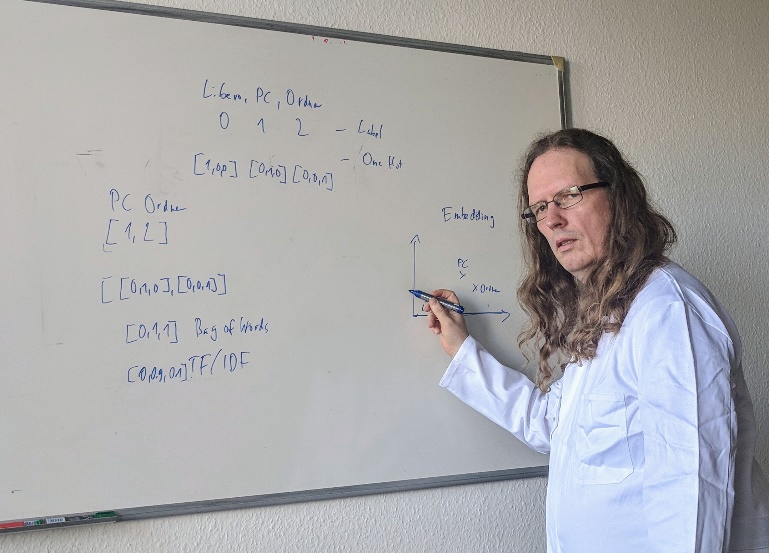
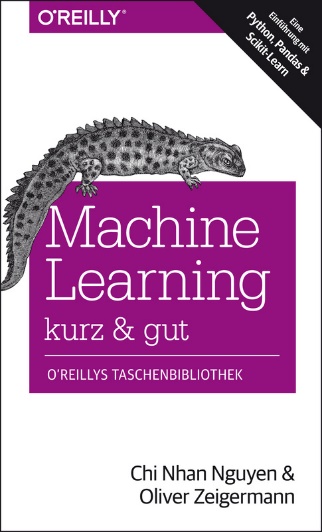
# Portfolio AI/Machine Learning/Big Data

## Oliver Zeigermann, <http://zeigermann.eu/>, [OliverZeigermann@gmail.com](mailto:OliverZeigermann@gmail.com)

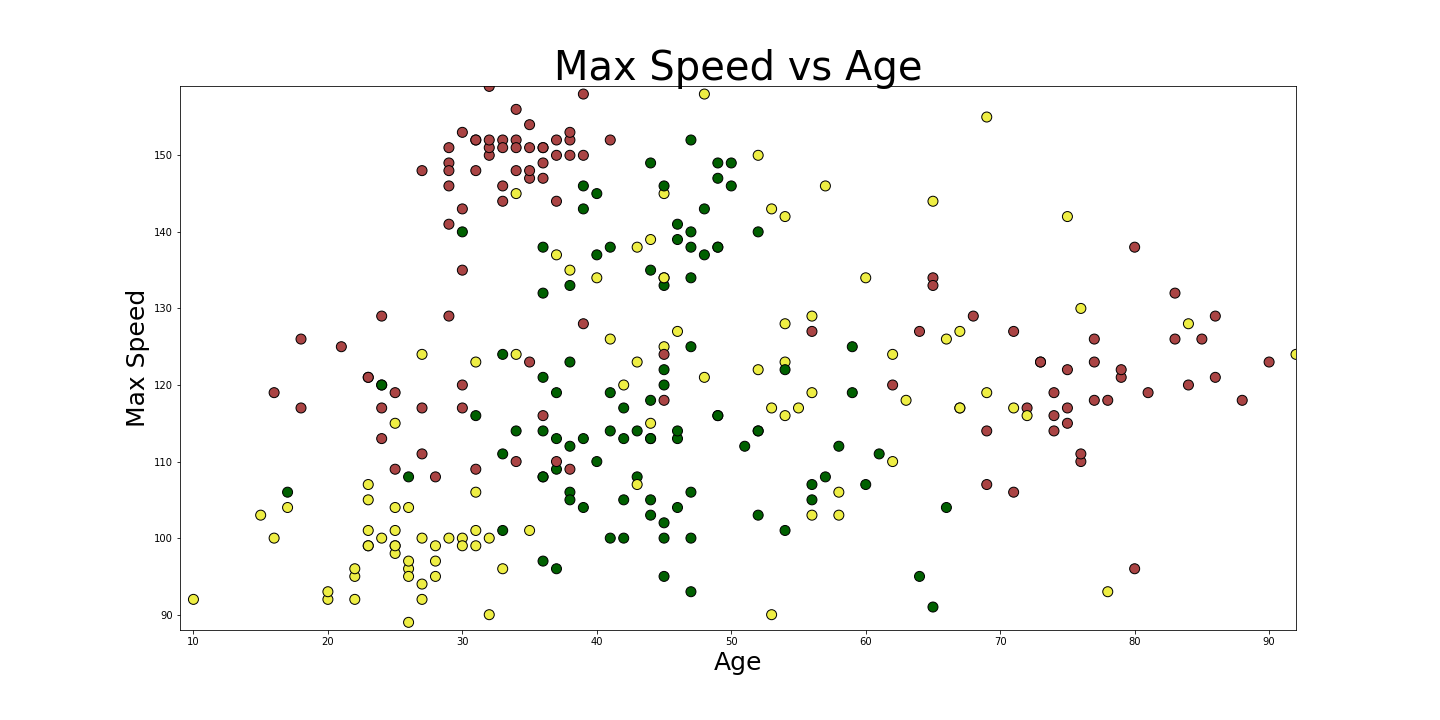
## Diplom Informatiker (German M. Sc.) Uni Hamburg / Software Engineer / Architect / Data Scientist / Machine Learning Expert

