

GitHub Klassifizierung

Andreas Grafberger, Martin Keßler, Michael Leimstädtner, Stefan Grafberger

Team Universität Augsburg



Andreas
Grafberger



Martin
Keßler



Michael
Leimstädtner



Stefan
Grafberger

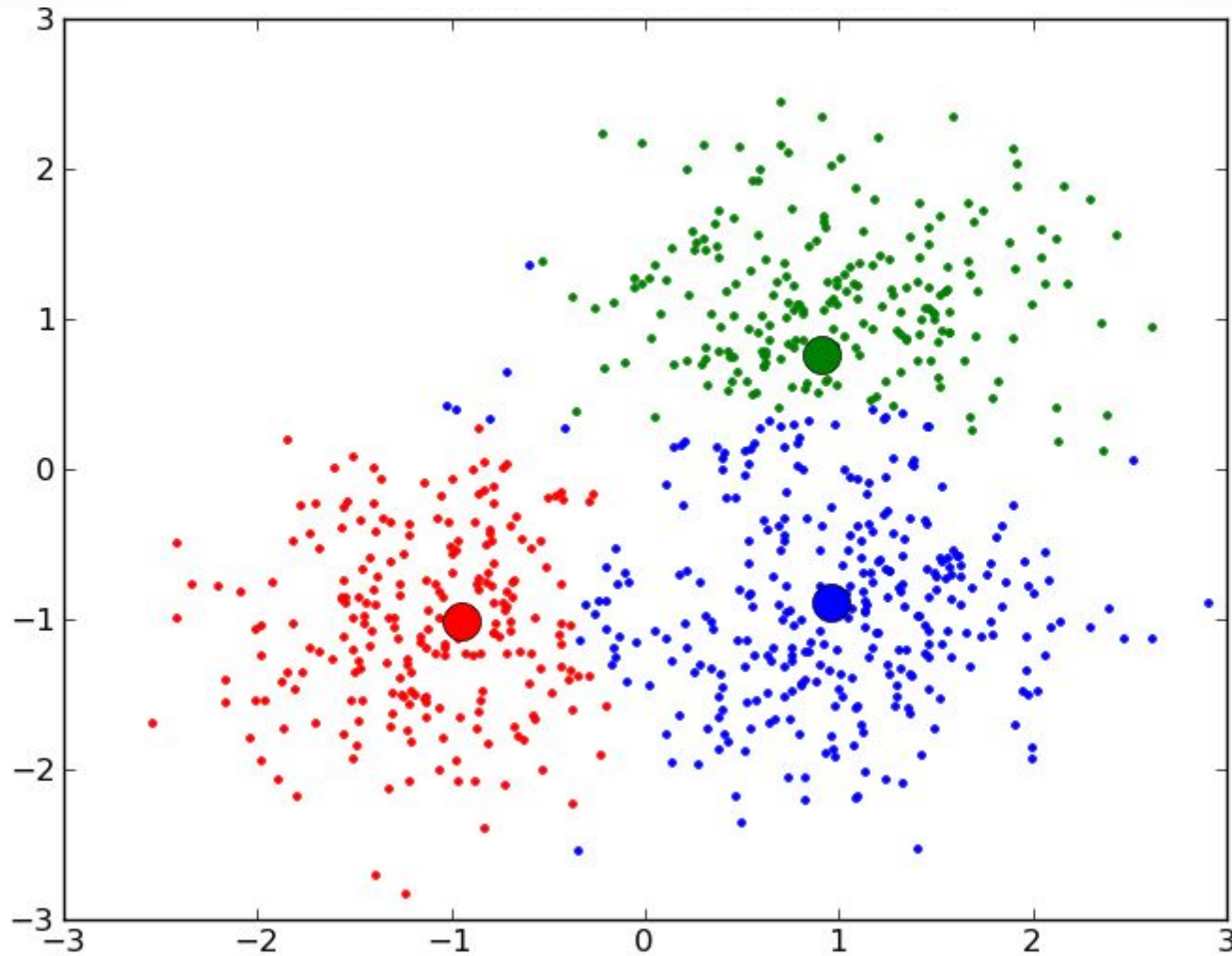
Gliederung

- Herausforderungen und unser **Lösungsansatz**
- **Demonstration** der Anwendung
- **Features** und **Vorhersagemodelle**
- **Fragen**

