



Sentiment Analysis

Nico Mirchandani



Gliederung

1. Sentiment Analysis
2. Eigener Ansatz
3. Andere Ansätze
4. Evaluation



Sentiment Analysis

- Bei der Sentiment Analysis werden Sprachverarbeitung Techniken genutzt um zu erkennen, ob ein Text positiv oder negativ ist
- Benutzt für Marktforschung, bspw. um Kundenrezensionen zu analysieren

Wörter löschen

Satz Aufteilung

Löschung von Zeichen

Negationen kennzeichnen

Wörter aufteilen

Preprocessing

Dies ist ein Beispielsatz, in dem ich einen Link [zur Hochschule verwenden](#). Dies hier "S/169" ist ein Copyright-Zeichen. Es ist ein Copyright-Zeichen, wovon das kein Zeichen als Unicode gespeichert.

URLs und Emails werden gelöscht, da sie keinen Mehrwert haben und immer neutral sind

Wörter löschen

Satz Aufteilung

Löschung von Zeichen

Negationen kennzeichnen

Wörter aufteilen



Preprocessing

[["dies ist ein beispielsatz"], [","], ["in dem ich einen link zur hochschule verwende"], ["."], ["dies hier
"©" ist ein copyright-zeichen"], ["."], ["es wurde aber kein zeichen"], [","], ["sondern als unicode
gespeichert"], ["."]]

Sätze werden in Arrays gespeichert, zudem werden Wörter kleingeschrieben

Wörter löschen

Satz Aufteilung

Löschung von Zeichen

Negationen kennzeichnen

Wörter aufteilen



Preprocessing

[[“dies ist ein beispielsatz”], [“”], [“in dem ich einen link zur hochschule verwende”], [“.”], [“dies hier “”
ist ein copyright-zeichen”], [“.”], [“es wurde aber kein zeichen”], [“.”], [“sondern als unicode
gespeichert”], [“.”]]

Bestimmte Zeichen sind unnötig, werden gelöscht

Wörter löschen

Satz Aufteilung

Löschung von Zeichen

Negationen kennzeichnen

Wörter aufteilen



Preprocessing

[[["dies ist ein beispielsatz"], [","], ["in dem ich einen link zur hochschule verwende"], [","], ["dies hier
"" ist ein copyright-zeichen"], [","], ["es wurde aber kein zeichen"], [","], ["sondern als unicode
gespeichert"], [","]], [0, 0, 0, 0, 0, 1, 0, 0, 0]]

Bestimmte Zeichen sind unnötig, werden gelöscht

Wörter löschen

Satz Aufteilung

Löschung von Zeichen

Negationen kennzeichnen

Wörter aufteilen

Preprocessing

[[["dies"], ["ist"], ["ein"], ["beispielsatz"]], [{";"}], [{"in"}, {"dem"}, {"ich"}, {"einen"}, {"link"}, {"zur"}, {"hochschule"}, {"verwende"}], [{"."}], [{"dies"}, {"hier"}], [{"ist"}, {"ein"}, {"copyright-zeichen"}], [{"."}], [{"es"}, {"wurde"}, {"aber"}, {"kein"}, {"zeichen"}], [{","}], [{"sondern"}, {"als"}, {"unicode"}, {"gespeichert"}], [{"."}]], [0, 0, 0, 0, 0, 1, 0, 0, 0]]

Bestimmte Zeichen sind unnötig, werden gelöscht

Training

Wörter durchgehen

Schauen ob es zur Negation gehört

Sentiment Wert speichern

```
"Wort": {  
  "-1": x,  
  "1": y  
}
```



Bewertung

Wörter durchgehen

Schauen ob es zur Negation gehört

Sentiment Wert bestimmen

Wörter die häufiger im System
eingetragen wurden, haben eine
kleinere Gewichtung als Wörter die
sehr wenig vorkommen

