Multiclass Classification

Multiclass Classification wurde in unserem Projekt verwendet, um die Tweets von 8 Twitter-Accounts unterschiedlicher Zeitungen zuordnen zu können. Hierbei wurden auch die unterschiedlichen Nuancen der Schreibstile der jeweiligen Accounts maschinell untersucht.

Preprocessing der Daten

Bei der **Vorverarbeitung**, oder auch dem **Preprocessing** der Daten wurde die größtmögliche gemeinsame Schnittmenge von 160 Tweets gewählt. Dies ist gleichzeitig die Anzahl an Tweets des Datensatzes des Satiremagazines “Glasauge”, dem kleinsten vorliegenden Datensatzes. Dieser Vorgehensweise wurde gewählt, um einen ausbalancierten Datensatz mit gleichgroßen Datensätzen für alle Elemente zu gewährleisten. So wurden von jeder im Vorfeld gescrapten JSON-Datei der 8 Größten Zeitschriften die 160 ersten Objekte entnommen und zusammen in eine CSV-Datei geschrieben.

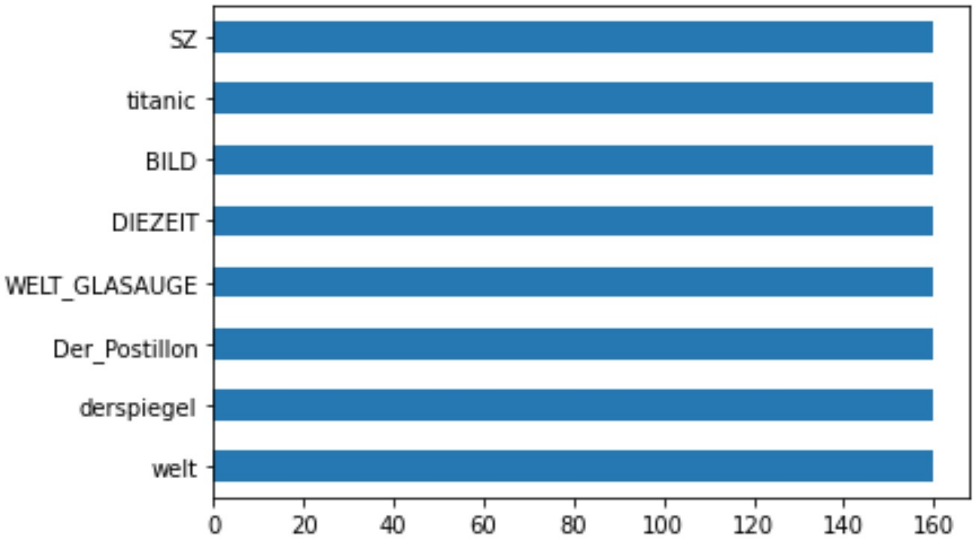


Abb.: Verteilung der Tweet-Inhalte mit dem dazugehörigen Label.

Das Augenmerk wurde auf die Eigenschaften der Objekte mit dem Schlüsseln: „content“ und „author“ gelegt. Die übrigen Eigenschaften wurden für die Untersuchungen nicht benötigt und entfernt.

```
(...)
"content": "Die Digitalisierung soll unser Leben einfacher machen, sagen alle. Aber genau das Gegenteil ist
passiert: Wir haben uns eine Welt geschaffen, der wir nicht mehr gewachsen sind. Und ich würde gerne meinen
Computer töten. https://t.co/rXf0Qw9AHR",
"author": "derspiegel",
(...)
```

Abb.: Beispielausschnitt aus dem Datensatz mit den Eigenschaften „content“ und „author“.

Bei dem Preprocessing der Daten wurden aus dem „content“ die Verlinkungen der Quelle des Originalartikels, die deutschen Stopwörter, Zahlen und die Interpunktion entfernt und in eine Pandas-DataFrame überführt, zusätzlich wurde die Kleinschreibung angewendet.

```
die digitalisierung leben einfacher machen sagen alle aber genau gegenteil passiert wir welt geschaffen
mehr gewachsen sind und gerne computer töten ,derspiegel
```

Abb.: Beispielausschnitt nach dem Preprocessing.

Wahl des Machine Learning Algorithmus

Es gibt unterschiedliche Machine Learning Algorithmen, um diese Klassifizierung durchzuführen, dazu gehören unter anderem:

- Linear Support Vector Machine,
- Multinomial Naive Bayes,
- Logistic Regression,
- Random Forest.