Besonderheiten und Herausforderungen

Probleme

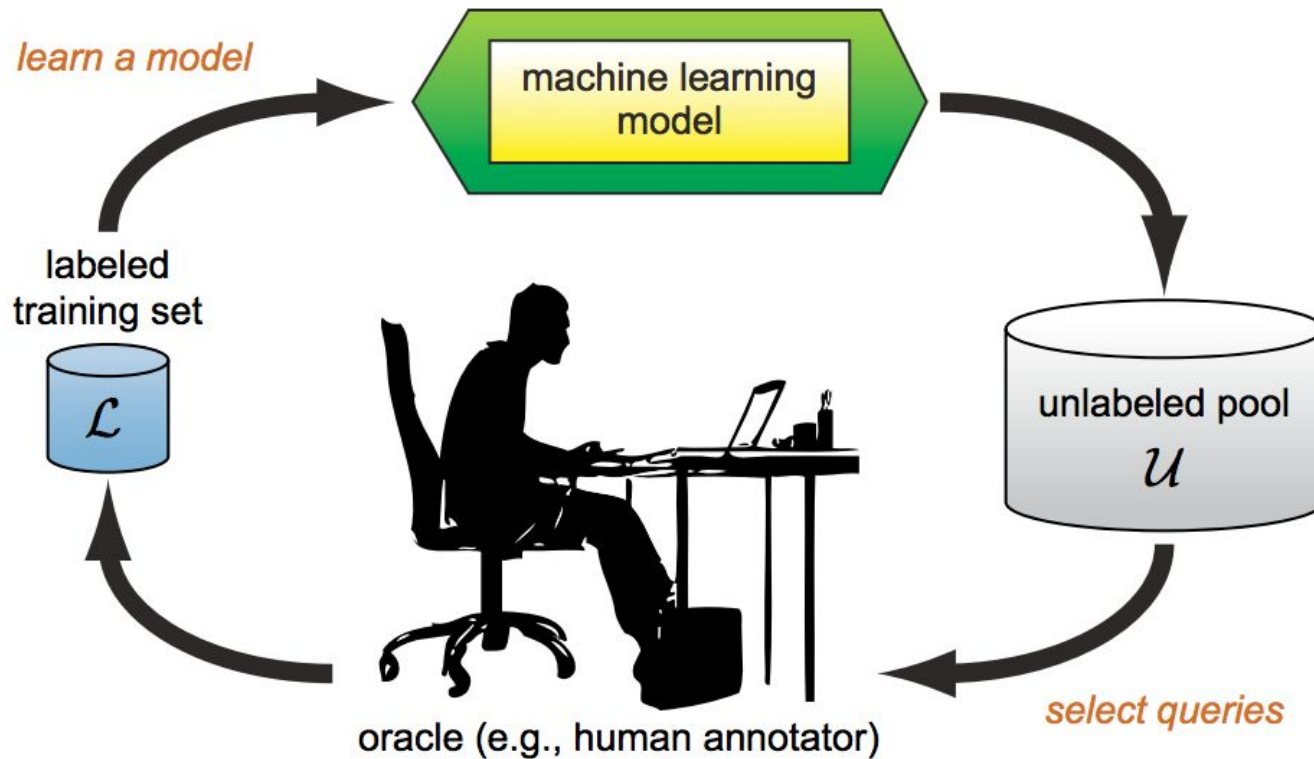
- Unklare **Klassenbeschreibungen**
- **Ungleichgewicht** der Klassenverteilung
- **Hochdimensionaler** unbekannter Feature-Raum



Quelle: http://scikit-learn.sourceforge.net/0.5/_images/plot_mean_shift.png

Generierung der Trainingsdaten - Probleme

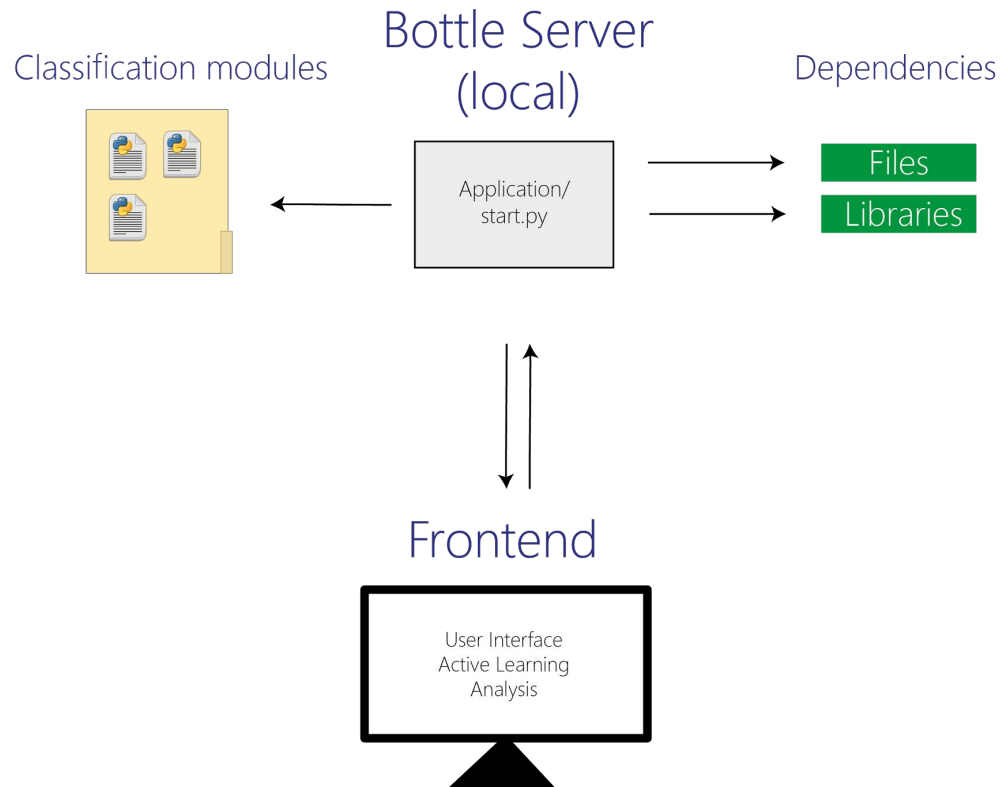
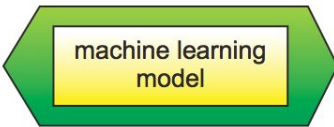
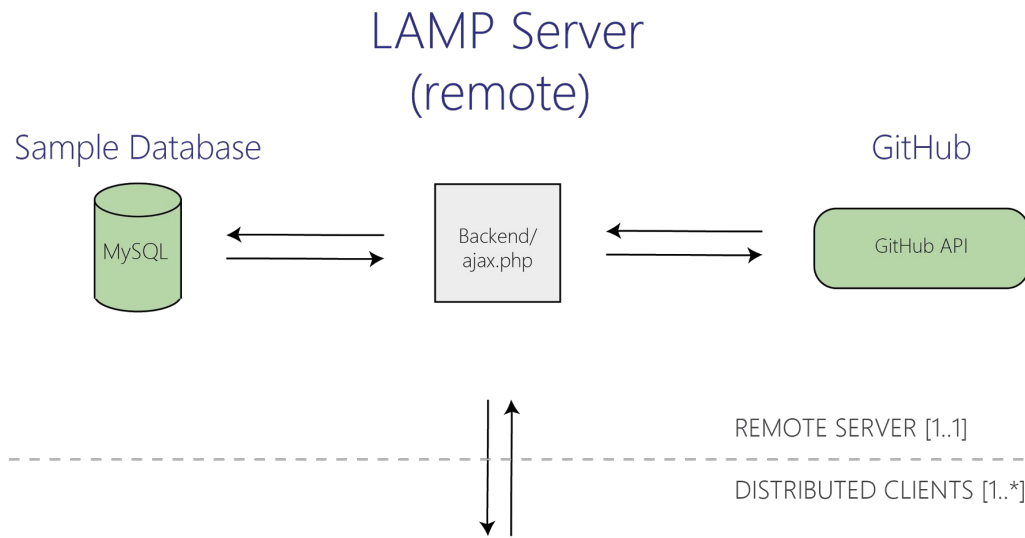
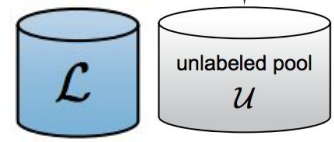
- Welche Repositories interessant?
- Keine scharfen **Grenzen** zwischen den Klassen
- **Majority** class
- Großer Feature-Vektor