## TensorFlow / TensorFlow.js / Scikit-Learn / Pandas / Matplotlib / Elasticsearch

## Consulting / Coaching / Workshops / Projects

# Classic Supervised Machine Learning



**Objective**: Predict a category/value from tabular input data

**Requirements**: You have tabular data, and sample solutions

**Example**: Classify perspective car insurance customers into risk groups based on top-speed, age, and mileage

**Technology**: Python, Pandas, Matplotib, Scitkit-Learn, Random Forest

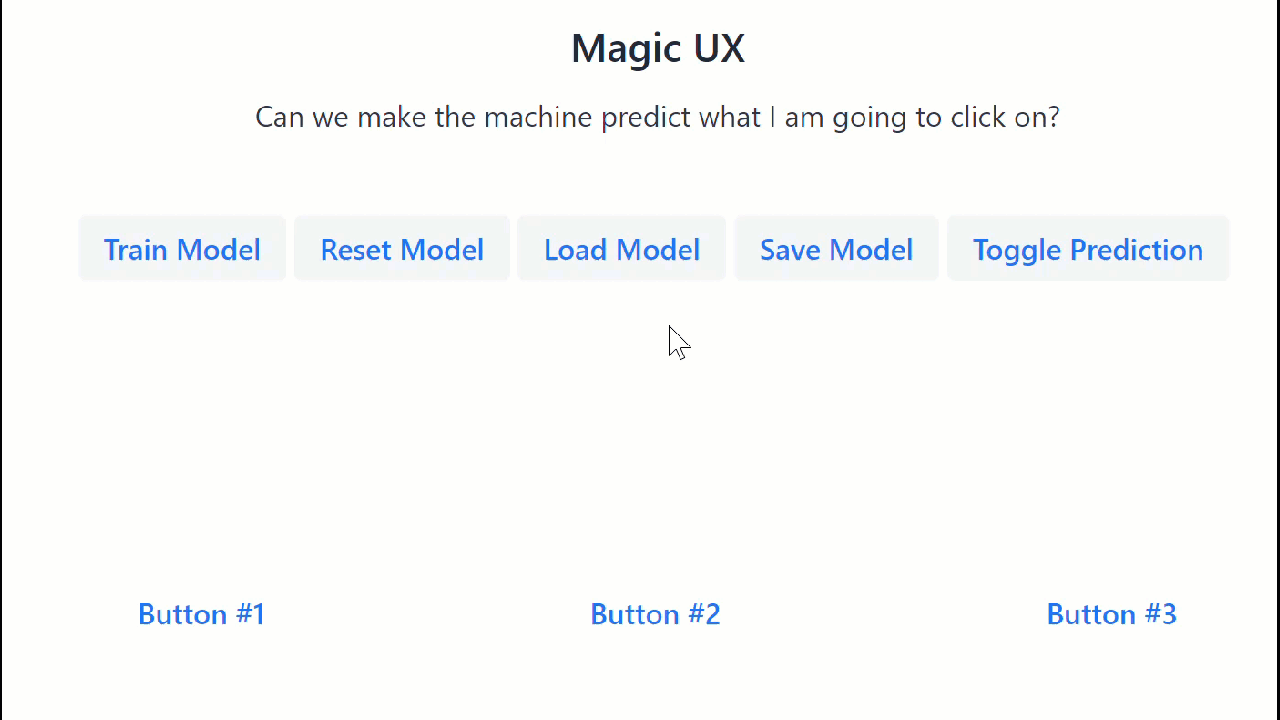
**References**: (Colab Notebook)

