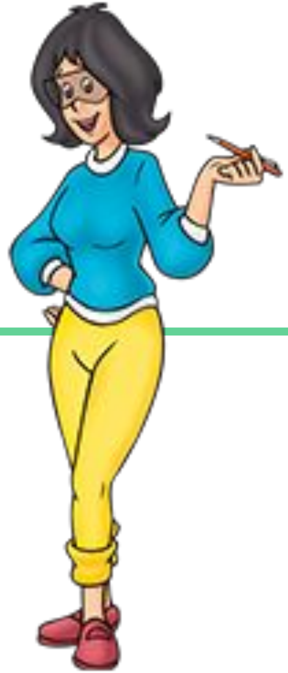
Zeitungsartikel-klassifikation



Die rasenden Reporter, 26.01.2023

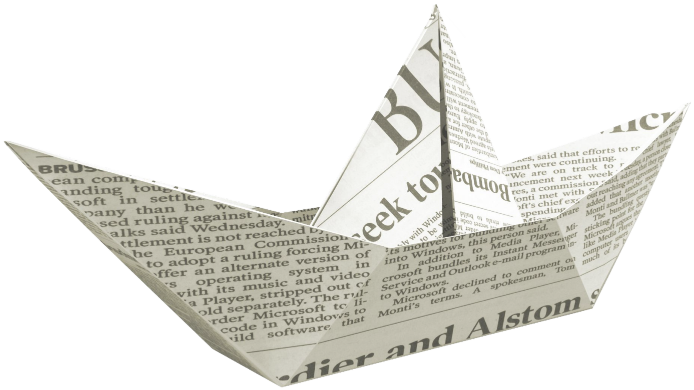
Caroline Schmidt, Marvin Spurk, Hannah Schult, Sofie Pischl, Viet Duc Kieu

Agenda

- Ziel & Use Case
- Vorgehen
- Datensatz
- Datenanalyse
- Datenverarbeitung
- Modellauswahl
- Anwendung
- Fazit

Zielsetzung

- Automatische Klassifikation von Zeitungsartikeln
- Nur Titel und Beschreibung benötigt
- Bestes Modell finden
- Anwendungsoberfläche bereitstellen



Use Case

- Einsortierung von Zeitungsartikeln
    - Nach Upload werden Artikel zugeordnet
    - Ersetzt manuelle Zuweisung

- Recommendation systems
    - Webcrawler durchsuchen das Internet
    - Newsartikel müssen für Nachrichtendienste geordnet werden

Vorgehen

- Vorgehen grob nach Scrum
  - Weekly/ Bi-weekly Sprints
  - Verwendung von Trello und Github

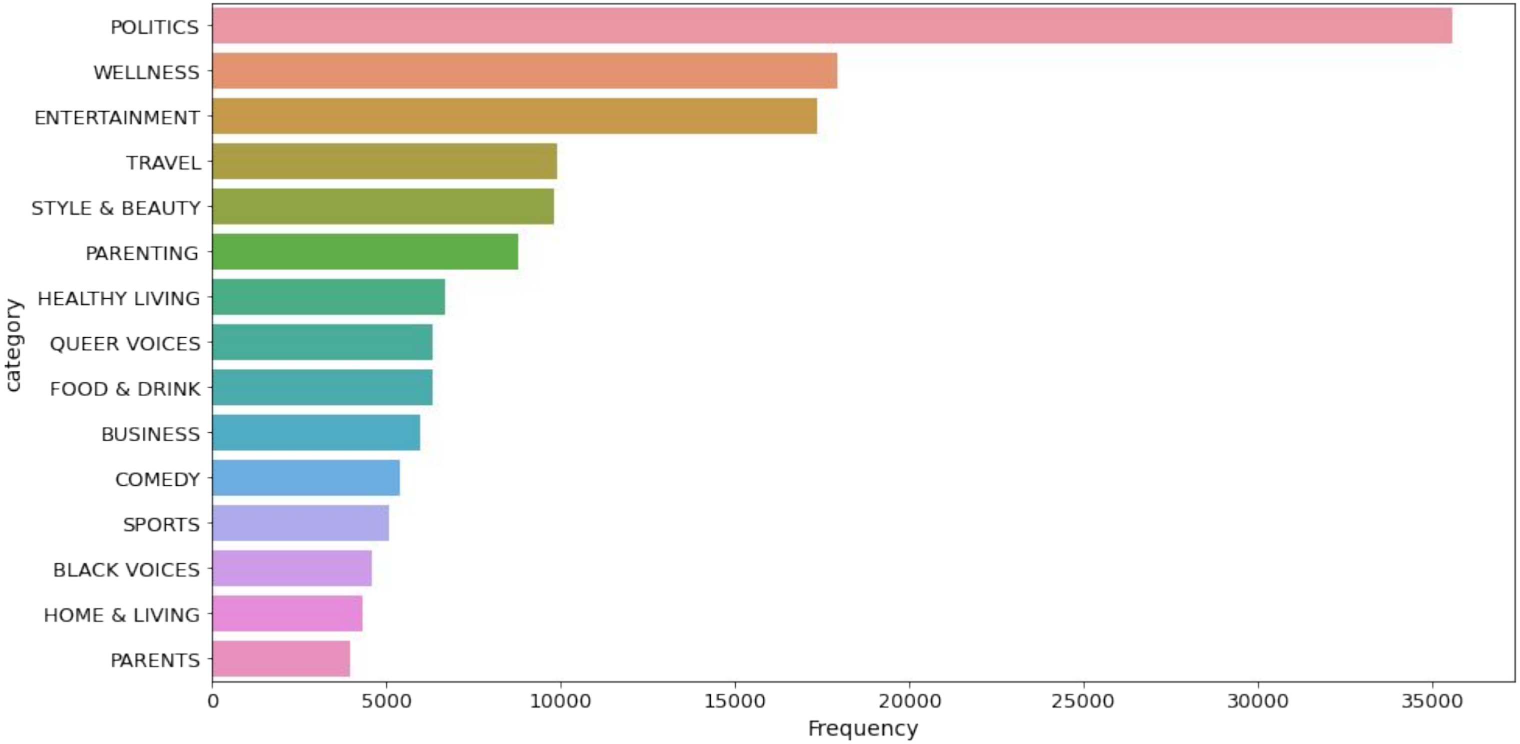
- Teamaufteilung
  - Hannah: Modellentwicklung & -auswahl
  - Sofie: Modellentwicklung, Präsentation
  - Duc: Modell- & Anwendungsentwicklung
  - Marvin: EDA, Use-Cases
  - Caro: EDA, Präsentation