Generierung der Trainingsdaten - Active Learning

- Stream- und Pool based
- Unsicherheit auf Basis der Klassen-Wahrscheinlichkeiten

⇒ geringe Redundanz, 2.000 Samples



Softwaredesign

- lokal: **Python - Server**
- remote: **LAMP - Server**
- DB: **MySQL**
- GUI: **HTML, CSS, Vue.JS**

Classification Overview

Note: Although we list more than one classification module, only the top listed preordered module should be considered as our submission.

☐ Stream Based AL

☐ Pool Based AL

☒ Test all Classifiers

☐ Handle user input

Retrain all classifiers

Save all classifiers

☒ Use extended test set

Test classifiers

Statistics

The **testing option** runs each classifier against a predefined set of test samples, thus updating the confusion matrix and precision per class.

Input

Test sample distribution

Class	#Test samples
DATA	32
DEV	89
DOCS	48
EDU	35
HW	53
OTHER	26
WEB	29

Classifiers

Order classifiers by

Preordered

Stacking NN

66%

DEV

HW

EDU

DOCS

WEB

DATA

OTHER

0.46

0.78

0.41

0.56

0.89

1

0.84

Description and Metadata Gradient Tree Boosting

62%

DEV

HW

EDU

DOCS

WEB

DATA

OTHER

0.52

0.82

0.33

0.36

0.65

1

0.8

ALL Support Vector Classifier

61%

DEV

HW

EDU

DOCS

WEB

DATA

OTHER

0.48

0.63

0.44

0.53

0.83

0.86

0.69

All NN

60%

DEV

HW

EDU

DOCS

WEB

DATA

OTHER

0.63

0.63

0.43

0.45

0.66

0.78

0.63

Readme and Meta Support Vector Classifier

58%

DEV

HW

EDU

DOCS

WEB

DATA

OTHER

0.5

0.79

0.32

0.42

0.63

0.93

0.6

Output

Top-most listed classifier performance

Final classifications are being made by **Stacking NN**. Its measures are listed below

Measure	Result
Precision M	70.4%
Recall M	52.4%
Fscore M	65.8%
Average Accuracy	80.8%
Error Rate	59.5%
Precision μ	56.1%
Recall μ	56.1%
Fscore μ	56.1%

Best precision of each classifiers

Class	#Best scores
DATA	7
WEB	1
DEV	1

Live Demo

- Struktur
- Use Cases

[samshadwell](#) / [TrumpScript](#)

122



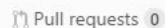
5,501



329



Code



Pull requests 0



Projects 0



Pulse



Graphs

Make Python great again

235 commits

3 branches

0 releases

34 contributors

MIT

Branch: master ▾

[New pull request](#)[Find file](#)[Clone or download ▾](#)

samshadwell committed on GitHub Made the final update.

Latest commit a76f078 on 1 Feb

bin	Update TRUMP	a year ago
src/trumpscript	Merge pull request #157 from fredkneeland-wf/rigged	4 months ago
test	Should I submit a CVE for this?	4 months ago
.gitignore	Use a relative path to module folder	a year ago
Brainstorming.txt	Added our sweet new logo	a year ago
Grammar.txt	Built the skeleton of my parser	a year ago
LICENSE	Initial commit	a year ago
README.md	Made the final update.	a month ago
TrumpScript.jpg	Added our sweet new logo	a year ago

README.md

Final Update

It's been a while since we made any updates to TrumpScript, and we just wanted to make it official that our development on this project has stopped and that we will no longer be accepting issues or pull requests on this repo.

