Seminar Klassifikation & Clustering

Klassifikation mit

Perzeptron

@CIS-LMU
Wintersemester 2022/23
Dr. Stefan Langer

19.12.2022
Clara Görlach
Xaver Krückl

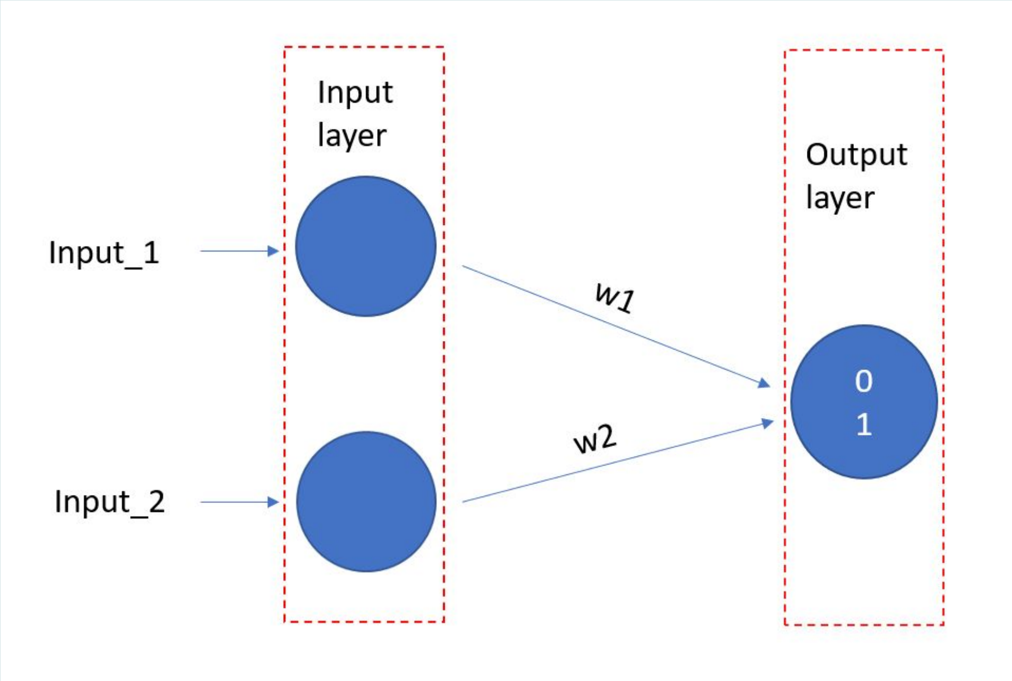
Gliederung

- Perzeptron - Algorithmus
- Stärken/Schwächen
- Verwendung
- Experimente
- Ergebnisse
- Fazit
- Quellen

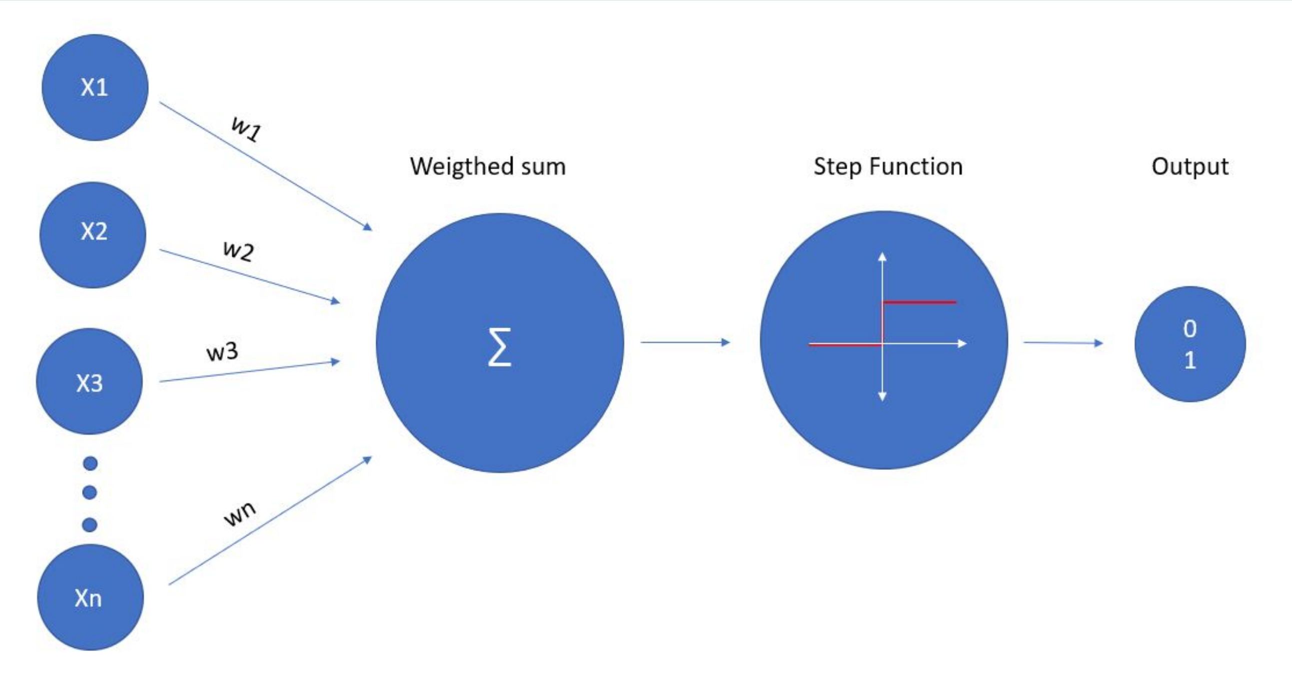
Perzeptron - Algorithmus

- ursprüngliches Modell von Frank Rosenblatt (1957)
- einfaches neuronales Netz
- Daten müssen linear trennbar sein
- überwachtes Lernen
- binäre Ausgabe

Perzeptron - Algorithmus



Perzeptron - Algorithmus



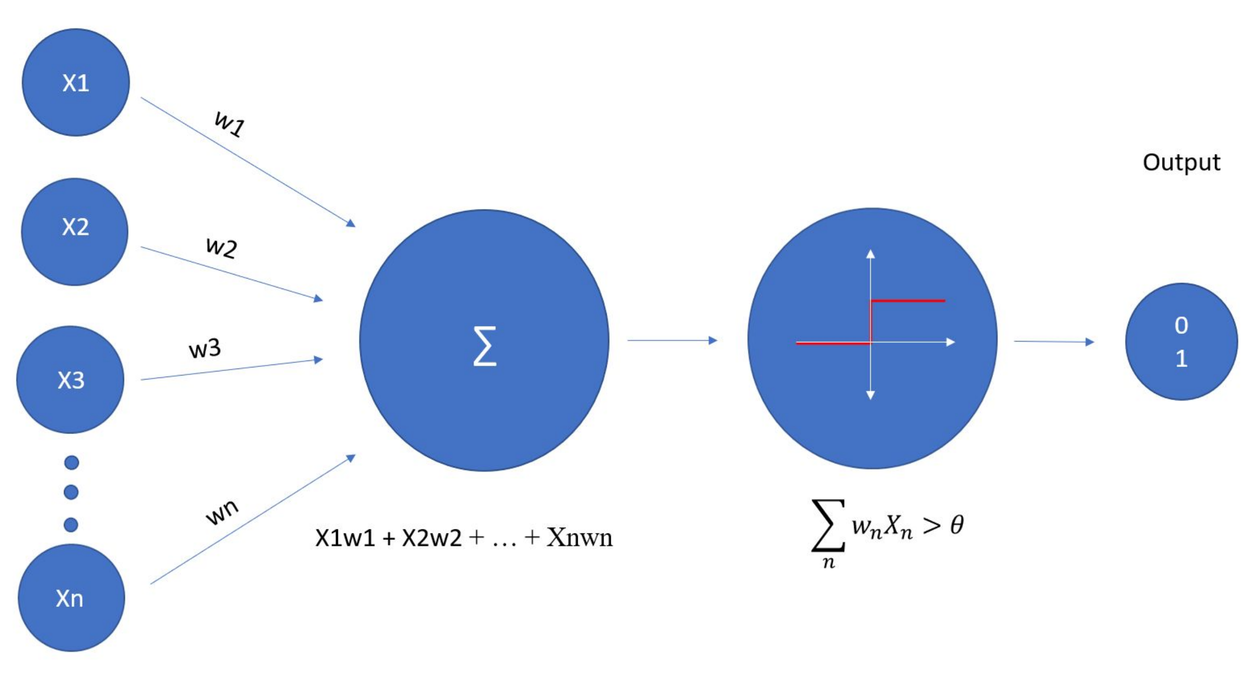
Perzeptron - Algorithmus

- Gewichte mit zufälligem Wert initialisieren
- Lernrate festlegen (optional)
- Schwellenwert θ oder Bias Neuron festlegen

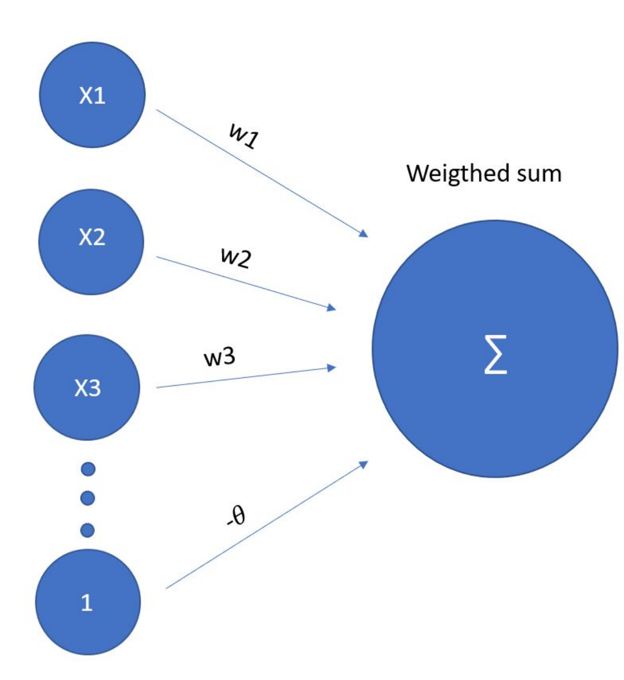
$$s = \sum_{i=1}^{n} \underbrace{x_i}_{\in\{0,1\}} \cdot \underbrace{w_i}_{\in\mathbb{R}}$$

$$o = f(s) = \begin{cases} 1 & \text{falls } s \geq \theta, \\ 0 & \text{sonst.} \end{cases}$$

