Projekt Fake News

In diesem Projekt entwickeln Sie ein System zur Erkennung von Fake News mithilfe von Natural Language Processing (NLP) und Machine Learning. Ziel ist es, Inhalte aus einem Datensatz analysieren, Merkmale zu extrahieren und ein Klassifikationsmodell (Random Forest Classifier) zu trainieren, das zwischen echten Nachrichten und Fake News unterscheiden kann.

Zielsetzung

- Verständnis von NLP-Methoden wie Tokenisierung, Stoppwörter-Entfernung, POS-Tagging und *Named Entity Recognition* (NER).
- Anwendung von Machine Learning für Klassifikationsprobleme.
- Entwicklung einer praxisnahen Lösung für ein reales Problem.

Aufgabenstellung im Detail

Datensatz einlesen und aufbereiten

Laden Sie den bereitgestellten Datensatz *news.csv*, der Nachrichten und ihre Klassifizierung enthält. Der Datensatz hat folgende Spalten:

	id	url	Titel	Body	Kategorie	Datum	Quelle	Fake	Art
0	773233	http://www.der-postillon.com/2018/01/grokoleak...	Exklusiv! Das geheime WhatsApp-Chat-Protokoll ...	Die Sondierungsgespräche zwischen Union und SP...	wirtschaft	2018-01-18 00:00:00	Postillion	1	NaN
1	773234	http://www.der-postillon.com/2018/01/trump-san...	Trump droht, jeden zu verspeisen, der an seine...	Nun ist es auch medizinisch offiziell bestätig...	wirtschaft	2018-01-17 00:00:00	Postillion	1	NaN
2	773235	http://www.der-postillon.com/2018/01/fdp-sondi...	Soli runter, keine Steuererhöhungen, kein Klim...	Es waren zähe Verhandlungen, doch die Freien D...	wirtschaft	2018-01-12 00:00:00	Postillion	1	NaN
3	773236	http://www.der-postillon.com/2018/01/joachim-s...	Hat sie eine Affäre? Joachim Sauer glaubt Ange...	Wo treibt sie sich immer bis spät in die Nacht...	wirtschaft	2018-01-09 00:00:00	Postillion	1	NaN
4	773237	http://www.der-postillon.com/2018/01/halb-so-s...	"Er hat ja nur HALBneger gesagt": So begründet...	Der Parteivorstand drückt nochmal ein Auge zu:...	wirtschaft	2018-01-08 00:00:00	Postillion	1	NaN

Abbildung 1: Der Datensatz news.csv

Wir sind ausschließlich an den Spalten *Body* und *Fake* interessiert. Die Spalte *Body* enthält die Nachricht, die es zu klassifizieren gilt. In *label* ist wiederum hinterlegt, ob es sich um eine „echte“ Nachricht (0) oder „Fake News“ (1) handelt. Erstellen Sie basierend auf diese Spalten das DataFrame `data` und ändern Sie hierbei die Spaltenbezeichner auf *text* bzw. *label*.

	text	label
0	Die Sondierungsgespräche zwischen Union und SP...	1
1	Nun ist es auch medizinisch offiziell bestätig...	1
2	Es waren zähe Verhandlungen, doch die Freien D...	1
3	Wo treibt sie sich immer bis spät in die Nacht...	1
4	Der Parteivorstand drückt nochmal ein Auge zu:...	1

Abbildung 2: Der auf die Spalten „text“ und „label“ reduzierte Datensatz.

Textaufbereitung

Wie im nachfolgenden Screenshot (Abbildung 3) zu sehen ist, werden dem DataFrame *data* im Rahmen der Textaufbereitung vier weitere Spalten hinzugefügt.

	text	label	cleaned_text	adjective_count	adverb_count	entity_count
0	Visionär: Konzeptpapier läßt "Bermudadreieck d...	1	visionär konzeptpapier läßt bermudadreieck pro...	33	23	23
1	Zu einem tödlichen Zwischenfall kam es heute i...	1	tödlichen zwischenfall steirischen krankenhaus...	9	26	9
2	15. November 2017 Wegen eines peinlichen Fehl...	1	november peinlichen fehlers beliebten kalender...	16	32	13
3	Es ist einfach zum Verzweifeln! Schon seit Woc...	1	einfach verzweifeln wochen liegt hochschwanger...	10	58	16
4	Nach dem Referendum in Griechenland sehen die ...	0	referendum griechenland sehen finanzminister b...	10	6	18