Frankly, this joke isn't funny anymore. Rather than spend your time beating the "Trump is ridiculous" meme to death, please actually do something instead and donate to:

- American Civil Liberties Union
- National Resources Defense Council
- Planned Parenthood

Features

- Welche gibt es?
- Welche brauchen wir Menschen?
- Welche eignen sich?
- Codierung / Pre-Processing

Text Features - Codierung

1. Bag-of-words model
 - a. Term frequency (**Tf**) + inverse document frequency (**Tf-idf**)
 - b. Stemming
2. N-grams
3. Character-sequence (**RNN**, **LSTM**, ...)
4. Word2Vec

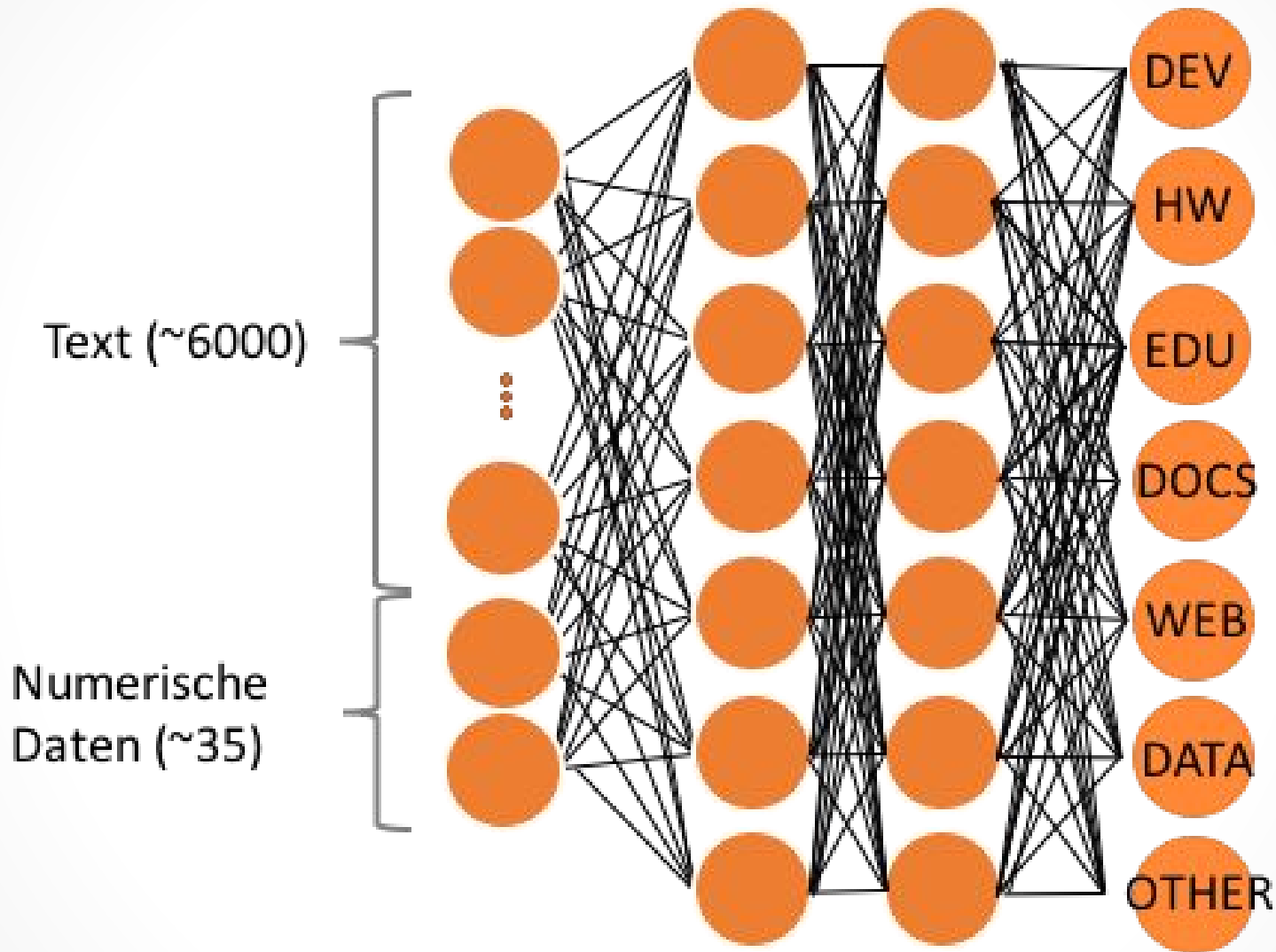
Vorhersagemodelle

Base Classifier

- Support Vector Machine
- Linear Classifiers
- Naive Bayes
- Decision Trees
- **Neuronale Netze**