Um herauszufinden, welcher Algorithmus die besten Ergebnisse für die vorgesehene Klassifizierung liefert, wurde ein Test mit vorverarbeiteten Daten durchgeführt.
Die Ergebnisse einer 10-fachen Kreuzvalidierung haben gezeigt, dass Logistic Regression die besten Ergebnisse erzeugt.

```
cv_df.groupby('model_name').accuracy.mean()
```

```
model_name
LinearSVC                 0.531250
LogisticRegression        0.545312
MultinomialNB             0.507812
RandomForestClassifier    0.519531
```

Abb.: Die Ergebnisse der 10-fachen Kreuzvalidierung.

Multinominale logistische Regression

Nach der Auswahl des Algorithmus wird dieser nochmals auf die vorverarbeiteten Daten angewendet. Die Datensätze wurden im Verhältnis 80 % zu 20 % in Training- und Testdatensätze aufgeteilt und erneut der logistischen Regression unterzogen.
Die Evaluationstabelle zeigt, welche Ergebnisse der Lernalgorithmus, gemessen in Precision, Recall und Accuracy in Bezug auf die einzelnen Klassen erreicht hat:

	precision	recall	f1-score	support
BILD	0.55	0.40	0.47	42
DIEZEIT	0.97	1.00	0.98	31
Der_Postillon	0.36	0.57	0.44	21
SZ	0.63	0.50	0.56	34
WELT_GLASAUGE	0.65	0.77	0.71	31
derspiegel	0.58	0.44	0.50	41
titanic	0.47	0.70	0.56	23
welt	0.61	0.58	0.59	33
accuracy			0.60	256
macro avg	0.60	0.62	0.60	256
weighted avg	0.61	0.60	0.60	256

Abb.: Die Evaluationstabelle.

Ein weiteres Indiz ist die Konfusionsmatrix. Sie zeigt, wie zuverlässig der gewählte Algorithmus die einzelnen Klassen, d.h. die Twitter-Accounts der Zeitungen, anhand der Testdatensets analysiert:

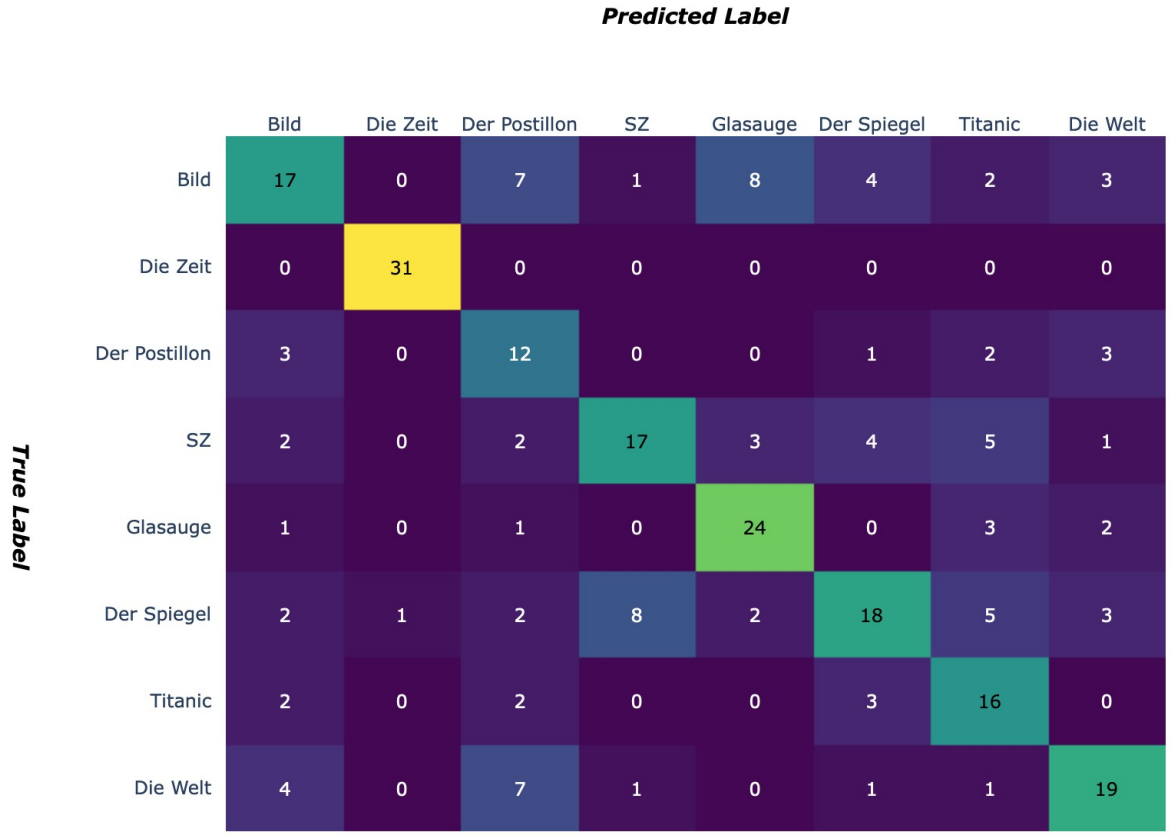


Abb.: Konfusionsmatrix.

Bei näherer Betrachtung der beiden Grafiken fallen hohe Werte bei der Zeitschrift 'Die Zeit' auf, dies könnte ein Indiz für häufige Wiederholung einzelner Wörter sein.

Bei dem Verfahren werden die einzelnen Wörter in dem Inhalt des Tweets vektorisiert und von dem Algorithmus gewichtet. Es ist von außen normalerweise nicht sichtbar, wie die Wörter gewichtet sind, jedoch mit der Verwendung der Python-Bibliothek ELI5 ist es möglich, das zu beobachten und Einfluss auf die Vorverarbeitung der Daten zu nehmen.

y=BILD top features		**y=DIEZEIT** top features		**y=Der_Postillon** top features		**y=SZ** top features		**y=WELT_GLASAUGE** top features		**y=derspiegel** top features	
Weight[?]	**Feature**	**Weight[?]**	**Feature**	**Weight[?]**	**Feature**	**Weight[?]**	**Feature**	**Weight[?]**	**Feature**	**Weight[?]**	**Feature**
+1.646	bildlive	+7.675	red	+1.784	leserbriefe	+3.534	szplus	+1.624	endlich	+1.583	nun
+0.955	telegram	+4.401	abo	+1.380	woche	+1.708	aktuell	+1.286	em	+1.124	die
+0.952	bayerns	+1.110	sagt	+1.359	kw	+1.213	ein	+1.239	merkel	+1.056	wurde
+0.919	bild	+0.888	ein	+0.999	sie	+1.090	jahr	+1.201	impfung	+1.002	seit
+0.907	omikronfakten	+0.732	menschen	+0.953	nachrichten	+1.089	von	+1.161	classics	+0.991	neue
+0.881	wegen	+0.643	sein	+0.947	sonntagsfrage	+0.944	die	+1.028	btw	+0.888	der
+0.828	politik	+0.617	zeit	+0.916	ehrliche	+0.828	leicht	+1.021	laschet	+0.868	spiegel
+0.791	hoffnung	+0.578	die	+0.916	videos	+0.814	ist	+0.856	rtl	+0.718	zehn
+0.766	telefonat	+0.558	schreibt	+0.840	weihnachtsmann	+0.774	zuversicht	+0.839	impfungen	+0.711	mehrere
+0.752	fpiatov	+0.510	weihnachten	+0.814	reinlauschen	+0.744	auf	+0.794	uefaeuro	+0.674	hier
+0.715	coronazahlen	+0.504	gespräch	+0.798	mal	+0.725	vierschanzentournee	+0.776	baerbock	+0.658	sie
+0.666	story	+0.495	doch	+0.771	alle	+0.681	in	+0.775	astrazeneca	+0.647	experten
+0.666	keine	+0.485	erzählt	+0.760	postillonlinkservice	+0.667	maxwell	+0.761	spd	+0.636	kosten
+0.663	richterleinspruch	+0.458	leben	+0.749	künftig	+0.667	ukrainekonflikt	+0.751	spahn	+0.634	fachleute
+0.602	kim	+0.451	archiv	+0.736	podcastillon	+0.661	corona	+0.732	berlin	+0.595	doch
+0.601	chinametropole	+0.449	weniger	+0.724	mehr	+0.651	datenlage	+0.724	ungeimpfte	+0.588	eine
… 746 more positive …		+0.410	natürlich	+0.699	danke	+0.642	martinbernstei	+0.713	lockdown	+0.586	geht
… 5126 more negative …		+0.404	sollte	*… 631 more positive …*		+0.591	beamte	*… 745 more positive …*		*… 1447 more positive …*	
-0.619	meistgelesen	*… 1358 more positive …*		*… 5241 more negative …*		*… 1406 more positive …*		*… 5127 more negative …*		*… 4425 more negative …*	
-0.637	der	*… 4514 more negative …*		-0.777	der	*… 4466 more negative …*		-0.747	sie	-0.609	meistgelesen
-0.641	abo	-0.439	szplus	-1.073	red	-0.675	abo	-0.838	die	-0.673	abo
-1.139	red	-0.469	meistgelesen	-1.172	die	-1.185	red	-1.042	red	-1.199	red