Perzeptron - Algorithmus



Perzeptron - Algorithmus



Perzeptron - Gewichts-Update

- Label und vorhergesagtes Label stimmen überein -> keine Änderung des Gewichts
- vorhergesagtes Label = True und korrektes Label = False -> Gewicht wird reduziert; error = 1
- vorhergesagtes Label = False und korrektes Label = True -> Gewicht wird erhöht; error = -1

Update für das Gewicht wj vom Input xj:

*wj = wj - error * xj*

Stärken

- mit geringem Aufwand in verschiedene Programmiersprachen übersetzbar
- leicht verständlich (Einstieg in neuronale Netze)
- Daten schnell binär teilbar (wenn linear trennbar)
  -> schnelle Entscheidungsfindung

Schwächen

- Leistungsfähigkeit hängt sehr von Input ab -> schlechter Input, schlechtes Ergebnis
- Daten müssen linear trennbar sein
- der Output ist binär

Verwendung

- Klassifikation
- Bilderkennung
- Stimmungsanalyse
- Spracherkennung

Experimente - Settings

- Implementierungen:
  - manuell (Python Standard Bibliotheken)
  - Scikit-Learn
- Manuell - Feature Auswahl & Hyperparameter
  - word-count basiert
  - Anzahl Iterationen
  - top_n Features
- sklearn - Feature Auswahl:
  - CountVectorizer (BOW)
  - TfidfVectorizer (normalisierter BOW)
  - Word/Character N-Grams (“besserer” BOW)

Experimente - Settings

- sklearn - Hyperparameter Perceptron (Auswahl):
  - *penalty* (Regularisierung) und *alpha* (Faktor Regularisierung) optional
  - *max_iter* (vgl. n Iterationen) und *shuffle* (Mischen der Daten)
  - *early_stopping* und *n_iter_no_change* (Abbruch Iterations)
  - *eta0* (Lernrate) optional
- GridSearch
  - Versuch, einige der oben genannten Parameter zu Optimieren
  - Ergebnis: default-Werte lieferten beste Ergebnisse
- Multiclass Klassifikation
  - OneVsRest (OneVsAll) - ein Klassifikator pro Klasse gegen Rest
  - OneVsOne - jede Klasse gegen jede, prediction = bestes Ergebnis
  - OneVsRest schneller aufgrund geringerer Komplexität

  (- Multilayer Perceptron MLP)

Ergebnisse Sentiment-Daten

<u>**manuelle Implementierung:**</u>

- 100 Iterationen
- Top-N Features = None (hier für Vergleichbarkeit mit sklearn Implementierung):

Accuracy: 89.12
Precision: 89.13
Recall: 89.11
F1-Score: 89.12

-> Ergebnisse ähnlich zu sklearn Perceptron mit default settings (Ergebnisse folgen)
-> aber: deutlich längere Trainingszeit!

- 100 Iterationen, Top_N Features = 50:

Accuracy: 64.25
Precision: 64.52
Recall: 64.27
F1-Score: 64.39

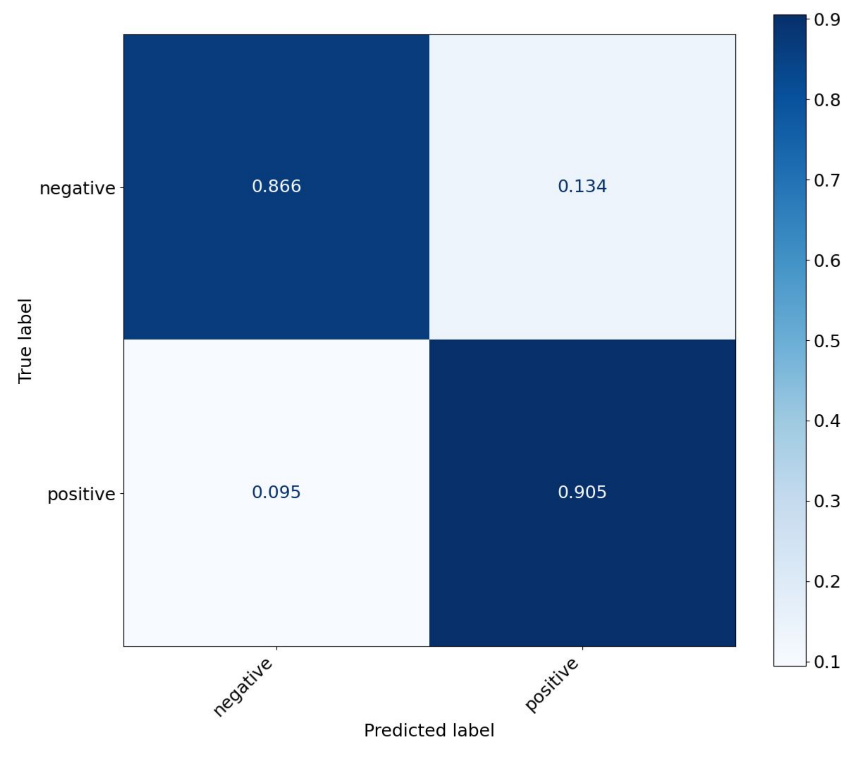
Ergebnisse Sentiment-Daten

<u>Scikit-Learn Implementierung:</u>

<u>CountVectorizer</u>

Perceptron - default settings

	precision	recall	f1-score	support
negative	0.901	0.866	0.883	4985
positive	0.872	0.905	0.888	5015
accuracy			0.886	10000
macro avg	0.886	0.886	0.886	10000
weighted avg	0.886	0.886	0.886	10000



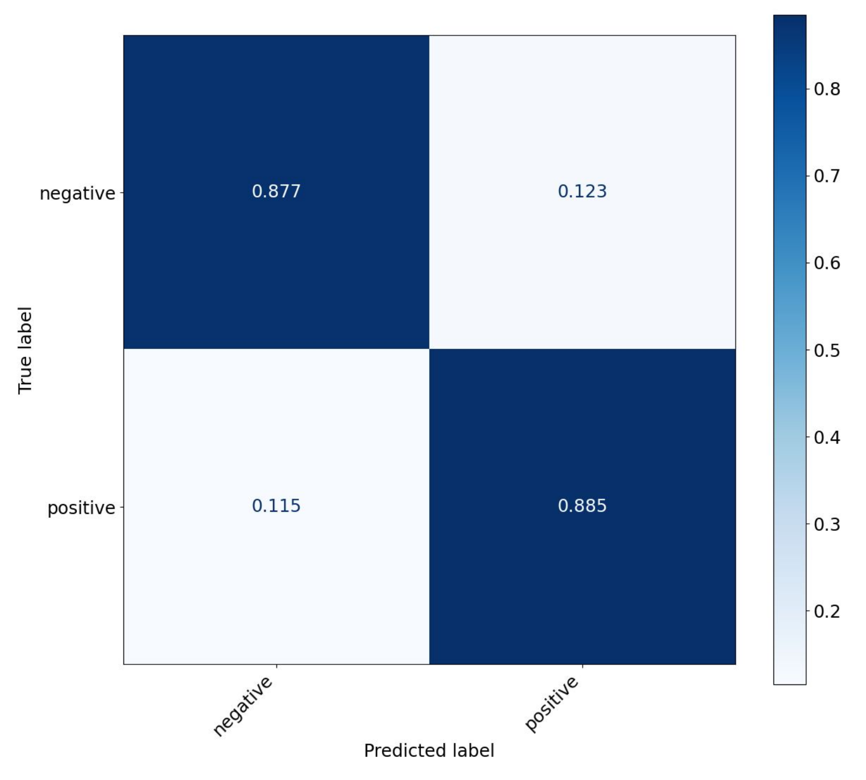
Ergebnisse Sentiment-Daten

Scikit-Learn Implementierung:

TfidfVectorizer

Perceptron - default settings

	precision	recall	f1-score	support
negative	0.883	0.877	0.880	4985
positive	0.878	0.885	0.881	5015
accuracy			0.881	10000
macro avg	0.881	0.881	0.881	10000
weighted avg	0.881	0.881	0.881	10000



Ergebnisse Sentiment-Daten

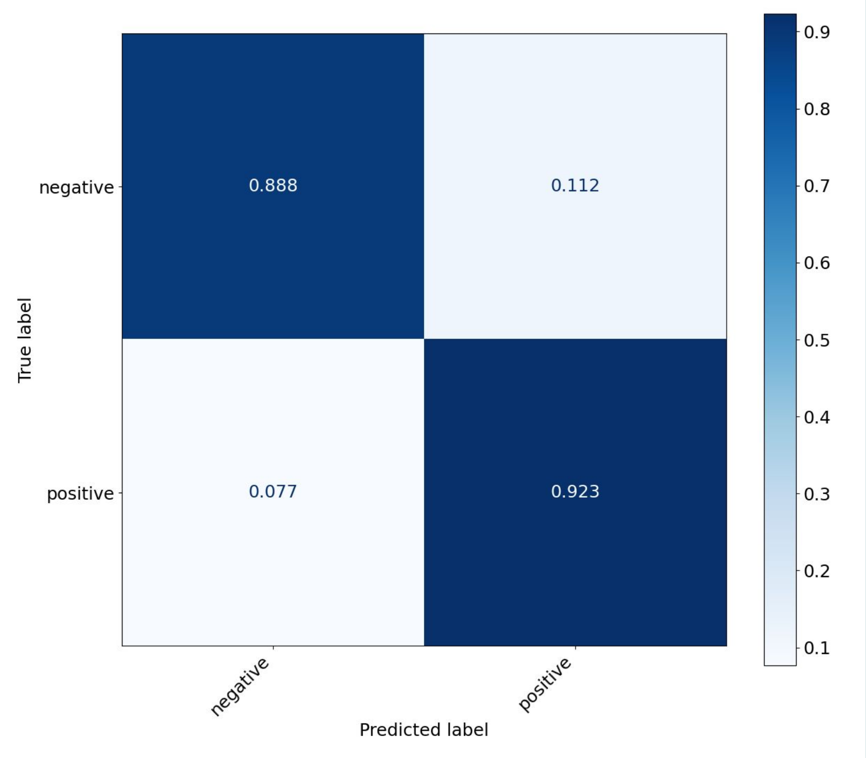
Scikit-Learn Implementierung:

CountVectorizer mit Character N-Grams

Perceptron - default settings

analyzer='char', ngram_range = (7,7)

	precision	recall	f1-score	support
negative	0.920	0.888	0.904	4985
positive	0.892	0.923	0.908	5015
accuracy			0.906	10000
macro avg	0.906	0.906	0.906	10000
weighted avg	0.906	0.906	0.906	10000



Ergebnisse Sentiment-Daten

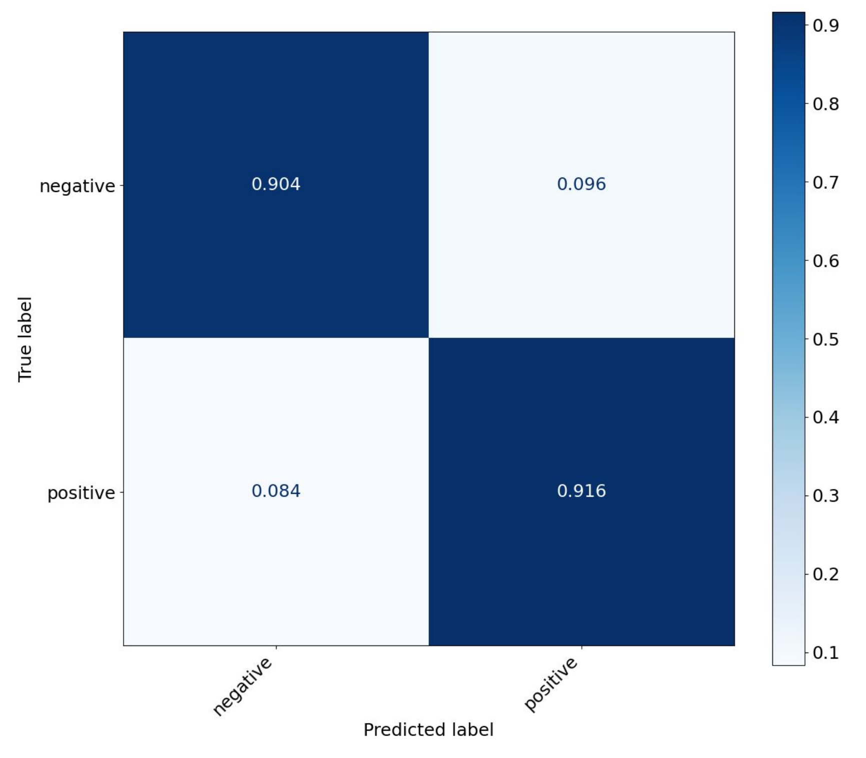
Scikit-Learn Implementierung:

TfidfVectorizer mit Character N-Grams

Perceptron - default settings

analyzer='char', ngram_range = (6,6)

	precision	recall	f1-score	support
negative	0.915	0.904	0.909	4985
positive	0.905	0.916	0.911	5015
accuracy			0.910	10000
macro avg	0.910	0.910	0.910	10000
weighted avg	0.910	0.910	0.910	10000



bestes Ergebnis

Briefsammlung - Autoren und Sprachen

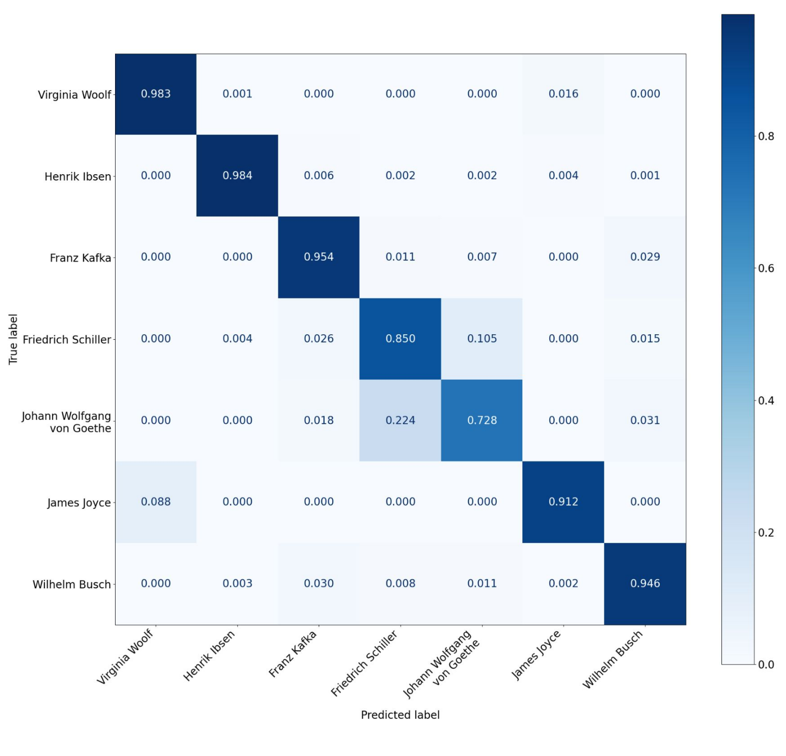
Scikit-Learn Implementierung:

- Multiclass Klassifikation:
  -> OneVsRest Classifier und OneVsOne Classifier
- default Parameter
- darin: Perceptron Classifier mit default Parametern
- Perceptron - Features: CountVectorizer / TfidfVectorizer
  - Input = text
  - Wort Level / Character N-Grams
- Exkurs: Multiclass Klassifikation mit Multilayer Perceptron
  -> default Parameter
  -> Features: CountVectorizer(word level)

Ergebnisse Briefsammlung - Autoren

OneVsRest(Perceptron(CountVectorizer(words))):

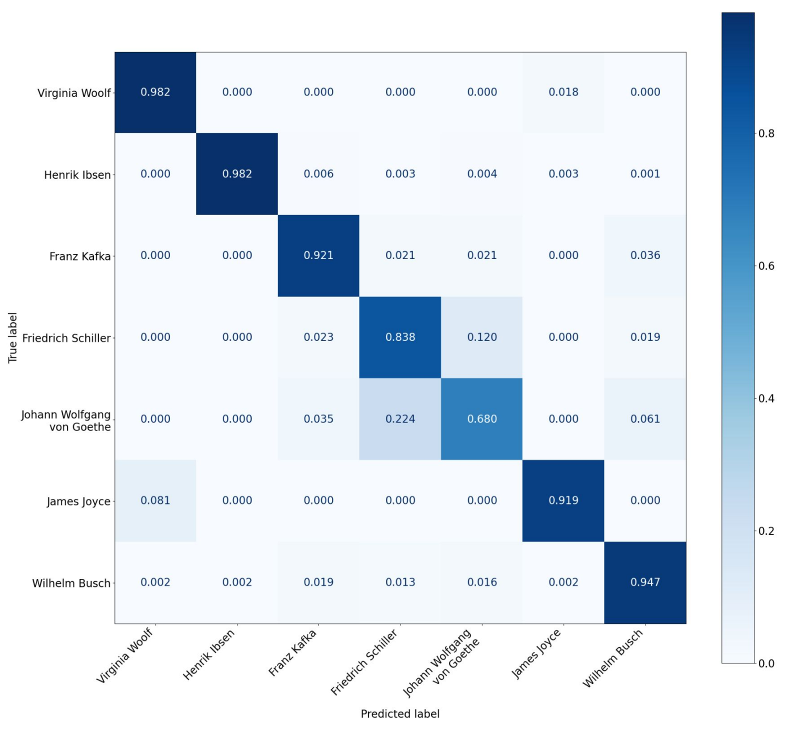
	precision	recall	f1-score	support
Virginia Woolf	0.969	0.983	0.976	1901
Henrik Ibsen	0.995	0.984	0.990	897
Franz Kafka	0.884	0.954	0.918	280
Friedrich Schiller	0.787	0.850	0.817	266
Johann Wolfgang von Goethe	0.810	0.728	0.767	228
James Joyce	0.945	0.912	0.928	682
Wilhelm Busch	0.967	0.946	0.956	627
accuracy			0.948	4881
macro avg	0.908	0.908	0.907	4881
weighted avg	0.948	0.948	0.948	4881



Ergebnisse Briefsammlung - Autoren

OneVsOne(Perceptron(CountVectorizer(words))):

	precision	recall	f1-score	support
Virginia Woolf	0.971	0.982	0.976	1901
Henrik Ibsen	0.999	0.982	0.990	897
Franz Kafka	0.893	0.921	0.907	280
Friedrich Schiller	0.766	0.838	0.801	266
Johann Wolfgang von Goethe	0.749	0.680	0.713	228
James Joyce	0.943	0.919	0.931	682
Wilhelm Busch	0.952	0.947	0.950	627
accuracy			0.943	4881
macro avg	0.896	0.896	0.895	4881
weighted avg	0.944	0.943	0.943	4881



Ergebnisse Briefsammlung - Autoren

OneVsRest(Perceptron(TfidfVectorizer(words))):