Evaluation

"**Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions."

```
def evaluate(truepositive, trueneegative, falsepositive, falsenegative, beta=1):
    accuracy = (trueneegative + truepositive)/(trueneegative + truepositive + falsepositive + falsenegative)

    precision = truepositive/(truepositive+falsepositive)

    recall = truepositive/(truepositive+falsenegative)

    f = (1+beta**2) * (((precision*beta**2)*recall)/(precision+recall))

    return f, accuracy, precision, recall
```

$$\text{recall} = \frac{\text{truepositives}}{\text{truepositives} + \text{falsenegatives}}$$

Wenn Zeit angegeben, dann wurde trainiert und alle 50.000
Datensätze eine Testung mit 10.000 Datensätzen gemacht.
Testung auf meinem PC:
Grafikkarte: RTX 3060 TI
Prozessor: AMD Ryzen 5600X.

		0	1
True Label	0	48 true negatives	8 false positives
	1	4 false negatives	37 true positives

Evaluation meines Ansatzes

Zeit: 1745.950s (ca. 29min)

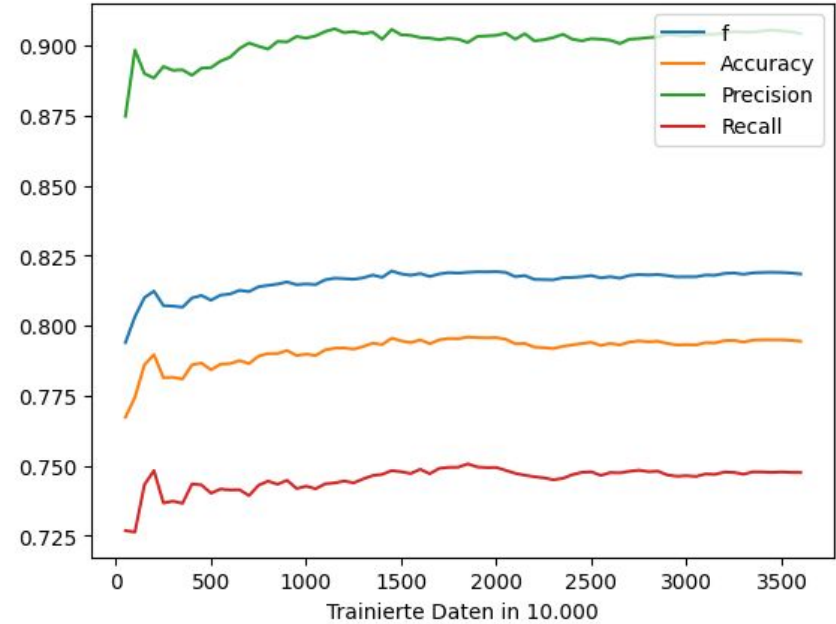
Accuracy: 0.7938019845049613

Precision: 0.89315

Recall: 0.7451019650536626

F: 0.8124364679957884

Positive	178630	21370
Negative	138890	61109
	true	false





Textblob

- Python Bibliothek für natürliche Sprachverarbeitung
- Funktionen:
 - Tokenisieren
 - Taggen
 - Parsen
 - Klassifizieren von Texten
- Kann Stimmung von Texten bestimmen: `TextBlob(TEXT).sentiment.polarity`
- Kann nicht weiter trainiert werden