Abb.: Ausschnitt aus dem Programm, die Python Bibliothek ELI5, welche erlaubt die Gewichtung der einzelnen Features, der Wörter, zu sehen.

Es ist zu sehen, dass das Wort „red“, was die Abkürzung von „Redaktion“ ist, und das Wort „abo“ bei den Tweets von "Die Zeit" sehr stark gewichtet wurde, d.h. häufig vorkommt. Somit ist es zum Overfeeding des Algorithmus gekommen, dieses Phänomen wird auch „Schlauer Hans“ genannt. Es bedeutet, dass die Tweets nicht anhand des „Schreibstils“ zugewiesen werden, sondern aufgrund der einzelnen Gewichtungen, welche nur in der Klasse vorkommen. Somit gewichtet der Algorithmus solche Wiederholungen bei der Klassifizierung höher.

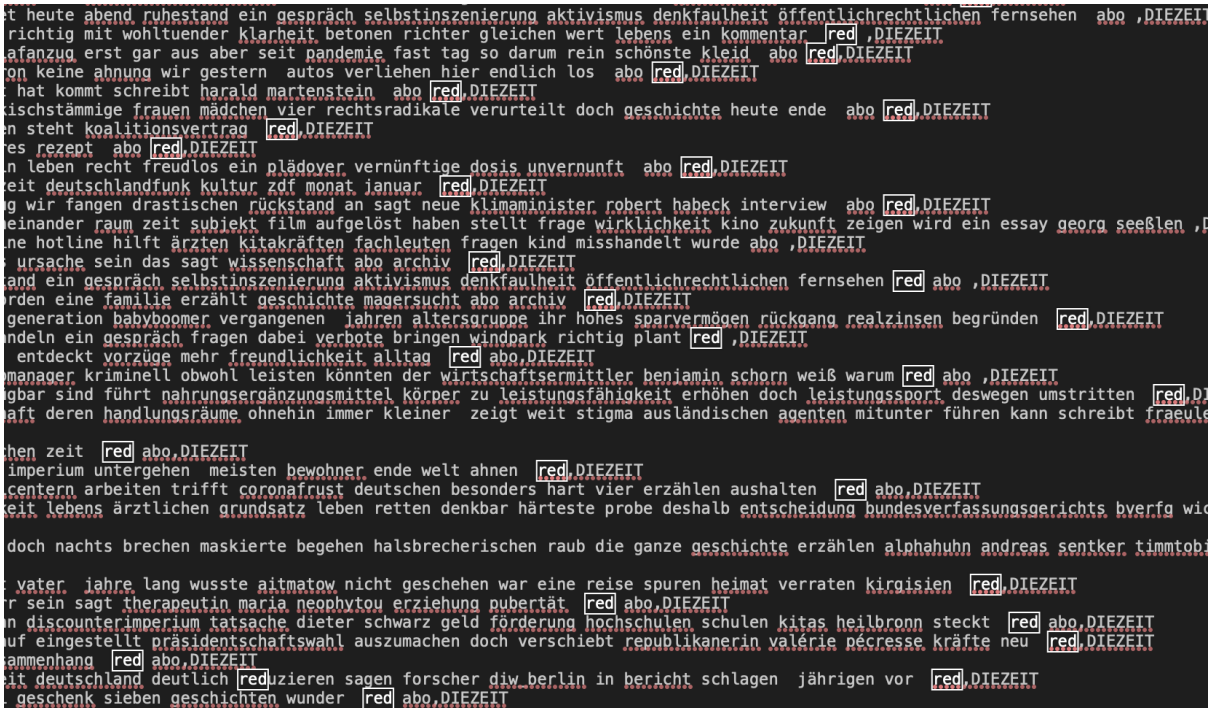


Abb.: Einblick in die CSV-Datei an der das Algorithmus gelernt hat, das Feature „red“ kommt in jedem Tweet von „Die Zeit“ vor.