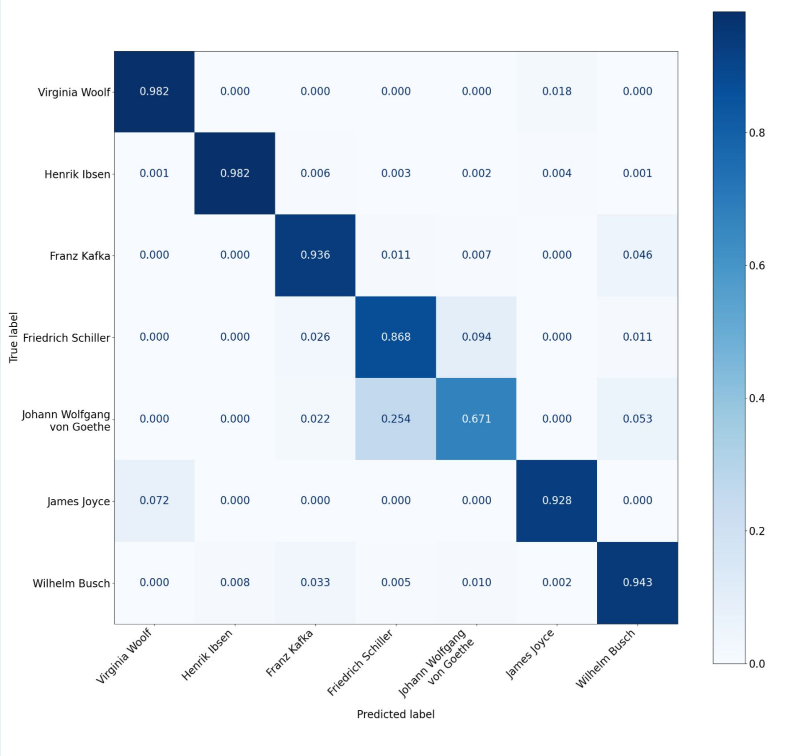
	precision	recall	f1-score	support
Virginia Woolf	0.981	0.981	0.981	1901
Henrik Ibsen	0.992	0.987	0.989	897
Franz Kafka	0.920	0.946	0.933	280
Friedrich Schiller	0.840	0.831	0.836	266
Johann Wolfgang von Goethe	0.778	0.768	0.773	228
James Joyce	0.943	0.947	0.945	682
Wilhelm Busch	0.959	0.959	0.959	627
accuracy			0.954	4881
macro avg	0.916	0.917	0.916	4881
weighted avg	0.954	0.954	0.954	4881

bestes Ergebnis word level

Ergebnisse Briefsammlung - Autoren

<u>OneVsOne(Perceptron(TfidfVectorizer(words))):</u>

	precision	recall	f1-score	support
Virginia Woolf	0.974	0.982	0.978	1901
Henrik Ibsen	0.994	0.982	0.988	897
Franz Kafka	0.873	0.936	0.903	280
Friedrich Schiller	0.775	0.868	0.819	266
Johann Wolfgang von Goethe	0.814	0.671	0.736	228
James Joyce	0.942	0.928	0.935	682
Wilhelm Busch	0.953	0.943	0.948	627
accuracy			0.946	4881
macro avg	0.904	0.901	0.901	4881
weighted avg	0.946	0.946	0.946	4881



Ergebnisse Briefsammlung - Autoren

OneVsRest/OneVsOne(Perceptron(CountVectorizer(char_ngrams)):

- nochmals leichte Verbesserung von Accurracy/F1 Score
- beste n_gram Größen: 5 (6)
- kaum Verbesserung mit word_ngrams

OneVsRest/OneVsOne(Perceptron(TfidfVectorizer(char_ngrams)))

- nochmals leichte Verbesserung von Accurracy/F1 Score
- beste n_gram Größen: 5 (6)
- kaum Verbesserung mit word_ngrams

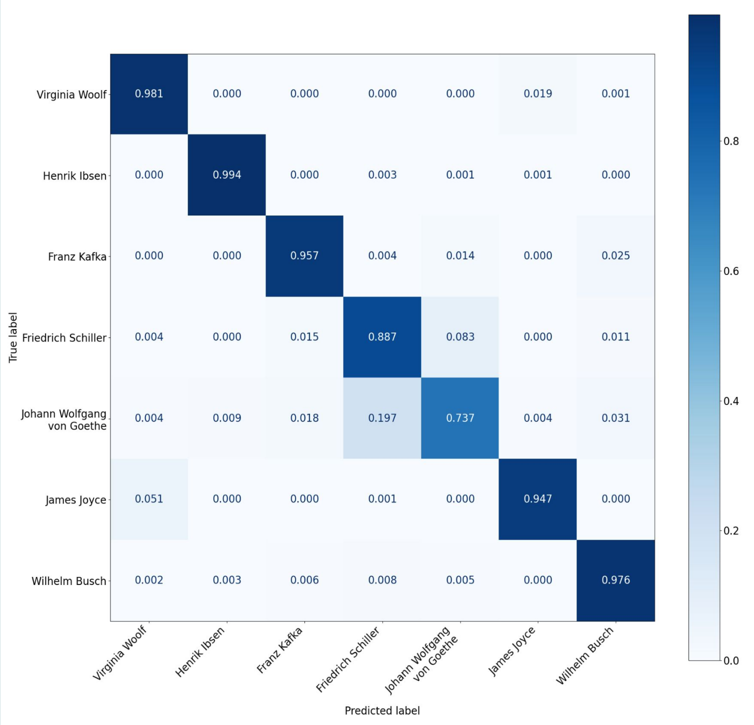
Ergebnisse Briefsammlung - Autoren

<u>OneVsRest(Perceptron(TfidfVectorizer(char5_ngrams)))</u>

Verbesserung durch char_n-grams?

bestes Ergebnis:

	precision	recall	f1-score	support
Virginia Woolf	0.980	0.981	0.980	1901
Henrik Ibsen	0.996	0.994	0.995	897
Franz Kafka	0.957	0.957	0.957	280
Friedrich Schiller	0.811	0.887	0.847	266
Johann Wolfgang von Goethe	0.848	0.737	0.789	228
James Joyce	0.944	0.947	0.946	682
Wilhelm Busch	0.971	0.976	0.974	627
accuracy			0.960	4881
macro avg	0.930	0.926	0.927	4881
weighted avg	0.960	0.960	0.960	4881

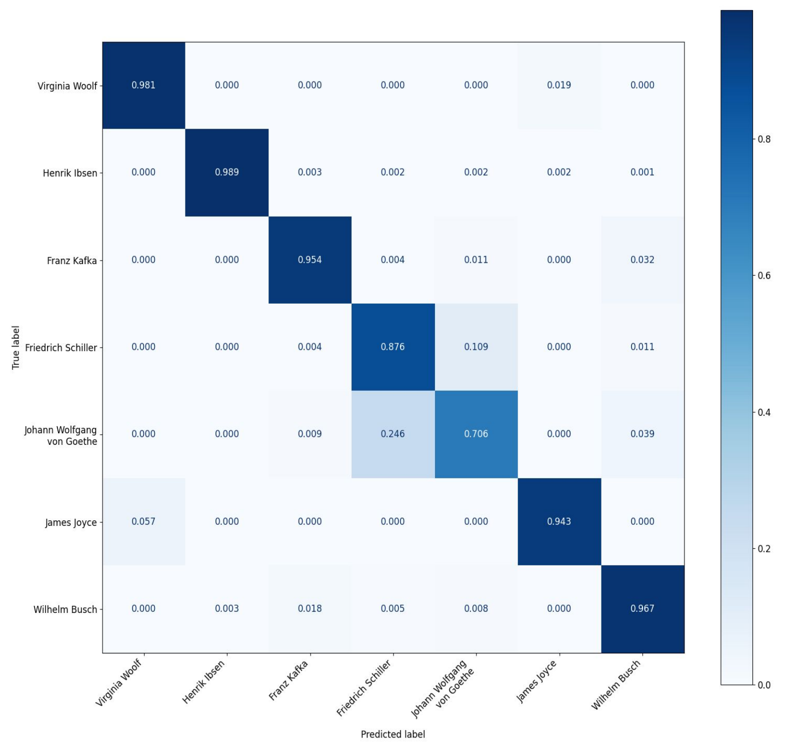


Ergebnisse Briefsammlung - Autoren

Exkurs: Multilayer Perceptron

MLP(CountVectorizer(words)):

	precision	recall	f1-score	support
Virginia Woolf	0.980	0.981	0.980	1901
Henrik Ibsen	0.998	0.989	0.993	897
Franz Kafka	0.940	0.954	0.947	280
Friedrich Schiller	0.790	0.876	0.831	266
Johann Wolfgang von Goethe	0.805	0.706	0.752	228
James Joyce	0.943	0.943	0.943	682
Wilhelm Busch	0.965	0.967	0.966	627
accuracy			0.955	4881
macro avg	0.917	0.916	0.916	4881
weighted avg	0.955	0.955	0.955	4881