<https://colab.research.google.com/github/djcordhose/ai/blob/master/notebooks/sklearn/dt-intro.ipynb>

<https://colab.research.google.com/github/djcordhose/ai/blob/master/notebooks/sklearn/overview.ipynb>

<https://djcordhose.github.io/crash-risk-calculator/>

# Deep Supervised Machine Learning



**Objective**: Predict a category/value from all kinds of input data

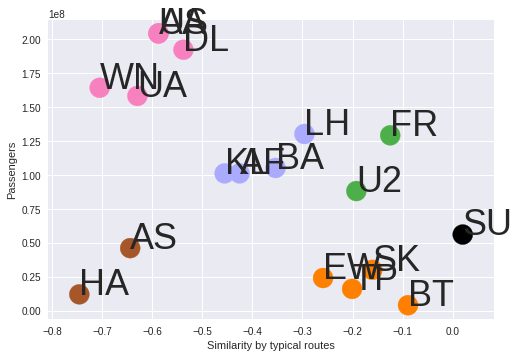
**Requirements**: You have a lot of data/image data/time series data, and sample solutions

**Example**: Predict what the user is going to click next and make the button more accessible based on a sequence mouse positions and directions

**Technology**: TensorFlow.js, GRU Recurrent Networks

**References**: (Github Project): <https://github.com/DJCordhose/ux-by-tfjs>

# Unsupervised Machine Learning



**Objective**: Find similarities or outliers in any kind of data

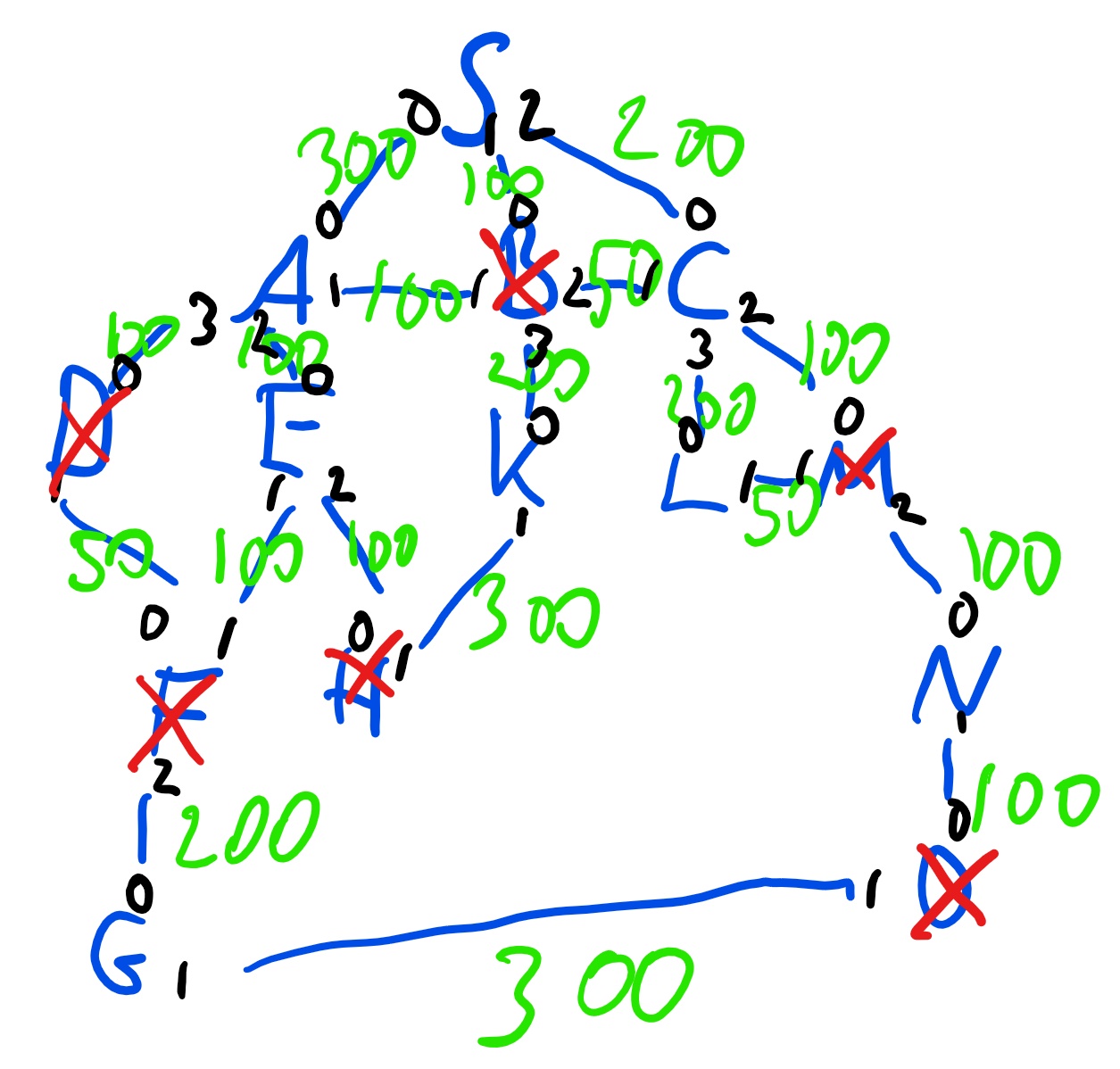
**Requirements**: structured, unstructured or text data, but no sample solution