Datensatz

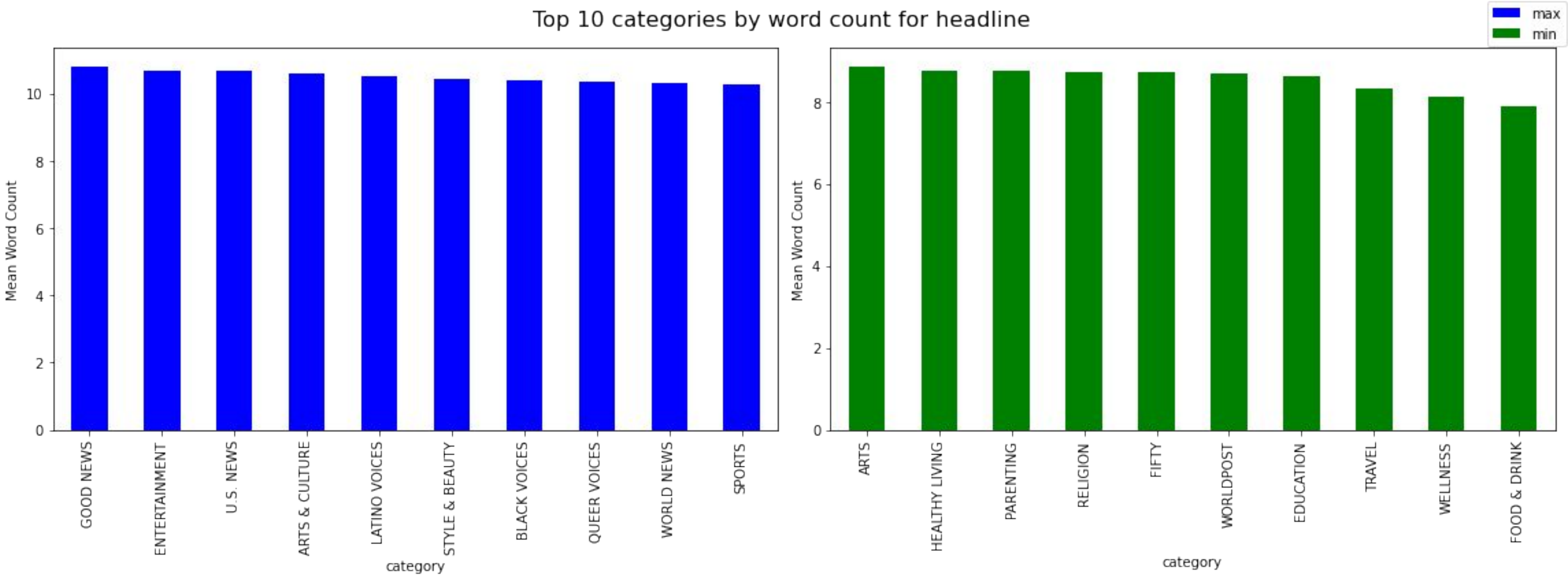
- Zeitungsartikel aus der HuffPost
- 210.000 Einträgen
- Englisch, 42 Kategorien
- Zwischen 2012 und 2022