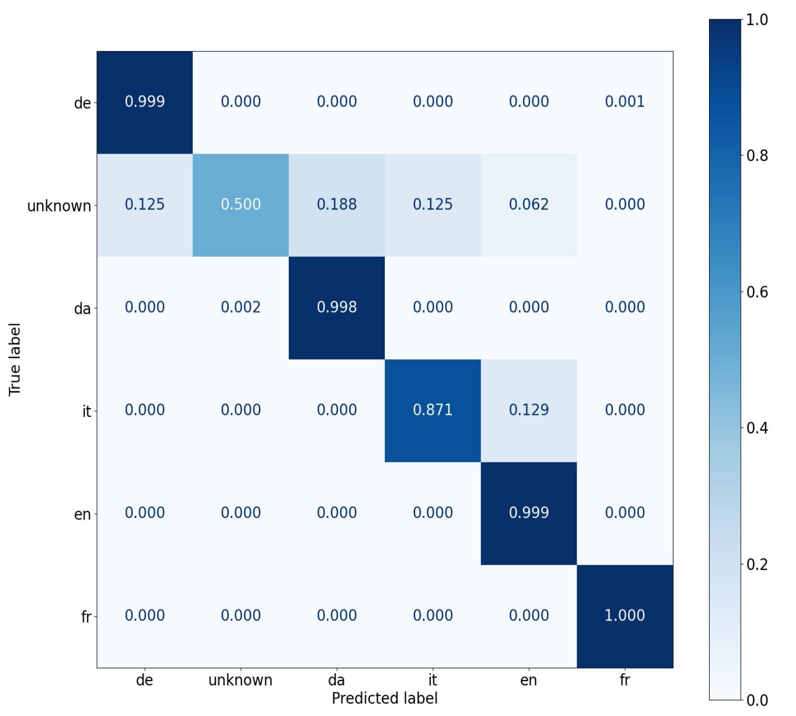


Ergebnisse ähnlich mit OneVsRest

Ergebnisse Briefsammlung - Sprachen

<u>OneVsRest(Perceptron(CountVectorizer(words))):</u>

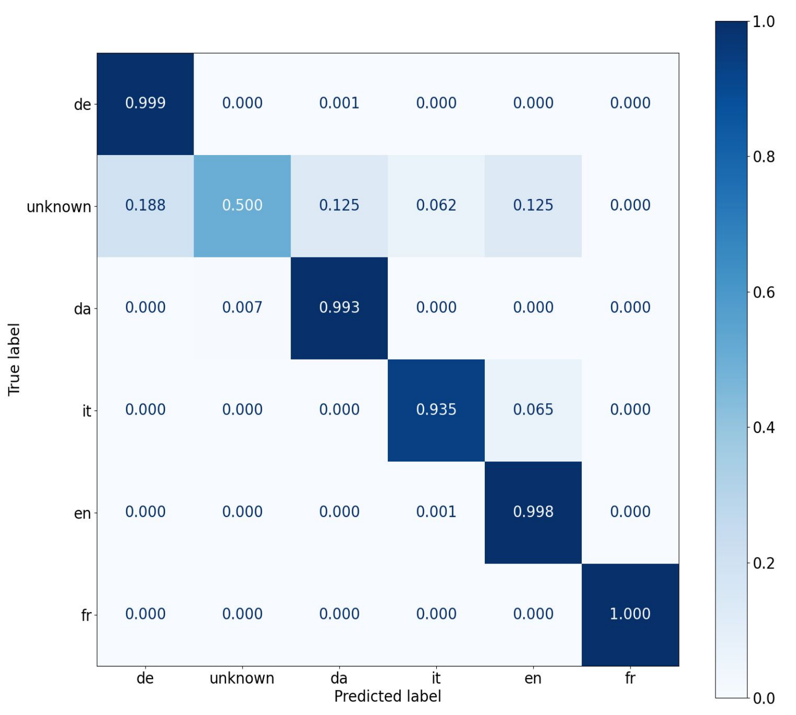
	precision	recall	f1-score	support
de	0.999	0.999	0.999	1443
unknown	0.727	0.500	0.593	16
da	0.996	0.998	0.997	838
it	0.900	0.871	0.885	31
en	0.998	0.999	0.999	2517
fr	0.973	1.000	0.986	36
accuracy			0.997	4881
macro avg	0.932	0.895	0.910	4881
weighted avg	0.996	0.997	0.996	4881



Ergebnisse Briefsammlung - Sprachen

OneVsOne(Perceptron(CountVectorizer(words))):

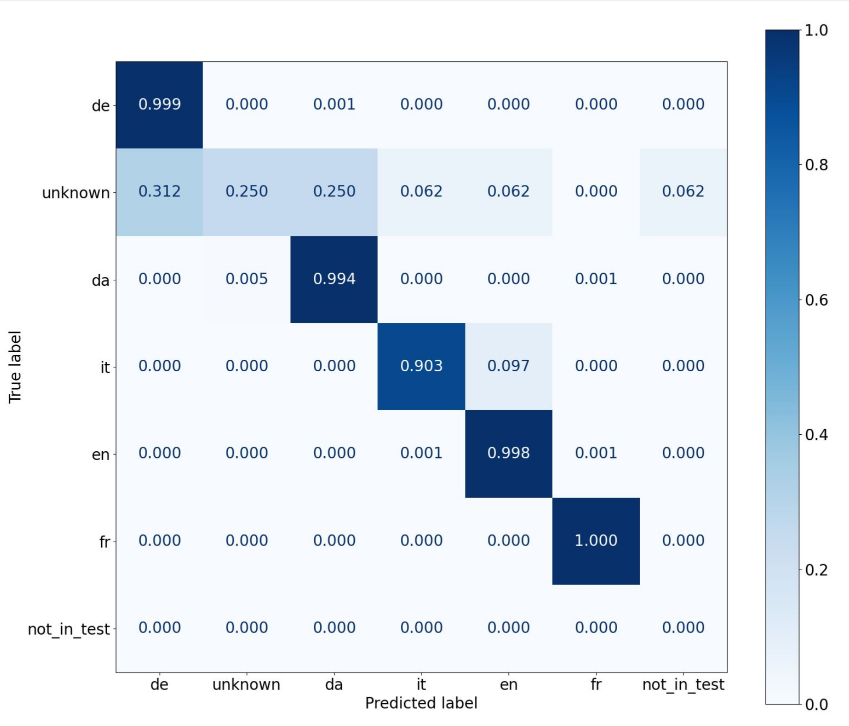
	precision	recall	f1-score	support
de	0.998	0.999	0.999	1443
unknown	0.533	0.500	0.516	16
da	0.996	0.993	0.995	838
it	0.879	0.935	0.906	31
en	0.998	0.998	0.998	2517
fr	1.000	1.000	1.000	36
accuracy			0.996	4881
macro avg	0.901	0.904	0.902	4881
weighted avg	0.996	0.996	0.996	4881



Ergebnisse Briefsammlung - Sprachen

OneVsRest(Perceptron(TfidfVectorizer(words))):

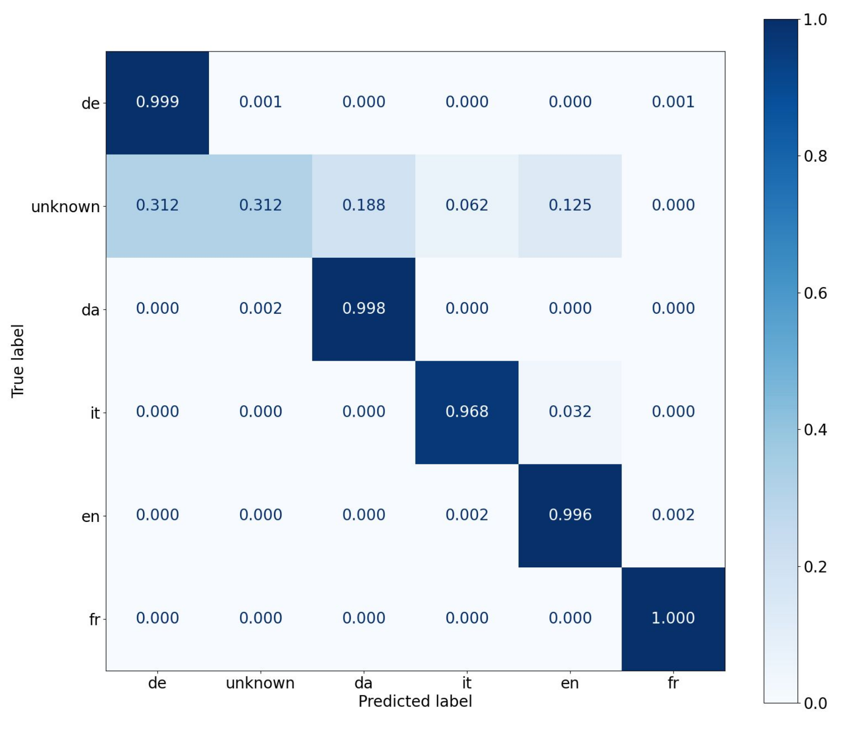
	precision	recall	f1-score	support
de	0.997	0.999	0.998	1443
unknown	0.444	0.250	0.320	16
da	0.994	0.994	0.994	838
it	0.903	0.903	0.903	31
en	0.998	0.998	0.998	2517
fr	0.923	1.000	0.960	36
not_in_test	0.000	0.000	0.000	0
accuracy			0.994	4881
macro avg	0.751	0.735	0.739	4881
weighted avg	0.994	0.994	0.994	4881



Ergebnisse Briefsammlung - Sprachen

OneVsOne(Perceptron(TfidfVectorizer(words))):

	precision	recall	f1-score	support
de	0.997	0.999	0.998	1443
unknown	0.556	0.312	0.400	16
da	0.996	0.998	0.997	838
it	0.857	0.968	0.909	31
en	0.999	0.996	0.998	2517
fr	0.878	1.000	0.935	36
accuracy			0.995	4881
macro avg	0.880	0.879	0.873	4881
weighted avg	0.994	0.995	0.995	4881



Ergebnisse Briefsammlung - Sprachen

OneVsRest/OneVsOne(Perceptron(CountVectorizer(char_ngrams)):

- keine relevante Verbesserung von Accurracy/F1 Score
- Maße bereits sehr hoch
- eher Verschlechterung mit word_ngrams

OneVsRest/OneVsOne(Perceptron(TfidfVectorizer(char_ngrams)))

- keine relevante Verbesserung von Accurracy/F1 Score
- Maße bereits sehr hoch
- eher Verschlechterung mit word_ngrams

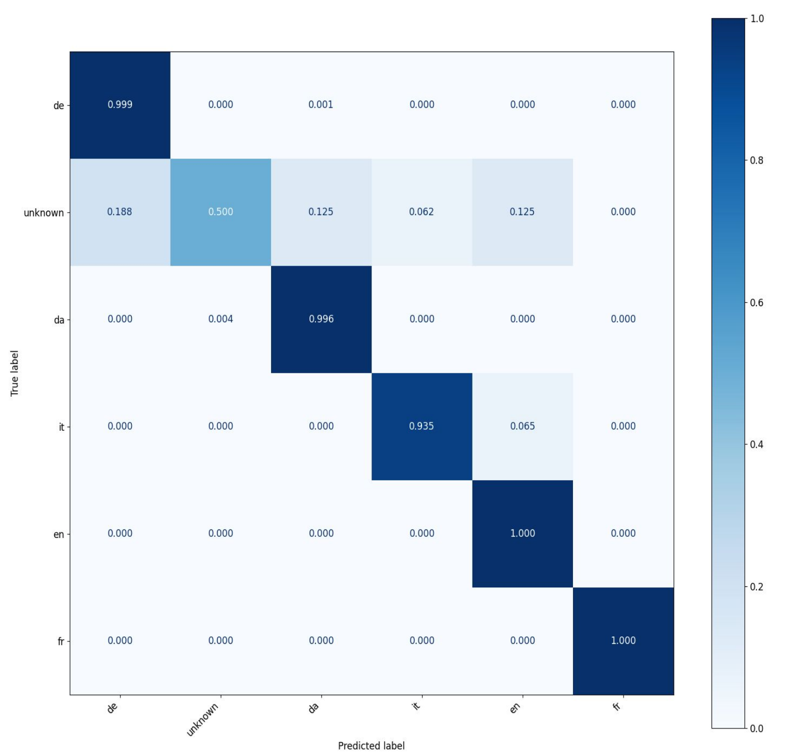
Ergebnisse Briefsammlung - Sprachen

Exkurs: Multilayer Perceptron

MLP(CountVectorizer(words)):

	precision	recall	f1-score	support
de	0.998	0.999	0.999	1443
unknown	0.727	0.500	0.593	16
da	0.996	0.996	0.996	838
it	0.935	0.935	0.935	31
en	0.998	1.000	0.999	2517
fr	1.000	1.000	1.000	36
accuracy			0.997	4881
macro avg	0.943	0.905	0.920	4881
weighted avg	0.997	0.997	0.997	4881

