probability;
        }
    }
    return $score;
}
```

- **Is this feature in our model? If not, skip it**

Or In Code...

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

```php
private function predictLabel($label, array $features)
{
    $score = $this->model->getLabelCount($label) / $this->model->getDocCount();
    foreach (array_keys($features) as $feature) {
        if ($this->model->labelModelsFeature($label, $feature)) {
            $probability = $this->model->getLabelFeatureModel($label, $feature);
            $score *= $probability;
        }
    }
    return $score;
}
```

- **P(Tag|Feature)**

Or In Code...

\Choult\Enamel\Classifier\MultiVariateNaiveBayes.php

```php
private function predictLabel($label, array $features)
{
    $score = log($this->model->getLabelCount($label) / $this->model->getDocCount());
    foreach (array_keys($features) as $feature) {
        if ($this->model->labelModelsFeature($label, $feature)) {
            $probability = $this->model->getLabelFeatureModel($label, $feature);
            $score += log($probability);
        }
    }
    return $score;
}
```

- **Multiplication makes small numbers smaller - so use logs**

Demo Time

https://flic.kr/p/8UhM1c

Human Learning

https://flic.kr/p/88bgji

Lessons Learned

- Bayes' Theorem is annoying
- Joind.in tags:
  - Heavily PHP-biased
  - Noisy!
- NB takes a lot of memory
  - 350 tags * 13200 features * 76 bytes == lots!

Next Steps

- Improve on feature selection
- Implement more classifiers and weighting
- Build model into Joind.in
- Improve performance!
  - Investigate matrix extensions
  - Caching etc.

The Code

- Enamel
  - https://github.com/choult/enamel
  - Work in progress!
- Joind.in Audit
  - https://github.com/choult/joindinaudit
  - Thanks to Larry Garfield


Questions?
http://bit.ly/1npSVI6

Thank you!

 @choult

 christopherhoult

@ chris@choult.com

Feedback please! https://joind.in/talk/b993b

This presentation was typeset in HandTIMES and Open Sans, using Adobe InDesign CS6
(because I'm a lunatic)