	link	headline	category	short_description	authors	date
0	https://www.huffpost.com/entry/covid-boosters-...	Over 4 Million Americans Roll Up Sleeves For O...	U.S. NEWS	Health experts said it is too early to predict...	Carla K. Johnson, AP	2022-09-23
1	https://www.huffpost.com/entry/american-airlin...	American Airlines Flyer Charged, Banned For Li...	U.S. NEWS	He was subdued by passengers and crew when he ...	Mary Papenfuss	2022-09-23
2	https://www.huffpost.com/entry/funniest-tweets...	23 Of The Funniest Tweets About Cats And Dogs ...	COMEDY	"Until you have a dog you don't understand wha...	Elyse Wanshel	2022-09-23
3	https://www.huffpost.com/entry/funniest-parent...	The Funniest Tweets From Parents This Week (Se...	PARENTING	"Accidentally put grown-up toothpaste on my to...	Caroline Bologna	2022-09-23
4	https://www.huffpost.com/entry/amy-cooper-lose...	Woman Who Called Cops On Black Bird-Watcher Lo...	U.S. NEWS	Amy Cooper accused investment firm Franklin Te...	Nina Golgowski	2022-09-22

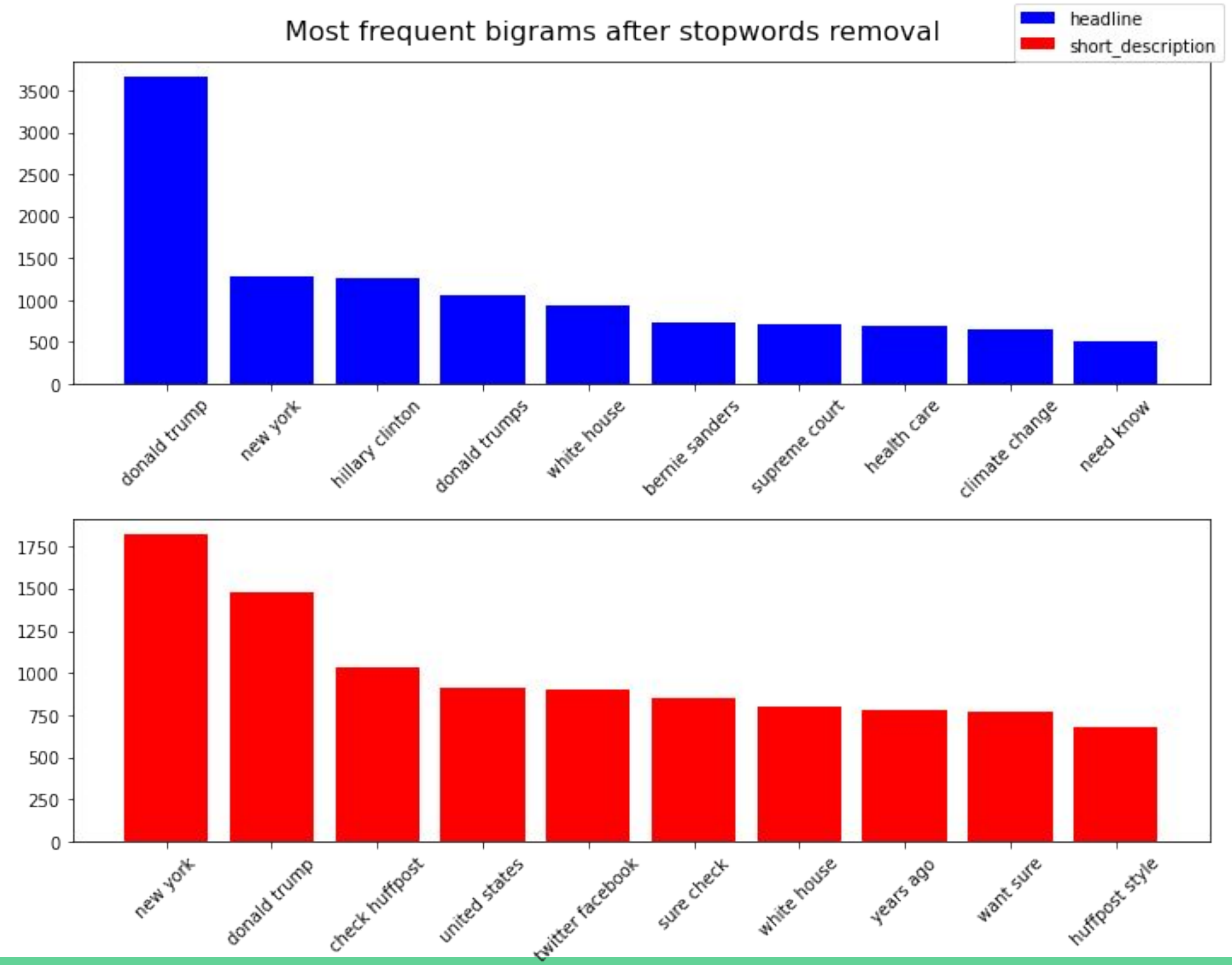
Datenanalyse



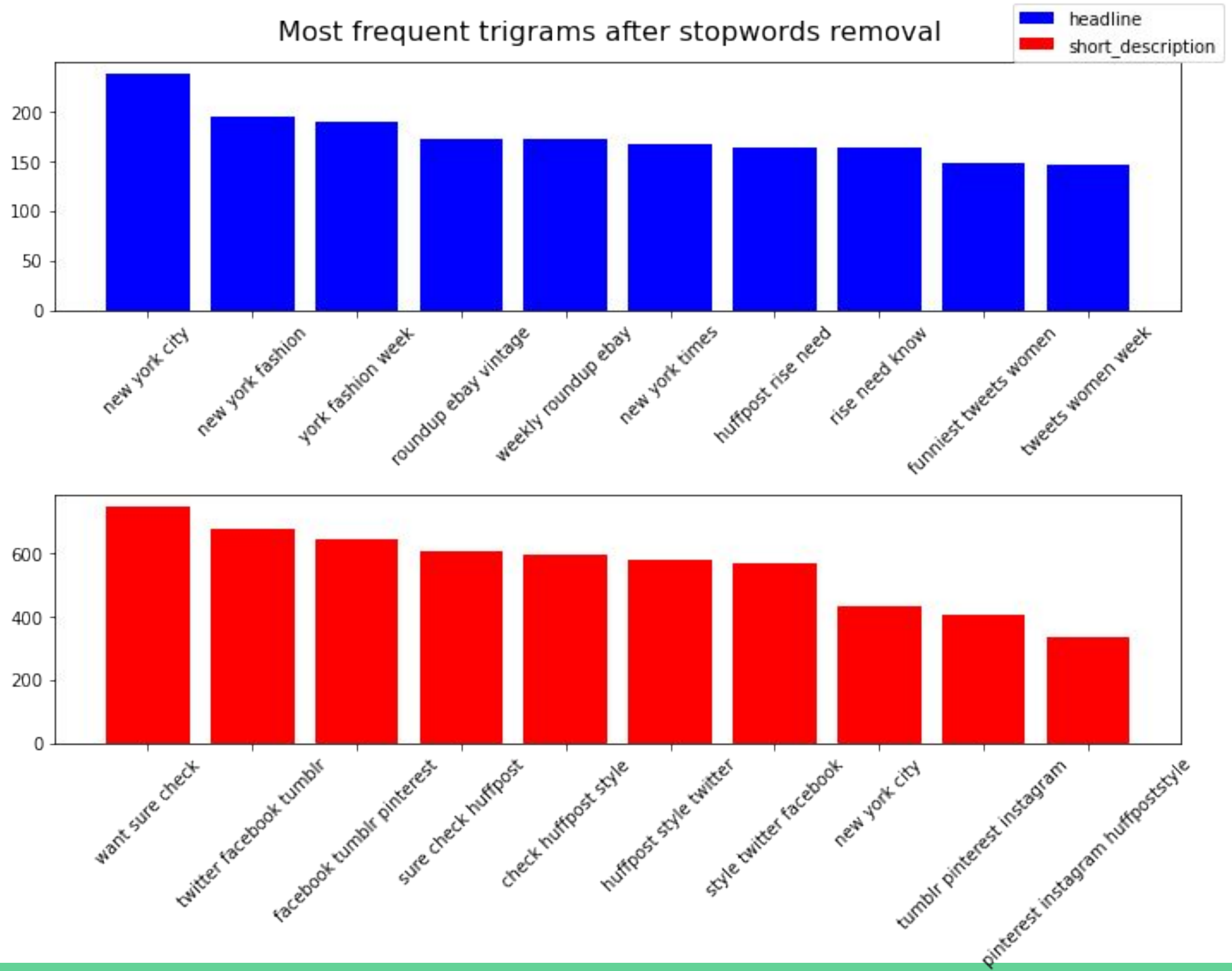
Datenanalyse



Daten-analyse

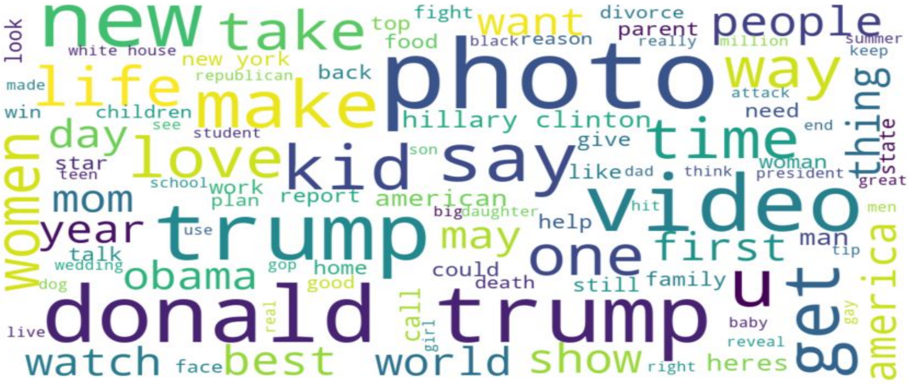