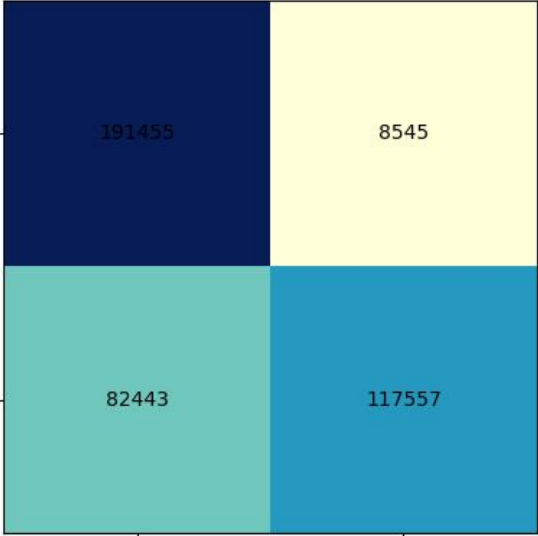
Textblob Evaluation

Accuracy: 0.684745

Precision: 0.957275

Recall: 0.6195714082301012

F: 0.7522612433498621



A confusion matrix for Textblob evaluation. The matrix is a 2x2 grid of colored squares. The top row is labeled 'Positive' on the left. The bottom row is labeled 'Negative' on the left. The bottom-left square is light teal and contains the number 82443. The bottom-right square is blue and contains the number 117557. The top-left square is dark blue and contains the number 191455. The top-right square is yellow and contains the number 8545. Below the grid, the columns are labeled 'true' and 'false'.

Positive	191455	8545
Negative	82443	117557
	true	false



SK Learning

- Ist eine Bibliothek für Python, die sich auf Machine Learning fokussiert
- Verfügt über eine große Auswahl an Classifier, die man einfach implementieren kann
- Classifier sind dabei die Algorithmen

Pro:

Einfach zu implementieren, gute Ergebnisse

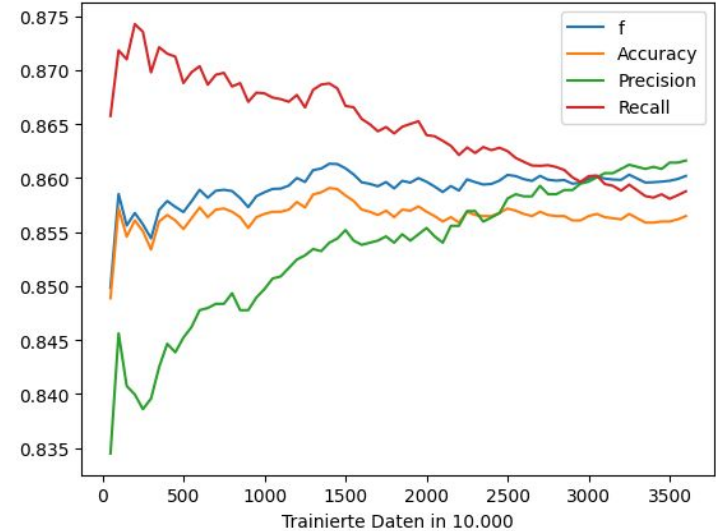
Contra:

Lange Zeiten im Vergleich zu meinem Modell

SK Learning - MultinomialNB

Time: 11314.689s (ca 188min)
Accuracy: 0.8530196325490814
Precision: 0.835375
Recall: 0.8659338039410808
F: 0.8503799542935089

Positive	167075	32925
Negative	174132	25867
	true	false



SK Learning - Tree Classifier

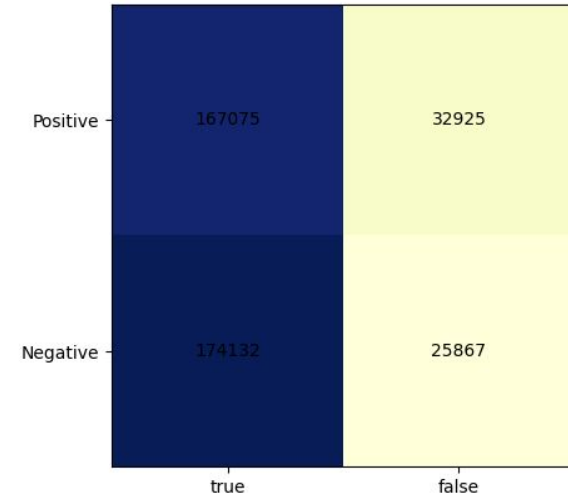
Accuracy: 0.8530196325490814

Precision: 0.835375

Recall: 0.8659338039410808

F: 0.8503799542935089

Zeit: >6h



A confusion matrix for a SK Learning Tree Classifier. The matrix is a 2x2 grid. The rows are labeled 'Positive' and 'Negative' on the left. The columns are labeled 'true' and 'false' at the bottom. The top-left cell (Positive, true) is dark blue and contains the value 167075. The top-right cell (Positive, false) is light yellow and contains the value 32925. The bottom-left cell (Negative, true) is dark blue and contains the value 174132. The bottom-right cell (Negative, false) is light yellow and contains the value 25867.