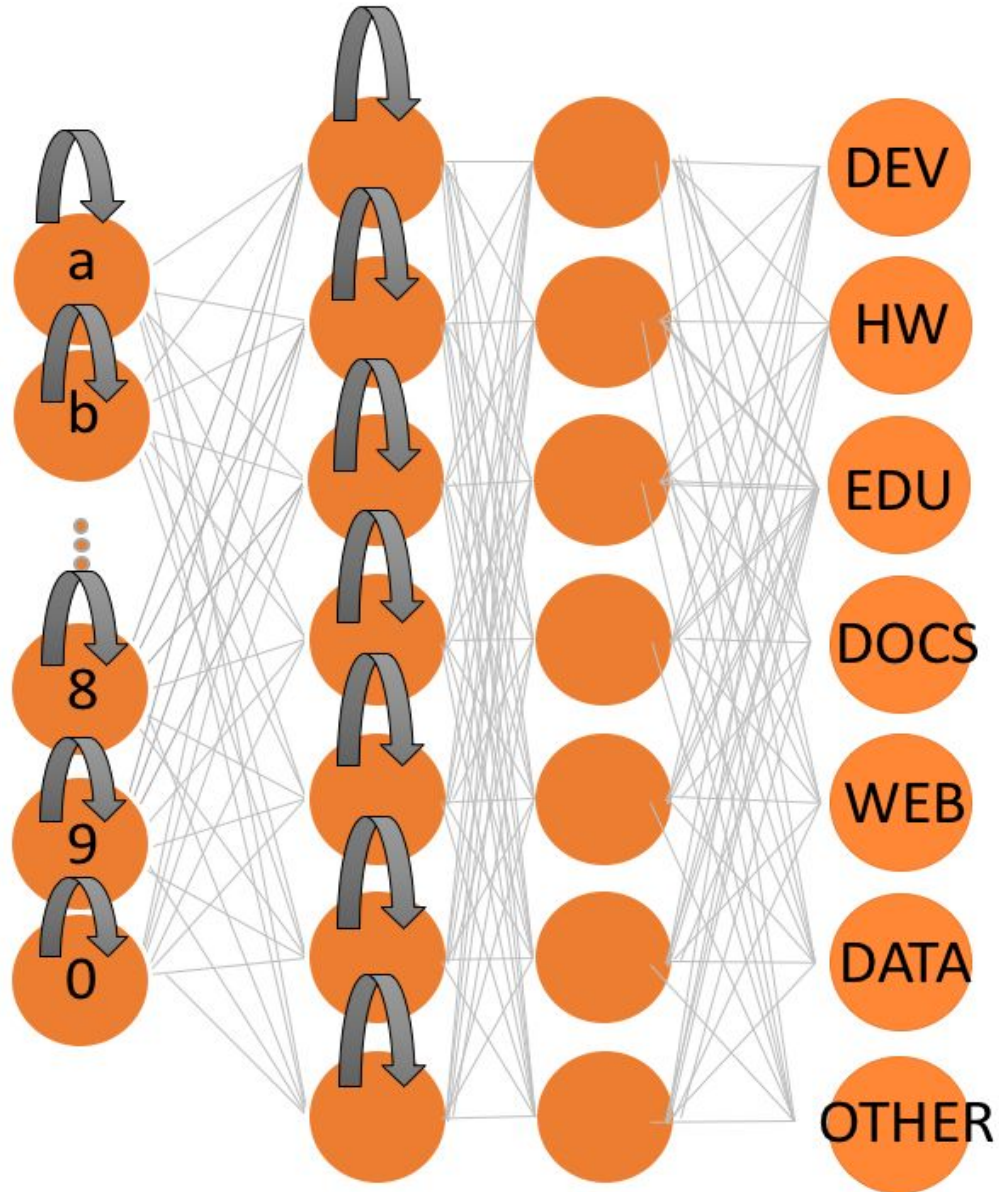
Ensemble Classifier

- Boosting
 - Gradient Tree Boosting
 - AdaBoost
- Bagging
 - Random Forest
- **Stacking**



