

ON COMPUTING NUMBERS, WITH AN APPLICATION TO  
PROBLEMS OF OUR SOCIETY

Journalism ♥ Computer Science

Stefan Wehrmeyer





# Investigative Journalism





# Investigative Journalism & Computers





# Natural Language Processing



# Machine Learning

The background is a complex collage. On the left, a large, detailed parrot with vibrant orange, yellow, and blue feathers is the central focus. To its right, a modern building at night is visible, with its windows glowing and the text 'CONGRESS CENTRUM HAMBURG' clearly legible on its facade. The right side of the image features a globe with a grid of latitude and longitude lines, set against a dark, starry background. The overall aesthetic is high-tech and global.





Los Angeles Times

**LAPD misclassified more than 25,000 serious crimes as minor, audit finds**



The background features a complex network graph with numerous nodes and edges. The nodes are represented by circles of varying sizes, with some being significantly larger than others. The edges are thin white lines connecting the nodes. The overall color palette is light blue and grey, with several nodes highlighted in a bright yellow-green. The text "Social Network Analysis" is centered in a black, serif font.

# Social Network Analysis



# Algorithmic Accountability



# Journalism & science



## Train the classifiers

For this analysis we used two machine learning classifiers. The first is a linear [support vector machine](#) from the stellar [scikit-learn Python library](#). The second is a [maximum entropy classifier](#). For the official analysis I used the [MegaM](#) optimization package to dramatically improve the training speed. Here, for simplicity, I'm using the NLTK built in trainer.

```
In [6]: # Train our classifiers. Let's start with Linear SVC
# Make a data prep pipeline
pipeline = Pipeline([
    ('tfidf', TfidfTransformer()),
    ('linearsvc', LinearSVC()),
])
# make the classifier
linear_svc = SklearnClassifier(pipeline)
# Train it
linear_svc.train(features)
```

```
Out[6]: <SklearnClassifier(Pipeline(steps=[('tfidf', TfidfTransformer(norm='l2', smooth_idf=True, sublinear_tf=False, use_idf=True)), ('linearsvc', LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True, intercept_scaling=1, loss='squared_hinge', max_iter=1000, multi_class='ovr', penalty='l2', random_state=None, tol=0.0001, verbose=0))]))>
```

```
In [7]: # Next, let's do the Maximum Entropy
maxent = MaxentClassifier.train(features)
```

```
==> Training (100 iterations)
```

```
Iteration    Log Likelihood    Accuracy
-----
```



# Software engineering in the newsroom



*Applications for our society.*



# Thank you.

Stefan Wehrmeyer  
correctiv.org  
[mail@stefanwehrmeyer.com](mailto:mail@stefanwehrmeyer.com)