Abbildung 3: Ergebnis der Textaufbereitung

Der Inhalt von *cleaned_text* entspricht dem ursprünglichen Text, jedoch mit einigen Anpassungen: Stoppwörter wurden entfernt, und es wurden nur Wörter berücksichtigt, die ausschließlich aus Buchstaben und/oder Ziffern bestehen. Außerdem wurde der gesamte Text in Kleinbuchstaben umgewandelt.

Die Spalten *adjective_count*, *adverb_count* und *entity_count* enthalten zusätzliche linguistische Merkmale (Features), die aus dem Text extrahiert wurden. Diese helfen dabei, das maschinelle Lernmodell mit mehr Informationen zu versorgen, die potenziell nützlich sein können, um zwischen echten Nachrichten und Fake News zu unterscheiden. Bedeutung der Spalten im Detail:

adjective_count

- Diese Spalte gibt an, wie viele Adjektive (z. B. „groß“, „schnell“, „schockierend“) im Text vorkommen.
- Adjektive können darauf hinweisen, ob ein Text emotional aufgeladen oder sachlich-neutral ist.
- Fake News enthalten oft viele Adjektive, um Aufmerksamkeit zu erregen und eine emotionale Reaktion hervorzurufen.

adverb_count

- Diese Spalte zählt die Anzahl der Adverbien (z. B. „extrem“, „offensichtlich“, „wahrscheinlich“) im Text.
- Adverbien können auf Übertreibungen oder unpräzise Aussagen hindeuten, die ebenfalls typisch für Fake News sein können.

entity_count

- Diese Spalte gibt die Anzahl der benannten Entitäten (Named Entities) im Text an, wie z. B. Namen von Personen, Organisationen, Orten oder Datumsangaben.
- Fake News könnten überproportional viele oder auffällige Entitäten enthalten, um den Text glaubwürdiger erscheinen zu lassen.

Die Umsetzung dieser Arbeitsschritte hat in der Python-Funktion `preprocess_text(text)` zu erfolgen.

Wichtig: Das Dataset umfasst über 60.000 Datensätze, deren Verarbeitung viel Zeit in Anspruch nehmen würde – eine typische Herausforderung bei NLP-Aufgaben. Um die Verarbeitungszeit zu reduzieren, empfiehlt es sich, die Anzahl der Datensätze zu verkleinern. Der Datensatz enthält 59.241 echte Nachrichten und 4.627 Fake News. Daher bietet es sich an, die echten Nachrichten auf 4.627 zu reduzieren, um eine symmetrische Verteilung der Klassen zu erreichen. Sollte die Verarbeitungszeit bei rund 9.600 Datensätzen dennoch zu hoch sein, kann der Datensatz weiter verkleinert werden.

Verwenden Sie die Funktion `sample`[1], um n-viele Datensätze zufallsbasiert aus einem DataFrame zu selektieren. `sample` kann auch zum Durchmischen eines DataFrames verwendet werden. Hierbei ist der Parameter `fract` auf „1“ zu setzen. Die Durchmischung ist dann notwendig, wenn Sie ein DatenFrame, das ausschließlich echte Nachrichten enthält, mit einem Fake News-DataFrame zusammenfügen.

Sämtliche Schritte der Textaufbereitung sind in der Funktion `preprocess_text(text)` zu erledigen. Diese speichert oder retourniert den aufbereiteten Text und die extrahierten Merkmale in einem neuen DataFrame.