**Example**: Find similar airlines and outliers, embed airlines into a 1-d embedding

**Technology**: TensorFlow, Embeddings, GRU Recurrent Networks

**References**: (Poster): <https://djcordhose.github.io/ai/poster-sf.png>

# Reinforcement Learning



**Objective**: Find near optimal solutions when having no training data

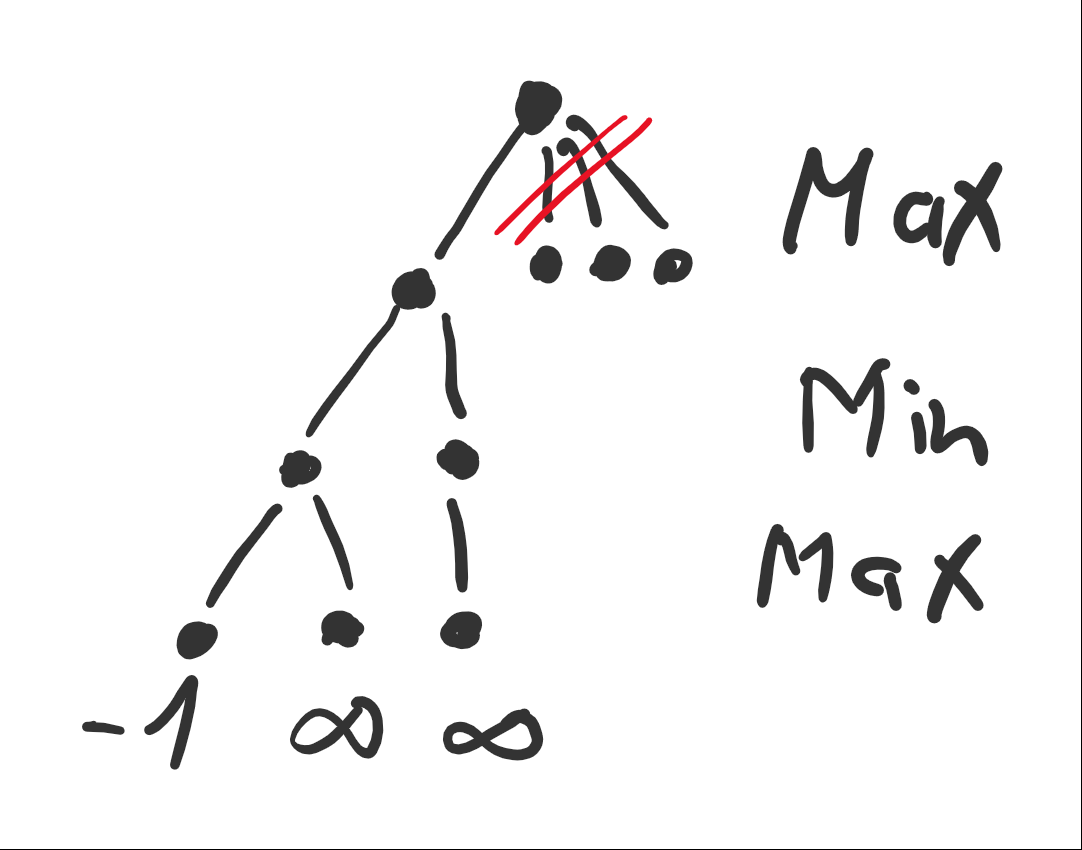
**Requirements**: no static data required, but a simulated environment that an agent can make experiments

**Example**: How to deploy consultants