Daten-analyse

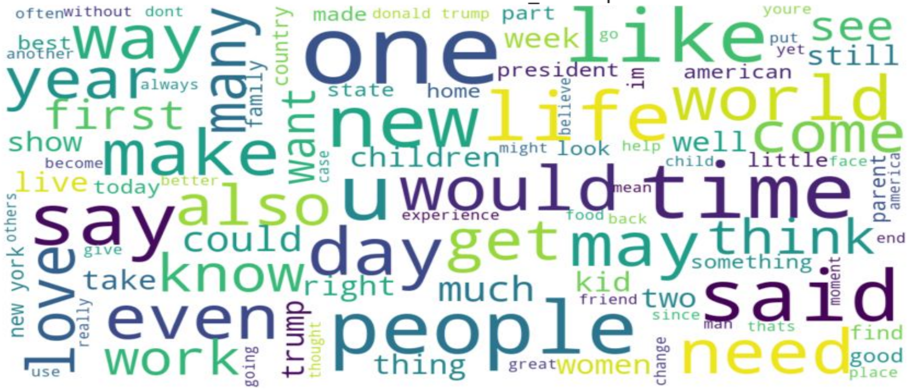


Datenanalyse

Wordcloud for field: headline



Wordcloud for field: short_description



Datenanalyse

- viele Kategorien mit wenig Einträgen
  - “Weird news”
  - “Green”
  - “Fifty”

- Verwechselbare Kategorien
  - “Money” <> “Business”
  - “World news” <> “World post”

➔ Reduktion auf Kategorien mit min. 4000 Artikeln

```
PARENTS           3955
THE WORLDPOST     3664
WEDDINGS          3653
WOMEN             3572
CRIME             3562
IMPACT            3484
DIVORCE           3426
WORLD NEWS        3299
MEDIA             2944
WEIRD NEWS        2777
GREEN             2622
WORLDPOST         2579
RELIGION          2577
STYLE             2254
SCIENCE           2206
TECH              2104
TASTE             2096
MONEY             1756
ARTS              1509
ENVIRONMENT       1444
FIFTY             1401
GOOD NEWS         1398
U.S. NEWS         1377
ARTS & CULTURE    1339
COLLEGE           1144
LATINO VOICES     1130
CULTURE & ARTS    1074
EDUCATION         1014
Name: category, dtype: int64
```

Datenanalyse

```
There are 14 categories
POLITICS          35602
WELLNESS          17945
ENTERTAINMENT     17362