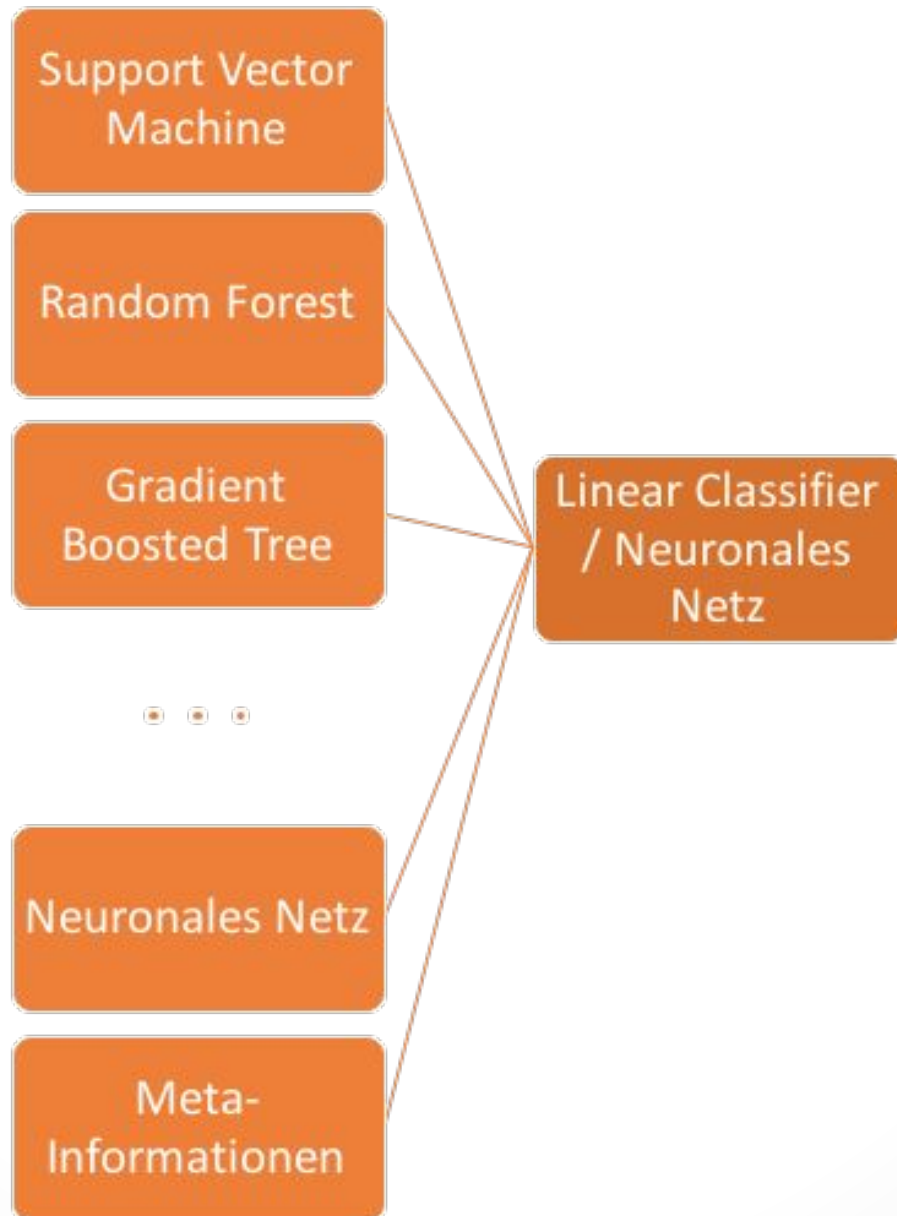
Vorhersagemodelle - Neuronale Netze

- Feed Forward Network
- 2-3 hidden layer



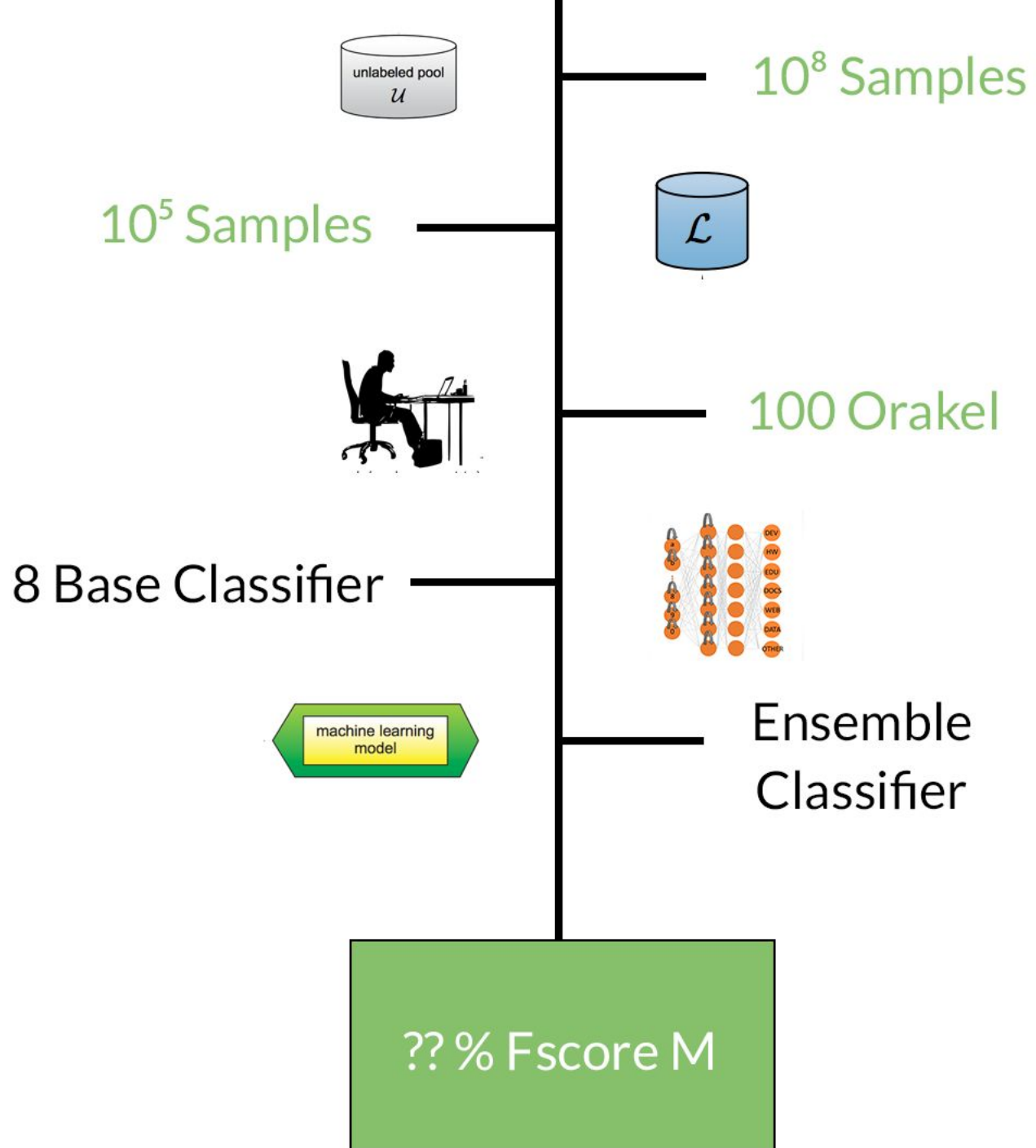
Vorhersagemodelle - Neuronale Netze

- Recurrent Neural Network
→ LSTM
- Benutzt Repository-Name



Vorhersagemodelle - Problemlösungen

- 60% - Grenze
→ Kombination aller Vorhersagen
- Class imbalance
→ Class-Weights
- Overfitting /
Vanishing Gradients



Zusammenfassung

Vielen Dank für Ihre Aufmerksamkeit



theano



Index der Zusatzfolien

- [Decision Tree / Random Forest](#)
- [Unsicherheitsformeln](#)
- [Libraries und Technologien](#)
- [Liste verwendeter Features](#)
- [Evaluation](#)
- [Erster Softwareentwurf](#)

Unsere Anwendung - Ziele

- Sandbox für verschiedene Klassifikatoren
- Active Learning
- Features, Trainings- und Testdaten
- Eine GUI für verschiedene Zwecke?
- Plattformunabhängig
- Leichte Evaluation verschiedener Modelle

Beispiel Decision Tree / Random Forest



Features

Text

- Repository - Name
- Autor
- Short - Description
- Readme
- Dateinamen
- Ordnernamen
- Dateitypen

Numerisch

- # Commits, Forks, Folders / Files, ...
- ø Länge der Commit-Nachricht
- ø Levenshtein - Distanz der Ordner-/Dateinamen
- Verwendete Sprachen

Boolean (1 oder 0):

- Besitzt Wiki
- Ist ein Fork, ...