Merkmalsextraktion

Maschinelle Lernverfahren können nicht direkt mit Rohtexten arbeiten, weil sie nur Zahlen verstehen. Die Merkmalsextraktion übersetzt den Text in Zahlen, die das Modell verarbeiten kann. Dabei geschieht Folgendes:

Zuerst wird aus allen Texten ein Wörterbuch erstellt, das alle einzigartigen Wörter enthält. Jedes dieser Wörter wird zu einer Spalte in einer Tabelle (siehe Abbildung 4). Anschließend wird gezählt, wie oft jedes Wort in einem Text vorkommt. Wenn ein Text zum Beispiel das Wort *"Politik"* dreimal enthält, steht in der entsprechenden Spalte eine 3. Dieses Vokabular ist wie eine Art Nachschlagewerk, das dem Computer hilft, die Texte in Zahlen umzuwandeln.

Am Ende der Merkmalsextraktion entsteht ein DataFrame mit einer Vielzahl an numerischen Merkmalen (siehe Abbildung 4). Jede Zeile repräsentiert einen einzelnen Text, und jede Spalte enthält spezifische Informationen über diesen Text, sei es die Häufigkeit eines bestimmten Wortes oder ein linguistisches Merkmal (siehe bspw. Abbildung 4, adjective_account oder adverb_account).

[1] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sample.html

Wort 1	Wort 2	Wort 3	...	Wort n	adjective_count	adverb_count	entity_count
2	0	1	...	3	5	2	1
1	4	0	...	0	3	1	0

Abbildung 4: Beispiel für kombinieren Datensätze

Erstellen Sie das Vokabular bzw. die Merkmale mit dem *CountVectorizer*[2]. Untersuchen Sie, ob es besser ist, das gesamte Vokabular (ca. 37.590 Wörter) oder eine reduzierte Anzahl von Wörtern (z.B. 1.000) zu verwenden, um den *RandomForest*-Classifier zu trainieren. Um die Anzahl der zu berücksichtigenden Merkmale zu reduzieren, ist der Parameter `max_features` des *CountVectorizers* zu verwenden. Dokumentieren Sie ihre Ergebnisse.

Die Abbildung 5 und 6 zeigen eine mögliche Ausgabe sämtlicher Merkmale/Features bei einem reduzierten Vokabular.

	aachen	aachener	aamodt	abartige	abbas	abbau	abbaubank	abbaubar	abbauen	abbeißen
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0

Abbildung 5: Vokabular-Merkmale kombiniert mit linguistischen Merkmalen (Teil 1)

Klassifikationsmodell trainieren

Gehen Sie folgendermaßen vor:

- Teilen Sie den vorbereiteten Datensatz in Trainingsdaten (80%) und Testdaten (20%) auf.
- Trainieren Sie einen *Random Forest Classifier*, um die Nachrichten zu klassifizieren.
- Evaluieren Sie das Modell mit der Genauigkeit und prüfen Sie, wie gut es echte Nachrichten von Fake-News unterscheiden kann.

Dokumentation

Erstellen Sie ein Prtokoll mit Deckblatt und Inhaltsverzeichnis, das sämtliche Schritte auf Punkt und Beistrich dokumentiert. Kopieren Sie hierzu den Quellcode in das Dokument und kommentieren Sie diesen. Dokumentieren Sie auch Zwischenschritte/-ergebnisse, so das maximale Transparenz gegeben ist.

[2] https://scikit-learn.org/1.5/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

...	üppige	üppigen	üppiger	אווירית	יחידה	ישראל	משטרת	adjective_count	adverb_count	entity_count
...	0	0	0	0	0	0	0	33	23	23
...	0	0	0	0	0	0	0	9	26	9
...	0	0	0	0	0	0	0	16	32	13
...	0	0	0	0	0	0	0	10	58	16
...	0	0	0	0	0	0	0	10	6	18
...	...	...	...	...	...	...	...	...	...	...
...	0	0	0	0	0	0	0	7	9	14
...	0	0	0	0	0	0	0	19	36	20
...	0	0	0	0	0	0	0	36	29	30
...	0	0	0	0	0	0	0	23	38	24
...	0	0	0	0	0	0	0	36	48	23

Abbildung 6: Vokabular-Merkmale kombiniert mit linguistischen Merkmalen (Teil 2)