TRAVEL             9900
STYLE & BEAUTY     9814
PARENTING          8791
HEALTHY LIVING     6694
QUEER VOICES       6347
FOOD & DRINK       6340
BUSINESS           5992
COMEDY             5400
SPORTS             5077
BLACK VOICES       4583
HOME & LIVING      4320
Name: category, dtype: int64
```

- Trotzdem stark ungleiche Verteilung
- Countertechniques:
  - Oversampling: Duplizieren unterrepräsentierter Klassen
  - Undersampling: Entfernen überrepräsentierter Klassen
  - SMOTE: Oversampling mit Interpolation
  - Data augmentation: Oversampling mit Data Transformation

➔ Verwendung Data Augmentation mit Synonymersetzung

Datenanalyse - nach Data Augmentation

```
There are 14 categories
POLITICS          35602
WELLNESS          17945
ENTERTAINMENT     17362
COMEDY            10000
PARENTING         10000
SPORTS            10000
BUSINESS          10000
STYLE & BEAUTY    10000
FOOD & DRINK      10000
QUEER VOICES      10000
HOME & LIVING     10000
BLACK VOICES      10000
TRAVEL            10000
HEALTHY LIVING    10000
Name: category, dtype: int64
```

Datenvorverarbeitung

“The 19-year-old reportedly fled into the Thames in a failed bid to escape police.”