Measure table

Measure	Result
Precision M	67.73 %
Recall M	53.93 %
Fscore M	64.43 %
Average Accuracy	79.6 %
Error Rate	62.24 %
Precision μ	58.97 %
Recall μ	58.97 %
Fscore μ	58.97 %

Evaluation - Metriken

- Precision vs. Recall
- Interessanter: **Precision**
- Noch besser: **FScore M**

Confusion matrix

▼ Class. \ Reference ►	DEV	HW	EDU	DOCS	WEB	DATA	OTHER	Total	Precision
DEV	9	4	1	1	0	1	0	16	0.56
HW	0	2	0	0	0	0	0	2	1
EDU	1	1	2	0	0	1	0	5	0.4
DOCS	0	0	1	2	0	0	0	3	0.67
WEB	0	0	1	0	3	0	0	4	0.75
DATA	0	0	1	0	0	0	0	1	0
OTHER	0	0	0	0	0	0	0	0	0
Total	10	7	6	3	3	2	0	31	0
Recall	0.9	0.29	0.33	0.67	1	0	0	0	0.58

Evaluation - Konfusionsmatrix

- Ermöglicht detaillierte Analyse der Klassifikatoren
- Schön ersichtlich, wo der Klassifikator noch Probleme hat

Evaluation - Unsere Ergebnisse in Zahlen

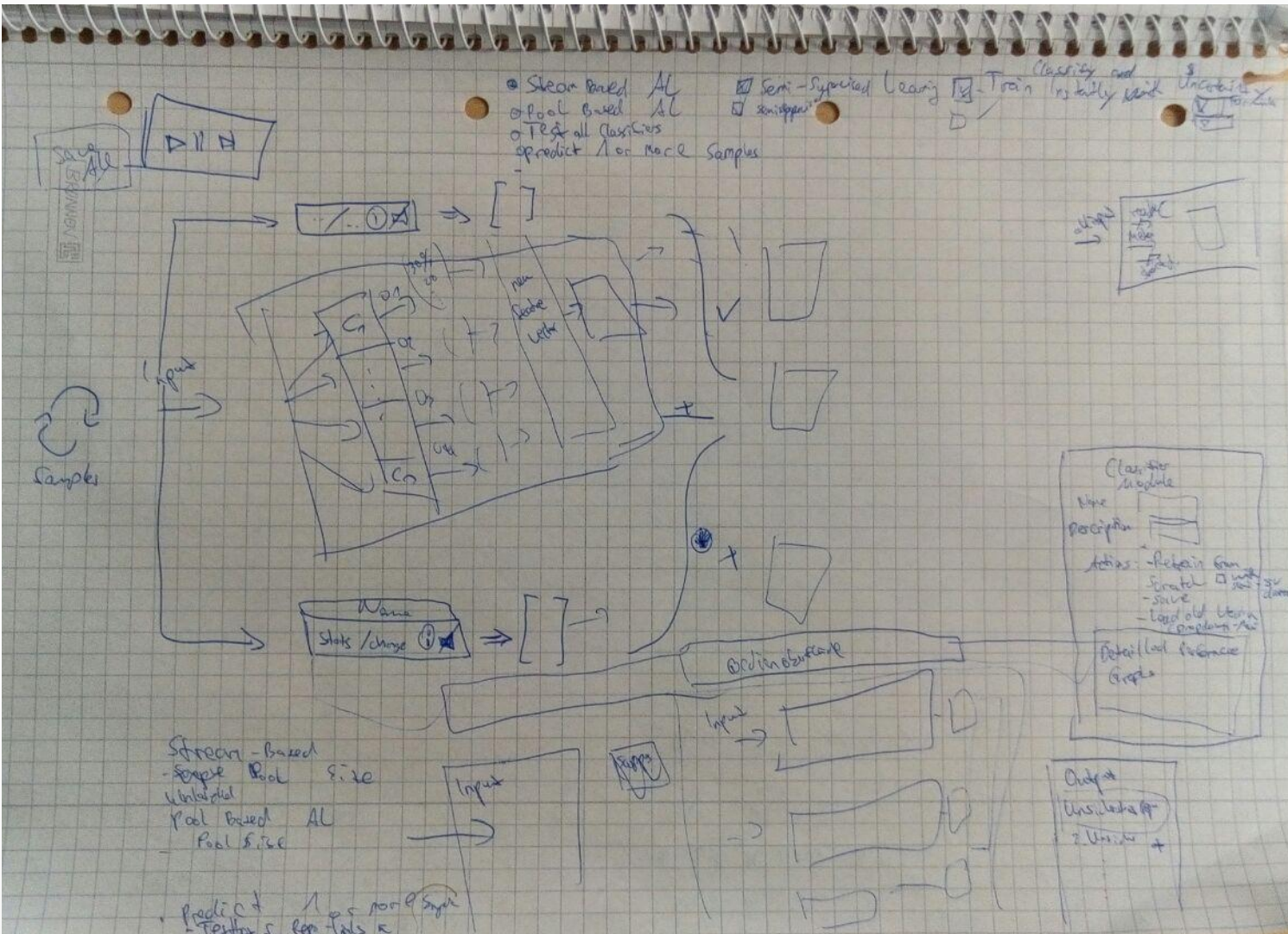
- **66 %** FScore M mittels **Ensemble** Klassifizierer
- **60 %** FScore M mittels **einzelner** Klassifizierer
- Problematisch bleibt die Unterscheidung von:
 - HW von DEV
 - DEV und nicht DEV
- Die besten Ergebnisse liefern:
 - DEV
 - EDU
 - DATA