Ergebnisse im selben Rahmen



Huffington-Post Daten

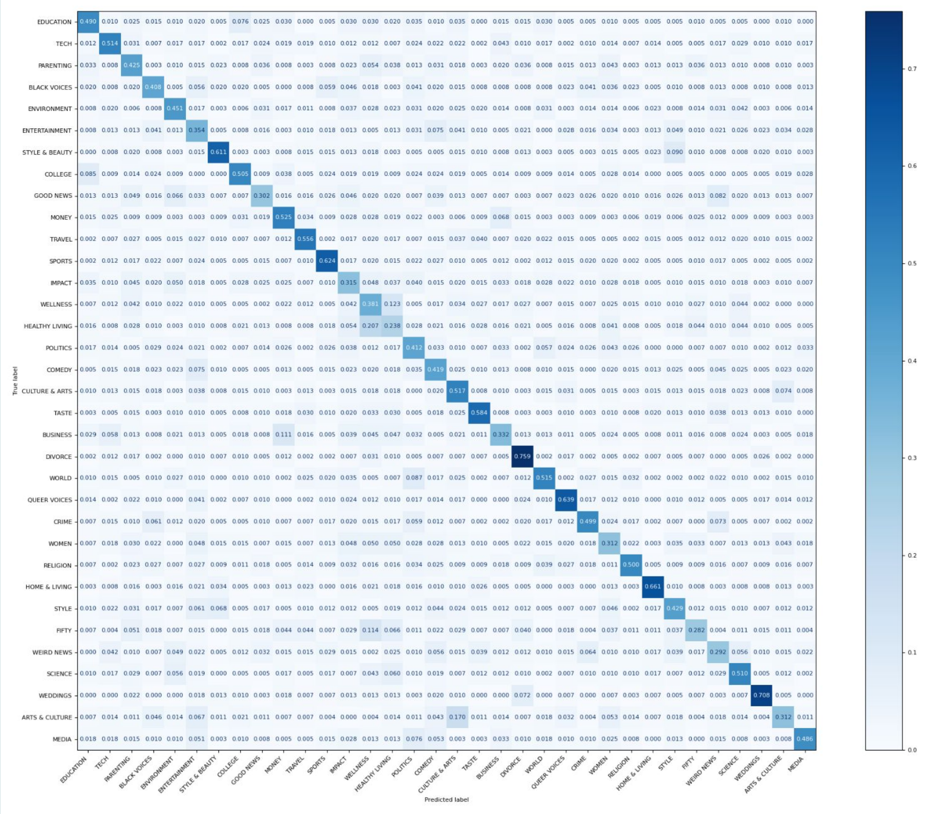
Scikit-Learn Implementierung:

- Multiclass Klassifikation:
  -> OneVsRest Classifier und OneVsOne Classifier
- default Parameter
- darin: Perceptron Classifier mit default Parametern
- Perceptron - Features: CountVectorizer / TfidfVectorizer (word level, input = headline + short description)
- Exkurs: Multiclass Klassifikation mit Multilayer Perceptron
  -> default Parameter
  -> Features: CountVectorizer(word level) / TfidfVectorizer(word level)

Ergebnisse Huffington-Post Daten

OneVsRest(Perceptron(CountVectorizer)):

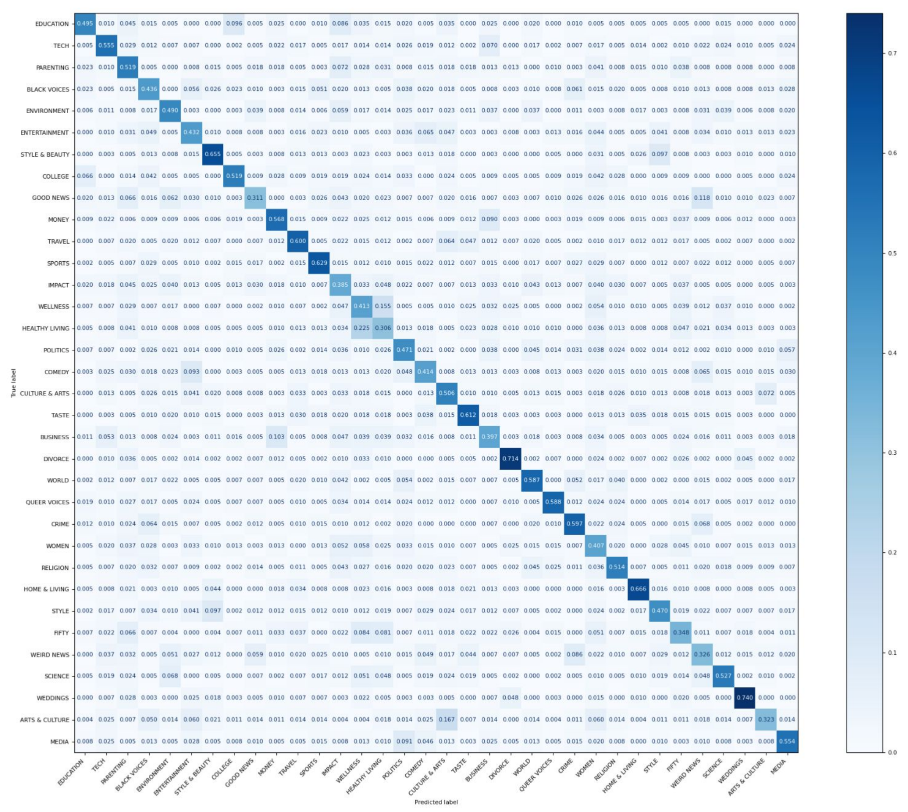
	precision	recall	f1-score	support
EDUCATION	0.394	0.490	0.437	198
TECH	0.554	0.514	0.534	416
PARENTING	0.387	0.425	0.405	391
BLACK VOICES	0.455	0.408	0.430	392
ENVIRONMENT	0.449	0.451	0.450	355
ENTERTAINMENT	0.297	0.354	0.323	387
STYLE & BEAUTY	0.666	0.611	0.637	391
COLLEGE	0.415	0.505	0.455	212
GOOD NEWS	0.364	0.302	0.330	305
MONEY	0.464	0.525	0.493	324
TRAVEL	0.603	0.556	0.578	405
SPORTS	0.607	0.624	0.615	410
IMPACT	0.292	0.315	0.303	400
WELLNESS	0.298	0.381	0.334	407
HEALTHY LIVING	0.231	0.238	0.234	386
POLITICS	0.369	0.412	0.389	420
COMEDY	0.358	0.419	0.386	399
CULTURE & ARTS	0.433	0.517	0.471	391
TASTE	0.620	0.584	0.602	397
BUSINESS	0.410	0.332	0.367	380
DIVORCE	0.624	0.759	0.685	419
WORLD	0.565	0.515	0.539	404
QUEER VOICES	0.598	0.639	0.618	415
CRIME	0.562	0.499	0.528	409
WOMEN	0.320	0.312	0.316	400
RELIGION	0.641	0.500	0.562	440
HOME & LIVING	0.676	0.661	0.668	386
STYLE	0.475	0.429	0.450	413
FIFTY	0.324	0.282	0.301	273
WEIRD NEWS	0.349	0.292	0.318	408
SCIENCE	0.504	0.510	0.507	414
WEDDINGS	0.722	0.708	0.715	400
ARTS & CULTURE	0.340	0.312	0.325	282
MEDIA	0.619	0.486	0.545	395
accuracy			0.471	12824
macro avg	0.470	0.467	0.466	12824
weighted avg	0.477	0.471	0.472	12824



Ergebnisse Huffington-Post Daten

<u>OneVsOne(Perceptron(CountVectorizer)):</u>

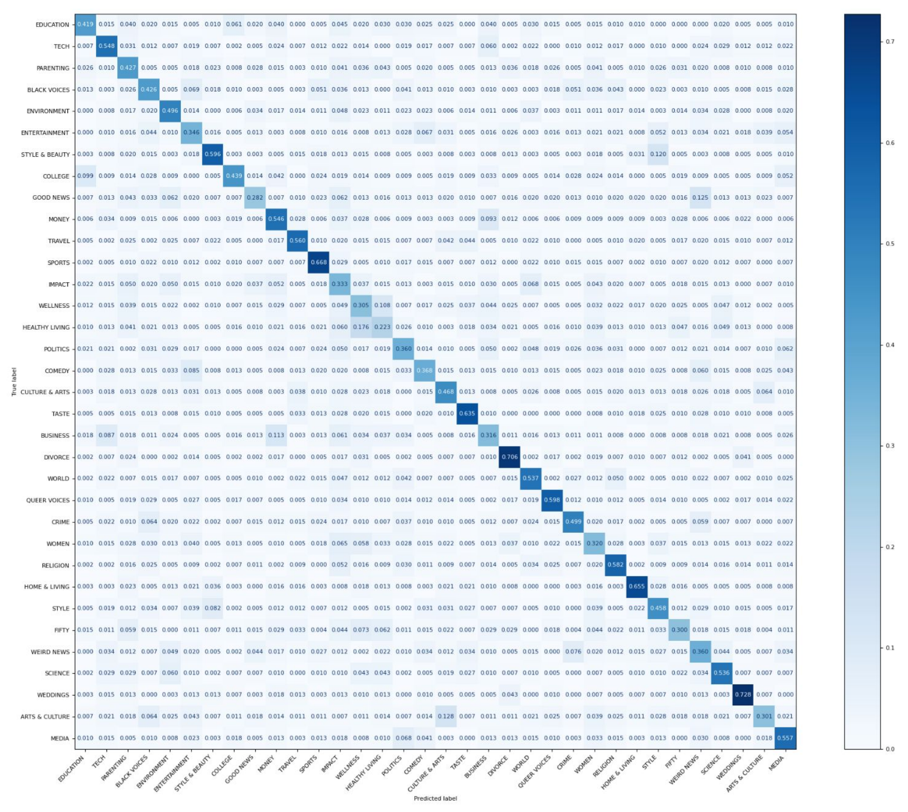
	precision	recall	f1-score	support
EDUCATION	0.505	0.495	0.500	198
TECH	0.573	0.555	0.564	416
PARENTING	0.422	0.519	0.466	391
BLACK VOICES	0.425	0.436	0.431	392
ENVIRONMENT	0.472	0.490	0.481	355
ENTERTAINMENT	0.416	0.432	0.424	387
STYLE & BEAUTY	0.637	0.655	0.646	391
COLLEGE	0.537	0.519	0.528	212
GOOD NEWS	0.411	0.311	0.354	305
MONEY	0.524	0.568	0.545	324
TRAVEL	0.600	0.600	0.600	405
SPORTS	0.634	0.629	0.632	410
IMPACT	0.324	0.385	0.352	400
WELLNESS	0.324	0.413	0.363	407
HEALTHY LIVING	0.294	0.306	0.300	386
POLITICS	0.449	0.471	0.460	420
COMEDY	0.434	0.414	0.424	399
CULTURE & ARTS	0.446	0.506	0.474	391
TASTE	0.626	0.612	0.619	397
BUSINESS	0.405	0.397	0.401	380
DIVORCE	0.767	0.714	0.739	419
WORLD	0.603	0.587	0.595	404
QUEER VOICES	0.733	0.588	0.652	415
CRIME	0.587	0.597	0.592	409
WOMEN	0.335	0.407	0.368	400
RELIGION	0.585	0.514	0.547	440
HOME & LIVING	0.698	0.666	0.682	386
STYLE	0.533	0.470	0.499	413
FIFTY	0.304	0.348	0.324	273
WEIRD NEWS	0.344	0.326	0.335	408
SCIENCE	0.614	0.527	0.567	414
WEDDINGS	0.733	0.740	0.736	400
ARTS & CULTURE	0.479	0.323	0.386	282
MEDIA	0.597	0.554	0.575	395
accuracy			0.507	12824
macro avg	0.511	0.502	0.505	12824
weighted avg	0.515	0.507	0.509	12824