- **Conversion to lowercase / removing new lines leading/trailing white spaces and non-alphanumeric characters and digits …**

  ‘the 19-year-old reportedly fled into the thames in a failed …’]

- **Tokenization**

  [‘the’ , ‘19-year-old’ , ‘reportedly’ , ‘fled’ , ‘into’ , ‘the’ , ‘thames’ , ‘in’ , ‘a’ , ‘failed’ , …]

- **Removing Stopwords**

  [‘19-year-old’ , ‘reportedly’ , ‘fled’ , ‘thames’ , ‘failed’ , ‘bid’ , ‘escape’ , ‘police’]

- **Rejoining Words**

  [‘19-year-old report fle thames fail bid escap police’]

Modelltraining - Naive Bayes Classifier

- Annahme: Unabhängigkeit der Feature

➜ “naiver” Classifier

Modelltraining - Naive Bayes

1. **Text vectorization:**
   Umwandlung von Trainingsdaten in sparse matrix von Tokenanzahl (numerische Form)
2. **Transformer:**
   Misst Wichtigkeit von Wörtern
   anhand Häufigkeit

   Tokenanzahl → Tf-idf

3. **MultinomialNB:**
   Wendet transformierte Daten in
   NB Modell an

Modelltraining - Naive Bayes - Ergebnisse

- accuracy 61 %
- 14 Kategorien ➔ baseline 7 %
- Entertainment, Food + Drinks, Home+ Living besonders gut

accuracy 0.6089492012602952

	precision	recall	f1-score	support
BLACK VOICES	0.92	0.22	0.36	1990
BUSINESS	0.94	0.21	0.34	2014
COMEDY	0.87	0.24	0.37	1971
ENTERTAINMENT	0.61	0.75	0.67	3582
FOOD & DRINK	0.87	0.74	0.80	1995
HEALTHY LIVING	0.95	0.04	0.07	2022
HOME & LIVING	0.91	0.75	0.82	1991
PARENTING	0.84	0.35	0.49	1993
POLITICS	0.48	0.98	0.64	7000
QUEER VOICES	0.97	0.35	0.51	2003
SPORTS	0.91	0.56	0.69	2006
STYLE & BEAUTY	0.89	0.66	0.76	2011
TRAVEL	0.89	0.56	0.69	2014
WELLNESS	0.45	0.87	0.59	3590
accuracy			0.61	36182
macro avg	0.82	0.52	0.56	36182
weighted avg	0.75	0.61	0.58	36182

Modelltraining - Naive Bayes - Ergebnisse

accuracy 0.8400585926703886

	precision	recall	f1-score	support
BLACK VOICES	0.89	0.87	0.88	1990
BUSINESS	0.90	0.80	0.85	2014
COMEDY	0.86	0.80	0.83	1971
ENTERTAINMENT	0.81	0.79	0.80	3582
FOOD & DRINK	0.89	0.91	0.90	1995
HEALTHY LIVING	0.84	0.58	0.68	2022
HOME & LIVING	0.92	0.95	0.94	1991
PARENTING	0.79	0.72	0.75	1993
POLITICS	0.85	0.93	0.89	7000
QUEER VOICES	0.91	0.84	0.87	2003
SPORTS	0.93	0.93	0.93	2006
STYLE & BEAUTY	0.88	0.82	0.85	2011
TRAVEL	0.85	0.80	0.82	2014
WELLNESS	0.66	0.85	0.75	3590
accuracy			0.84	36182
macro avg	0.86	0.83	0.84	36182
weighted avg	0.85	0.84	0.84	36182

Modelltraining - Linear SVM

- SVM sucht beste Grenzen im feature space, um Klassen zu trennen
- SGDClassifier kann als lineare SVM genutzt werden

accuracy 0.6865844895251783

	precision	recall	f1-score	support
BLACK VOICES	0.71	0.47	0.56	1990
BUSINESS	0.76	0.48	0.59	2014
COMEDY	0.68	0.37	0.48	1971
ENTERTAINMENT	0.73	0.60	0.66	3582
FOOD & DRINK	0.72	0.83	0.77	1995
HEALTHY LIVING	0.69	0.12	0.20	2022
HOME & LIVING	0.73	0.82	0.77	1991
PARENTING	0.69	0.68	0.68	1993
POLITICS	0.62	0.94	0.75	7000
QUEER VOICES	0.81	0.72	0.77	2003
SPORTS	0.78	0.77	0.77	2006
STYLE & BEAUTY	0.71	0.79	0.75	2011
TRAVEL	0.77	0.72	0.74	2014
WELLNESS	0.62	0.72	0.66	3590
accuracy			0.69	36182
macro avg	0.72	0.64	0.65	36182
weighted avg	0.70	0.69	0.67	36182