Evaluation - Interpretation

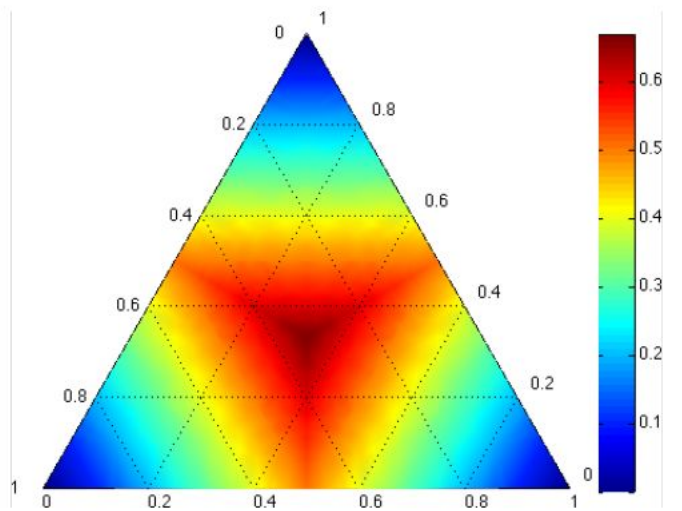
- Ergebnisse der Teams schwer zu vergleichen, da unterschiedliche Trainingsdaten und **unterschiedliche Interpretation** der Klassen
- auch unterschiedlichste Variationen und Klassifikatoren brachten keine Verbesserung => Möglichkeiten bei **Klassifikatoren ausgereizt**
- erwarten bessere Resultate mit höherer Anzahl **Trainingsdaten**
 - + zudem weitere **Konsistenzprüfung** der bisherigen Trainingsdaten
 - + bessere **Feature-Voranalyse**
- Für 10.000 - dimensionalen Inputvektor erstaunlich gute Ergebnisse, ggf. durch das **Active Learning**

Softwareentwurf

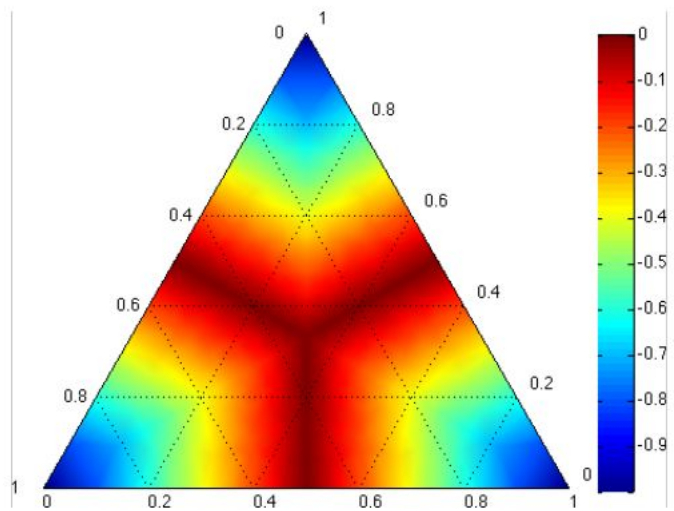
- Module als "Black Boxes"
- Enge Verknüpfung zwischen GUI und Logik - Schicht
- Model - View - Controller



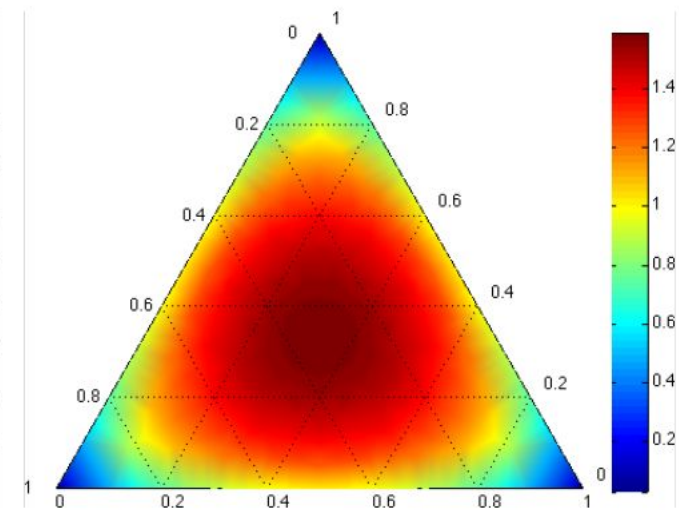
Unsicherheitsformeln



(a) least confident



(b) margin



(c) entropy

Quelle: Settles, Burr (2010), "Active Learning Literature Survey" Computer Sciences Technical Report 1648.

Libraries und Technologien

- Libraries:

- numpy
- scipy
- cherrypy / paste
- bottle
- keras
- sklearn
- nltk
- gensim
- pattern
- theano
- demjson
- Vue.JS

- Technologien:

- PHP
- Python
- MySQL
- JavaScript
- HTML
- CSS

Besonderheiten und Herausforderungen

Problem

- Unklare **Klassenbeschreibungen**
- **Ungleichgewicht** der Klassen
- Vergleich verschiedener **Algorithmen**
- **Hochdimensionaler** unbekannter Feature-Raum

Unser Ansatz

- “**Saubere**” Trainingsdaten
- **Active Learning**
- **Modularisierung, Ensemble Learning**
- **Feature engineering, Active Learning**