Ergebnisse Huffington-Post Daten

OneVsRest(Perceptron(TfidfVectorizer)):

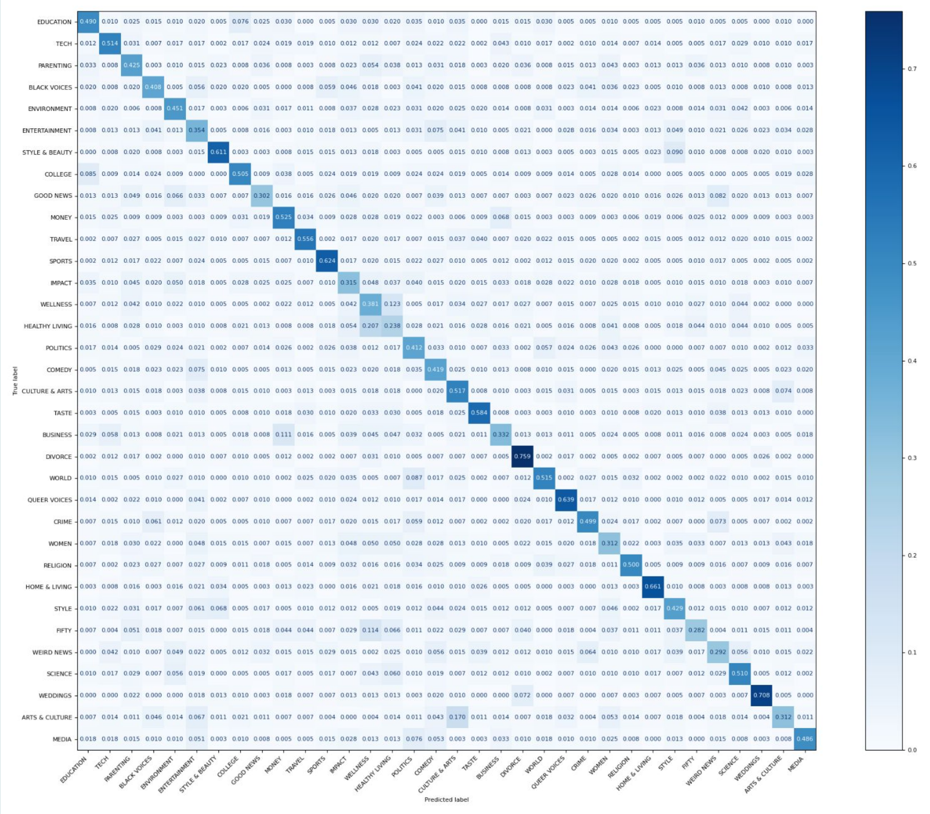
	precision	recall	f1-score	support
EDUCATION	0.426	0.419	0.422	198
TECH	0.529	0.548	0.538	416
PARENTING	0.389	0.427	0.407	391
BLACK VOICES	0.402	0.426	0.414	392
ENVIRONMENT	0.448	0.496	0.471	355
ENTERTAINMENT	0.346	0.346	0.346	387
STYLE & BEAUTY	0.631	0.596	0.613	391
COLLEGE	0.449	0.439	0.444	212
GOOD NEWS	0.382	0.282	0.325	305
MONEY	0.444	0.546	0.490	324
TRAVEL	0.622	0.560	0.590	405
SPORTS	0.617	0.668	0.642	410
IMPACT	0.256	0.333	0.289	400
WELLNESS	0.288	0.305	0.296	407
HEALTHY LIVING	0.261	0.223	0.240	386
POLITICS	0.407	0.360	0.382	420
COMEDY	0.434	0.368	0.398	399
CULTURE & ARTS	0.473	0.468	0.470	391
TASTE	0.625	0.635	0.630	397
BUSINESS	0.338	0.316	0.327	380
DIVORCE	0.653	0.706	0.679	419
WORLD	0.520	0.537	0.529	404
QUEER VOICES	0.633	0.598	0.615	415
CRIME	0.585	0.499	0.538	409
WOMEN	0.327	0.320	0.323	400
RELIGION	0.565	0.582	0.573	440
HOME & LIVING	0.691	0.655	0.673	386
STYLE	0.447	0.458	0.452	413
FIFTY	0.325	0.300	0.312	273
WEIRD NEWS	0.342	0.360	0.351	408
SCIENCE	0.534	0.536	0.535	414
WEDDINGS	0.700	0.728	0.713	400
ARTS & CULTURE	0.373	0.301	0.333	282
MEDIA	0.498	0.557	0.526	395
accuracy			0.474	12824
macro avg	0.469	0.468	0.467	12824
weighted avg	0.475	0.474	0.473	12824



Ergebnisse Huffington-Post Daten

OneVsOne(Perceptron(TfidfVectorizer)): bestes Ergebnis

	precision	recall	f1-score	support
EDUCATION	0.549	0.510	0.529	198
TECH	0.629	0.591	0.610	416
PARENTING	0.429	0.560	0.486	391
BLACK VOICES	0.455	0.452	0.453	392
ENVIRONMENT	0.530	0.541	0.536	355
ENTERTAINMENT	0.398	0.468	0.430	387
STYLE & BEAUTY	0.726	0.665	0.694	391
COLLEGE	0.576	0.519	0.546	212
GOOD NEWS	0.457	0.348	0.395	305
MONEY	0.578	0.574	0.576	324
TRAVEL	0.653	0.627	0.640	405
SPORTS	0.666	0.661	0.663	410
IMPACT	0.351	0.407	0.377	400
WELLNESS	0.362	0.415	0.387	407
HEALTHY LIVING	0.275	0.345	0.306	386
POLITICS	0.449	0.517	0.481	420
COMEDY	0.415	0.454	0.434	399
CULTURE & ARTS	0.478	0.491	0.484	391
TASTE	0.687	0.675	0.681	397
BUSINESS	0.395	0.416	0.405	380
DIVORCE	0.839	0.709	0.768	419
WORLD	0.634	0.587	0.609	404
QUEER VOICES	0.750	0.607	0.671	415
CRIME	0.600	0.592	0.596	409
WOMEN	0.374	0.450	0.409	400
RELIGION	0.651	0.552	0.598	440
HOME & LIVING	0.739	0.697	0.717	386
STYLE	0.533	0.528	0.530	413
FIFTY	0.338	0.348	0.343	273
WEIRD NEWS	0.364	0.390	0.376	408
SCIENCE	0.644	0.556	0.597	414
WEDDINGS	0.798	0.730	0.762	400
ARTS & CULTURE	0.505	0.344	0.409	282
MEDIA	0.599	0.580	0.589	395
accuracy			0.532	12824
macro avg	0.542	0.527	0.532	12824
weighted avg	0.546	0.532	0.537	12824



Ergebnisse Huffington-Post Daten

<u>OneVsRest/OneVsOne(Perceptron(CountVectorizer(char_ngrams)):</u>

- nur sehr geringe Verbesserung von Accurracy/F1 Score
- beste n_gram Größe: 5
- Verschlechterung mit word_ngrams

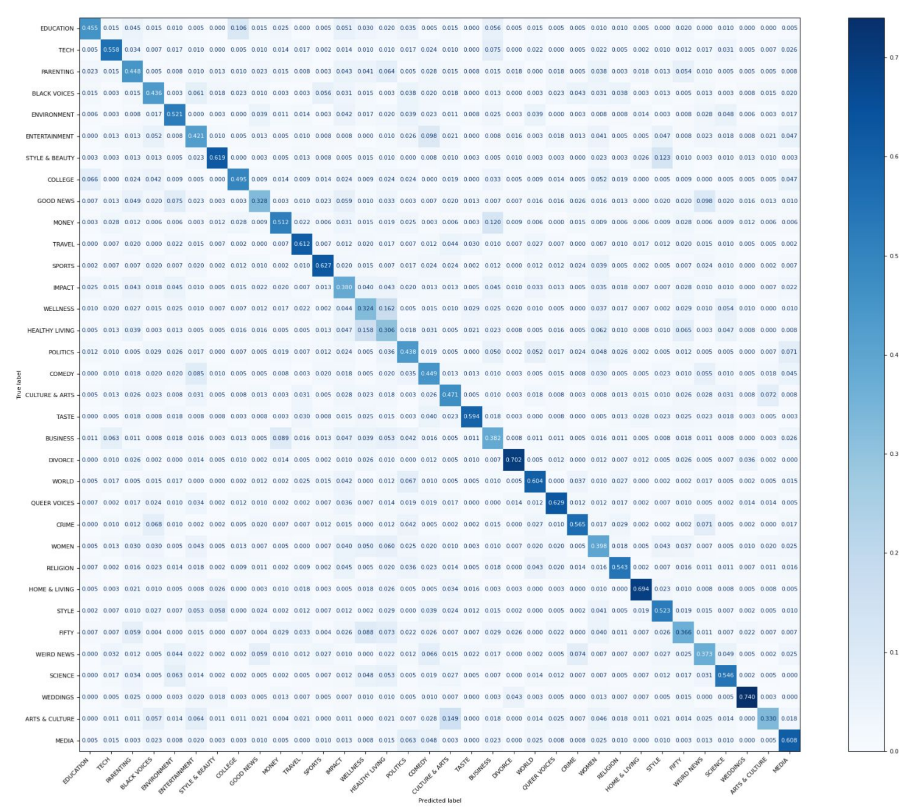
<u>OneVsRest/OneVsOne(Perceptron(TfidfVectorizer(char_ngrams)))</u>