to customers when size of problem does not allow for deterministic solution

**Technology**: TensorFlow, OpenAI Platform and Baselines

**References**: (Colab Notebook): <https://colab.research.google.com/github/DJCordhose/ai/blob/master/notebooks/rl/berater-v11.ipynb>

# Classic AI / Search



**Objective**: Find deterministic solutions for a problem that can be stated clearly

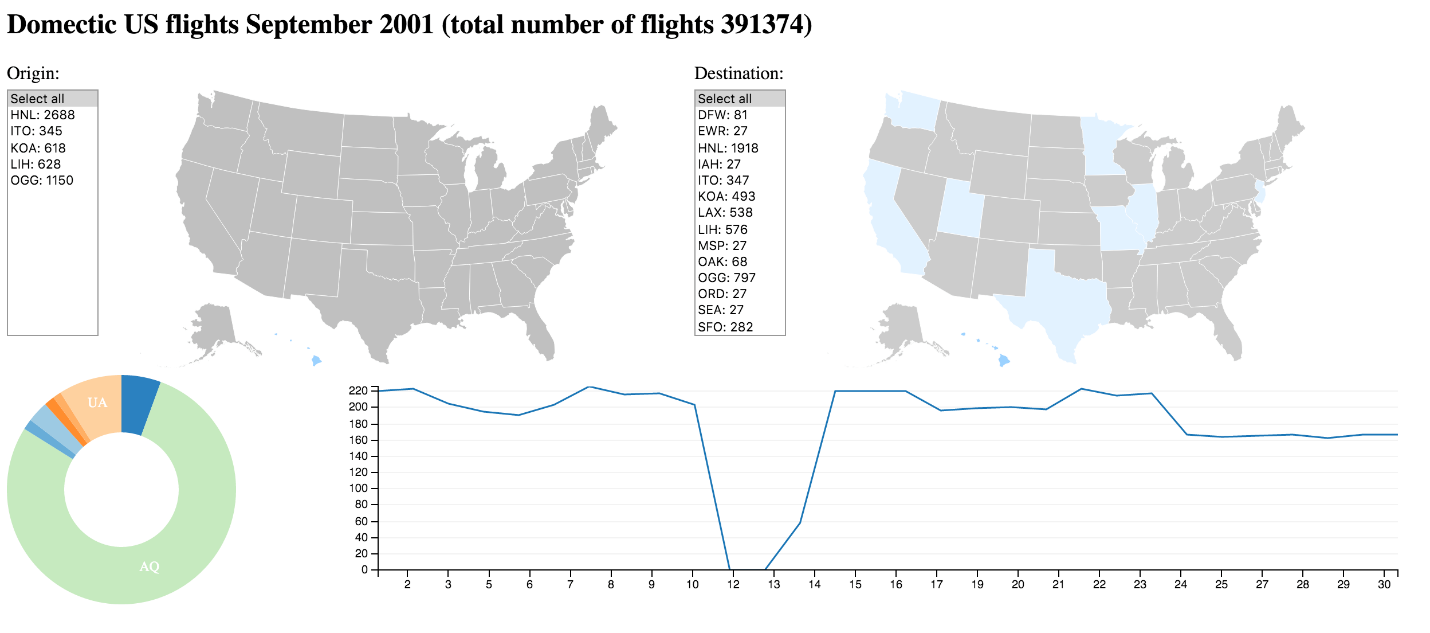
**Requirements**: no data required, problem must be translatable to search on a graph