Modelltraining - Linear SVM - Ergebnisse Hyperparameter

accuracy 0.6859764523796363

	precision	recall	f1-score	support
BLACK VOICES	0.71	0.48	0.57	1990
BUSINESS	0.76	0.48	0.59	2014
COMEDY	0.69	0.37	0.48	1971
ENTERTAINMENT	0.74	0.59	0.66	3582
FOOD & DRINK	0.72	0.83	0.77	1995
HEALTHY LIVING	0.68	0.12	0.20	2022
HOME & LIVING	0.72	0.82	0.77	1991
PARENTING	0.69	0.68	0.68	1993
POLITICS	0.61	0.94	0.74	7000
QUEER VOICES	0.82	0.72	0.77	2003
SPORTS	0.78	0.77	0.77	2006
STYLE & BEAUTY	0.72	0.79	0.75	2011
TRAVEL	0.77	0.71	0.74	2014
WELLNESS	0.61	0.73	0.66	3590
accuracy			0.69	36182
macro avg	0.72	0.64	0.65	36182
weighted avg	0.70	0.69	0.67	36182

➔ Verbesserung nur auf Software Engineering Ebene (bezieht auf die Größe im Vergleich zum vorherigen Modell)

Modelltraining - NN with BOW

- Fully connected neural network mit Bag of Words (BOW)
- BOW: Methode zur Feature extraction auf Textdaten

Modelltraining - NN with BOW - Parameter

```
# Set the maximum number of words to be included in the vocabulary
max_words = 1000
# Initialize the tokenizer
tokenize = text.Tokenizer(num_words=max_words, char_level=False)
# Fit the tokenizer only on the train text to create the vocabulary
tokenize.fit_on_texts(train_text)
```

Modelltraining - NN with BOW

```
model_nn = Sequential()
model_nn.add(Dense(512, input_shape=(max_words,)))
model_nn.add(Activation("relu"))
model_nn.add(Dropout(0.5))
model_nn.add(Dense(num_classes))
model_nn.add(Activation("softmax"))
```

```
model_nn.compile(loss="categorical_crossentropy",
                 optimizer="adam",
                 metrics=["accuracy"])
```

Modelltraining - NN with BOW - Ergebnisse

```
score = model_nn.evaluate(x_test, y_test,
                          batch_size=batch_size, verbose=1)
print("Test accuracy:", score[1])
```

```
1131/1131 [==============================] - 2s 1ms/step - loss: 1.4418 - accuracy: 0.4984
Test accuracy: 0.49842461943626404
```

Modelltraining - CNN with Embedding

- Convolutional Neural network
- Versteckter Vektor als Kurzzeitgedächtnis
- aber: langsamerer Berechnung

Modelltraining - CNN

```
# basline model using embedding layers and simpleRNN
model_cnn = Sequential()
# 50 represents the number of dimensions in the embedding space.
# This means that each word in the vocabulary will be represented by a vector of 50 numbers
model_cnn.add(Embedding(max_words, 50, input_length=maxlen))
model_cnn.add(Bidirectional(SimpleRNN(64, dropout=0.2, recurrent_dropout=0.20, activation="tanh", return_sequences=True)))
model_cnn.add(Bidirectional(SimpleRNN(64, dropout=0.3, recurrent_dropout=0.30, activation="tanh", return_sequences=True)))
model_cnn.add(SimpleRNN(32, activation="tanh"))
model_cnn.add(Dropout(0.4))
model_cnn.add(Dense(num_classes))
model_cnn.add(Activation("softmax"))
model_cnn.summary()
```