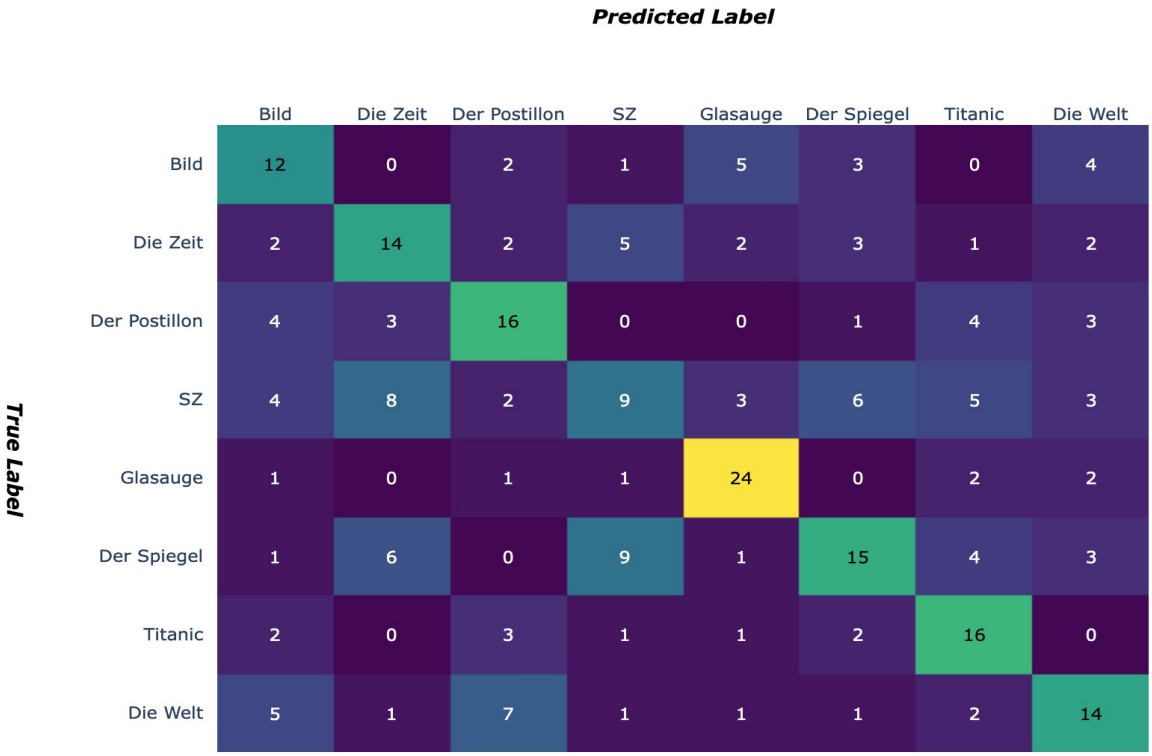
Dank der Möglichkeit, in den Algorithmus hineinzusehen, werden im nächsten Schritt die Daten noch mal verarbeitet und die stark gewichteten Features entfernt.

Preprocessing II und das Endmodel

Nach der Bereinigung der Datensätze von den vermeindlichen Overfeeding wurde die logistische Regression noch mal durchgeführt und das Model gespeichert. Dies sind die Metriken des aktuellen Models:

	precision	recall	f1-score	support
BILD	0.39	0.44	0.41	27
DIEZEIT	0.44	0.45	0.44	31
Der_Postillon	0.48	0.52	0.50	31
SZ	0.33	0.23	0.27	40
WELT_GLASAUGE	0.65	0.77	0.71	31
derspiegel	0.48	0.38	0.43	39
titanic	0.47	0.64	0.54	25
welt	0.45	0.44	0.44	32
accuracy			0.47	256
macro avg	0.46	0.48	0.47	256
weighted avg	0.46	0.47	0.46	256

Abb.: Die Evaluationstabelle Processing II



Das Model im Einsatz – Model Deployment

Besuchen Sie unsere Vorhersagen-Applikation.
Sehen Sie das trainierte Model im Einsatz.

Link → https://hauptseminardh.herokuapp.com