**Example**: Search for an ideal move in a chess or go game

**Technology**: Python, Classic Program Code

**References**: (Slides): <http://bit.ly/mlconf-search>

# Data Science



**Objective**: Insights rather than predictions

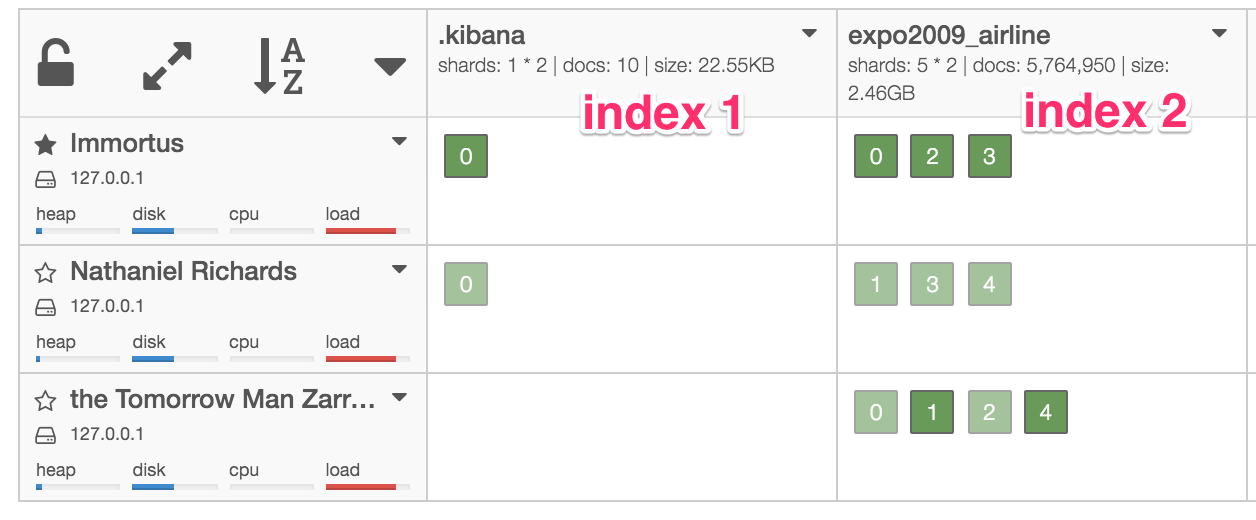
**Requirements**: any kind of structured, tabular data

**Example**: Exploration – what insights can we gain on domestic American flights

**Technology**: Pandas, Elasticsearch, D3

**References**: (Slides): <http://bit.ly/data-exploration-odsc>

# Database Systems



**Objective**: Make large amounts of data searchable

**Requirements**: any kind of semi-structured or tabular data

**Example**: Store large amounts of text data and it searchable in full text

**Technology**: Elasticsearch

**References**: (Slides): <http://djcordhose.github.io/introduction-to-elasticsearch/2016_devcon.html>