Modelltraining - CNN - Ergebnisse

```
score = model_cnn.evaluate(test_text_padseq, y_test,
                           batch_size=batch_size, verbose=1)
print("Test accuracy:", score[1])
```

```
566/566 [==============================] - 18s 33ms/step - loss: 2.1297 - accuracy: 0.2380
Test accuracy: 0.2380465418100357
```

```
Epoch 1/4
1810/1810 [==============================] - 383s 210ms/step - loss: 2.0961 - accuracy: 0.3623 - val_loss: 2.2464 - val_accuracy: 0.2494
Epoch 2/4
1810/1810 [==============================] - 379s 210ms/step - loss: 1.6415 - accuracy: 0.4908 - val_loss: 1.8901 - val_accuracy: 0.4020
Epoch 3/4
1810/1810 [==============================] - 376s 208ms/step - loss: 1.4393 - accuracy: 0.5544 - val_loss: 1.7075 - val_accuracy: 0.4968
Epoch 4/4
1810/1810 [==============================] - 368s 203ms/step - loss: 1.3237 - accuracy: 0.5879 - val_loss: 1.6945 - val_accuracy: 0.5156
```

Anwendungsentwicklung

- Entwicklung mit Streamlit Framework
- Für Python konzipiert
- Streamlit Community Hosting bis zu 1 GB
- Integrierte CI/CD Pipeline

Deployment

Keine Docker Deployment Spezifikation vom Betriebssystem, Python Version,... nicht möglich

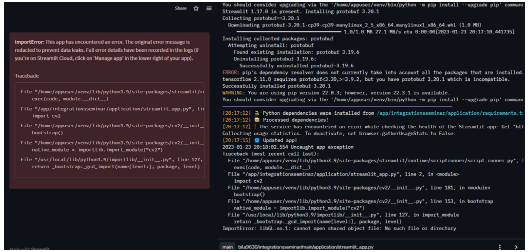
```
tensorboard==2.11.2
tensorboard-data-server==0.6.1
tensorboard-plugin-wit==1.8.1
tensorflow==2.11.0
tensorflow-estimator==2.11.0
tensorflow-intel==2.11.0
tensorflow-io-gcs-filesystem==0.30.0
```

```
scikit-learn==1.2.0
scipy==1.10.0
threadpoolctl==3.1.0
tqdm==4.64.1
tensorflow
keras
keras_preprocessing
```

=> Lösung: Keine Versionen zu den Paketen nennen

Modell einlesen dauert lange

```
@st.cache(allow_output_mutation=True)
def load_model_path():
    model_decision_tree = pickle.load(
        open("application/decision_tree.pkl", "rb"))
    model_cnn_1 = load_model("application/cnn_model_1.h5")
    model_cnn_2 = load_model("application/cnn_model_2.h5")
    model_cnn_3 = load_model("application/cnn_model_3.h5")
    model_rf = pickle.load(open("application/model_rf.pkl", "rb"))
    model_xgb = xgb.XGBClassifier()
    model_xgb.load_model("application/model_xgb.txt")
    return model_decision_tree, model_cnn_1, model_cnn_2, model_cnn_3, model_rf, model_xgb
```



=> Lösung: Caching und Daten in richtiger Format speichern

Anwendungsoberfläche

Category Naives Bayes: PARENTING

Category SVM: PARENTING

Category NN: PARENTING, with probability: 0.98

Category CNN: WELLNESS, with probability: 0.49

Welcome to the raving reporters!

Simply enter a news title and description and we'll classify it for you!

Newspaper title

Talking to kids when they need help

Newspaper description

As parents and teachers, you are the first line of support for kids and teens. It’s important for you to have an open line of communication with them and build a sense of trust. When your kids and teens are having difficulties, you want them to feel comfortable turning to you for help.

Classify

Anwendungsoberfläche

“5 savings mistakes people make when building their financial life”

Category Naives Bayes: WELLNESS

Category SVM: BUSINESS

Category NN: WELLNESS, with probability: 0.49

Category CNN: WELLNESS, with probability: 0.47

“Man describes disarming suspected Monterey Park gunman at second dance hall location”

Category Naives Bayes: POLITICS

Category SVM: POLITICS

Category NN: WELLNESS, with probability: 0.28

Category CNN: ENTERTAINMENT, with probability: 0.66

Fazit

- Zum Vergleich in unserer Anwendung alle zur Auswahl
- Für diese Anwendung “simples” Modell wie Naive Bayes besser
- NNs Accuracy könnte durch bessere Parameter erhöht werden
- Data Augmentation hat auch einen Einfluss auf die Accuracy

Live-Demo

SCAN ME

https://bit.ly/3XzzNdp

Vielen Dank

für eure Aufmerksamkeit

Quellen

Naive Bayes Explained. Naive Bayes is a probabilistic… | by Zixuan Zhang | Towards Data Science

Was ist der Naive Bayes Algorithmus? | Data Basecamp

1.4. Support Vector Machines — scikit-learn 1.2.1 documentation

Python Convolutional Neural Networks (CNN) with TensorFlow Tutorial | DataCamp

Convolutional Neural Networks (CNNs) and Layer Types - PyImageSearch

Streamlit • The fastest way to build and share data apps