Positive	167075	32925
Negative	174132	25867
	true	false



LightGBM & Spacy

- Leider nur bis 1,3 Millionen Datensätze gekommen.
- LightGBM ist eine Bibliothek für maschinelles Lernen
- Spacy ist eine Bibliothek für die Verarbeitung von Sprachen
 - Hier bekommt man auch Vektoren
- Zusammen verbunden kann man in einfachen Schritten eine Sentiment Analysis durchführen
- Im Verhältnis hat dies am Längsten gebraucht zu trainieren

LightGBM & Spacy - Evaluation

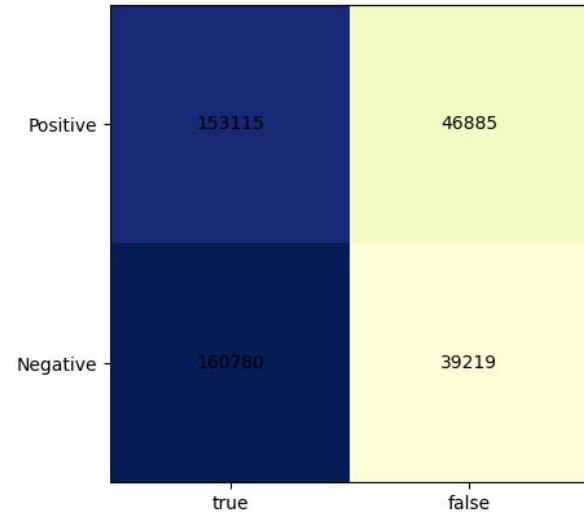
Time: für 1,3 Millionen Datensätze: ca. 6h

Accuracy: 0.7847394618486546

Precision: 0.765575

Recall: 0.7960890950118024

F: 0.780533932822544



A confusion matrix visualization showing the performance of a model. The matrix is a 2x2 grid. The rows are labeled 'Positive' and 'Negative' on the left. The columns are labeled 'true' and 'false' at the bottom. The cells contain the following counts: Positive-true is 153115 (dark blue), Positive-false is 46885 (light yellow), Negative-true is 160780 (dark blue), and Negative-false is 39219 (light yellow).

Positive	153115	46885
Negative	160780	39219
	true	false



Vergleich

Modell	F-Wert	Accuracy	Precision	Recall
Multinomial NB	0,85	0,853	0,83	0,866
Tree Classifier	0.85	0.85	0.835	0.866
Mein Modell	0,81	0,79	0,89	0,745
LightGBM + Spacy	0.78	0.78	0.77	0.79
Textblob	0,75	0,68	0,95	0,68
Random	0,5	0,5	0,5	0,5



Ausblick

- Aus zeitlichen Gründen & fehlenden Mitglied Twitter API nicht ausprobiert
 - theoretisch, wenn ich Zeit und Lust habe weiter zu arbeiten
- Twitter API ausprobieren
- Vergleichen mit anderen Datensätzen, vllt auch mal unausgewogenen
- [...]



Github

<https://github.com/nicomir02/Sentiment-Analysis> (derzeit auf Private gestellt)



Quellen

<https://mindsquare.de/knowhow/sentimentanalyse/>

<https://www.jeremyjordan.me/evaluating-a-machine-learning-model/>

<https://scikit-learn.org>

<https://spacy.io/>

<https://lightgbm.readthedocs.io/en/v3.3.2/>

[Zudem weitere Quellen in meinem Jupyter Notebook]



Danke für die Aufmerksamkeit

Nico Mirchandani