- nur sehr geringe Verbesserung von Accurracy/F1 Score
- beste n_gram Größe(n): 5 (4) (6)
- Verschlechterung mit word_ngrams

Ergebnisse Huffington-Post Daten

Exkurs: Multilayer Perceptron MLP(CountVectorizer)

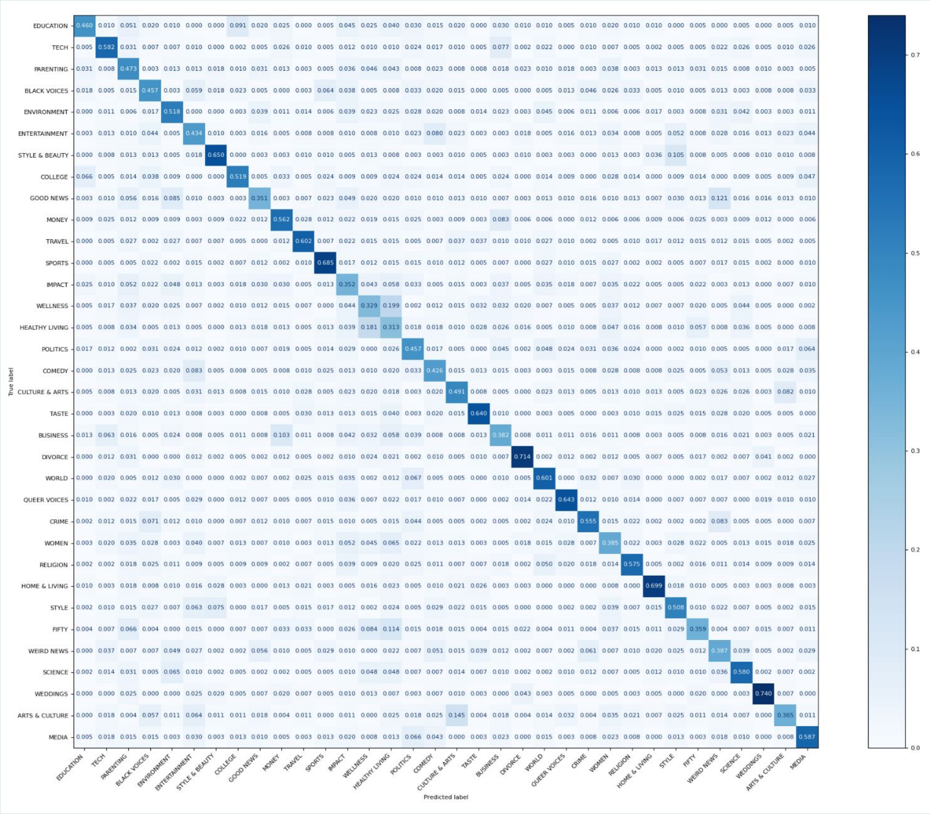
	precision	recall	f1-score	support
EDUCATION	0.523	0.455	0.486	198
TECH	0.603	0.558	0.579	416
PARENTING	0.407	0.448	0.426	391
BLACK VOICES	0.432	0.436	0.434	392
ENVIRONMENT	0.473	0.521	0.496	355
ENTERTAINMENT	0.384	0.421	0.402	387
STYLE & BEAUTY	0.714	0.619	0.663	391
COLLEGE	0.482	0.495	0.488	212
GOOD NEWS	0.386	0.328	0.355	305
MONEY	0.546	0.512	0.529	324
TRAVEL	0.629	0.612	0.621	405
SPORTS	0.673	0.627	0.649	410
IMPACT	0.328	0.380	0.352	400
WELLNESS	0.322	0.324	0.323	407
HEALTHY LIVING	0.246	0.306	0.273	386
POLITICS	0.424	0.438	0.431	420
COMEDY	0.383	0.449	0.413	399
CULTURE & ARTS	0.461	0.471	0.466	391
TASTE	0.720	0.594	0.651	397
BUSINESS	0.356	0.382	0.368	380
DIVORCE	0.772	0.702	0.735	419
WORLD	0.582	0.604	0.593	404
QUEER VOICES	0.666	0.629	0.647	415
CRIME	0.624	0.565	0.593	409
WOMEN	0.351	0.398	0.373	400
RELIGION	0.616	0.543	0.577	440
HOME & LIVING	0.728	0.694	0.711	386
STYLE	0.514	0.523	0.519	413
FIFTY	0.300	0.366	0.330	273
WEIRD NEWS	0.402	0.373	0.387	408
SCIENCE	0.545	0.546	0.545	414
WEDDINGS	0.779	0.740	0.759	400
ARTS & CULTURE	0.451	0.330	0.381	282
MEDIA	0.545	0.608	0.575	395
accuracy			0.506	12824
macro avg	0.511	0.500	0.504	12824
weighted avg	0.516	0.506	0.509	12824



Ergebnisse Huffington-Post Daten

Exkurs: Multilayer Perceptron MLP(TfidfVectorizer)

	precision	recall	f1-score	support
EDUCATION	0.520	0.460	0.488	198
TECH	0.610	0.582	0.595	416
PARENTING	0.405	0.473	0.436	391
BLACK VOICES	0.446	0.457	0.451	392
ENVIRONMENT	0.468	0.518	0.492	355
ENTERTAINMENT	0.402	0.434	0.417	387
STYLE & BEAUTY	0.715	0.650	0.681	391
COLLEGE	0.493	0.519	0.506	212
GOOD NEWS	0.407	0.351	0.377	305
MONEY	0.517	0.562	0.538	324
TRAVEL	0.654	0.602	0.627	405
SPORTS	0.657	0.685	0.671	410
IMPACT	0.323	0.352	0.337	400
WELLNESS	0.321	0.329	0.325	407
HEALTHY LIVING	0.231	0.313	0.266	386
POLITICS	0.438	0.457	0.448	420
COMEDY	0.443	0.426	0.434	399
CULTURE & ARTS	0.501	0.491	0.496	391
TASTE	0.677	0.640	0.658	397
BUSINESS	0.401	0.382	0.391	380
DIVORCE	0.757	0.714	0.735	419
WORLD	0.568	0.601	0.584	404
QUEER VOICES	0.681	0.643	0.662	415
CRIME	0.605	0.555	0.579	409
WOMEN	0.392	0.385	0.388	400
RELIGION	0.639	0.575	0.605	440
HOME & LIVING	0.728	0.699	0.713	386
STYLE	0.530	0.508	0.519	413
FIFTY	0.366	0.359	0.362	273
WEIRD NEWS	0.397	0.387	0.392	408
SCIENCE	0.571	0.580	0.576	414
WEDDINGS	0.767	0.740	0.753	400
ARTS & CULTURE	0.454	0.365	0.405	282
MEDIA	0.545	0.587	0.565	395
accuracy			0.517	12824
macro avg	0.519	0.511	0.514	12824
weighted avg	0.523	0.517	0.519	12824



Fazit Ergebnisse

- Sentiment Daten:
  - relativ “gutes” Ergebnis, im Bereich von Logistic Regression
  - besseres Ergebnis durch Character N-Grams (n = 5)
  - keine Verbesserung durch Entfernung von Stoppwörtern (nur word level)
  - keine Verbesserung durch dense Vektor Input
- Briefsammlung - Autoren:
  - “gutes” Ergebnis, OneVsRest besser als OneVsOne
  - besseres Ergebnis durch Character N-Grams (n = 5)
- Briefsammlung - Sprachen:
  - “sehr gutes” Ergebnis, kaum Unterschiede zwischen OneVsRest und OneVsOne
  - kaum zu verbesserndes Ergebnis durch Character N-Grams
- News Daten:
  - “mäßiges” Ergebnis aufgrund von geringem Input
  - Klassifikation mit OneVsOne besser als mit OneVsRest
  - MLP in ähnlichem Rahmen

Fazit Perzeptron

- schnelle und einfache Klassifikation linear trennbarer Daten
- durch OneVsRest und OneVsOne auch Multiclass-Klassifikation möglich
- überwachtes Lernen, Lernrate nicht erforderlich, keine Regularisierung notwendig
- Updates nur bei falscher Klassifikation
- “gute” Ergebnisse bei entsprechend klaren Inputs

Vielen Dank für eure Aufmerksamkeit!

Quellenverzeichnis

Forsström, J. J., et al. "Using data preprocessing and single layer perceptron to analyze laboratory data." Scandinavian Journal of Clinical and Laboratory Investigation 55.sup222 (1995): 75-81.

Jurafsky, Daniel, and James H. Martin. "Speech and Language Processing (Draft of December 29, 2021)." (2021).

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). “Text classification algorithms: A survey”. *Information*, 10(4), 150.

Manning, Chris and Schütze, Hinrich. (1999): Foundations of Statistical Natural Language Processing, MIT Press. Cambridge MA.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Raudys, Šarūnas. "Evolution and generalization of a single neurone: I. Single-layer perceptron as seven statistical classifiers." Neural Networks 11.2 (1998): 283-296.

Leonhard Grünschloß: Perzeptron, http://www.informatik.uni-ulm.de/ni/Lehre/WS04/ProSemNN/pdf/perzeptron.pdf ;2005