Project Write-up – Predicting Newspaper Publishers based on Article Images and Teaser Texts

Abstract

The goal of this project was to use neural networks to predict newspaper publishers for article images and article teaser texts published online to more deliberately control images and introductory texts of published stories to draw audience interests. Additionally, the model may provide evidence for journalists which article images and teaser text characteristics are used by competitors. My analysis was based on images and teaser texts of 4 German publishers (F.A.Z., Spiegel Online, Zeit Online and SZ Online) published in the category “politics” on their main homepage Feb’22 to Jul’22. In feature engineering, I converted images and teasers to numerical arrays. In terms of modeling, features were fed to convolutional neural net, LSTM and several pre-trained models by using transfer learning. Finally, models were integrated into a web app on streamlit enabling to pass images and teaser texts to predict publishers.

Design

The project topic supports a question of data-driven journalism. Feature data was scraped from publishers’ homepages and from corresponding URL of each image. After conversion to numerical arrays, baseline non-deep-learning models and neural net models were constructed and their performance compared. Lastly, correctly and falsely classified images and texts of were inspected within and between publishers. Inspecting differences in images and texts may support journalists to deliberately choose images and texts characteristics. On a more global level it may also help to detect differences between publishers with different readerships such as right- or left-wing.

Data

The original dataset contains 49,333 images and teaser texts of 4 different publishers. To account for imbalances in the type of categories covered by different publishers, dataset was narrowed down to include only articles published in the category “politics”. The image dataset for multiclass-comparison of all 4 publishers’ images contained 9,511 records and the image dataset for binary-comparison of FAZ and Spiegel contained 5,105 records. The classification between teaser texts’ of FAZ and Zeit was based on 4,110 records.

Algorithms

Feature Engineering

1. Convert RGB images to common shape of 150x150x3 dimensions used as input dimension for NN models
2. Images were randomly transformed using ImageDataGenerator to increase models’ generalization capabilities
3. Texts were tokenized to remove special characters and keep only the most 1000 frequent words
4. Tokenized sentences were converted to sequence format and padded to maximum length of 50

Models

For image classification, a logistic regression model was trained on PCA-reduced 2 dimensional dataset forming a non-deep-learning baseline model. A standard sequential Convolutional Neural Net with 6 layers and 642,824 parameters was constructed as deep-learning baseline model. Subsequently, pre-trained models Xception, VGG16, mobilenet were imported using transfer learning and trained on publishers’ images. For text classification LSTM model was trained as the only deep-learning model.

Model Evaluation and Selection

The entire dataset was randomly split into training, validation and test data. Models were evaluated based on their generalization performance using standard classification metrics with a focus on F1-score to judge model based on its ability to predict positives correctly among all predicted positives (precision) and to correctly identify positives among all present positives (recall). For multiclass image classification, VGG16 model had best F1-score of 0.41. Based on this, VGG16 was used for binary image classification and reached an F1-score of 0.72. For binary text classification, LSTM model had an F1-score of 0.68.

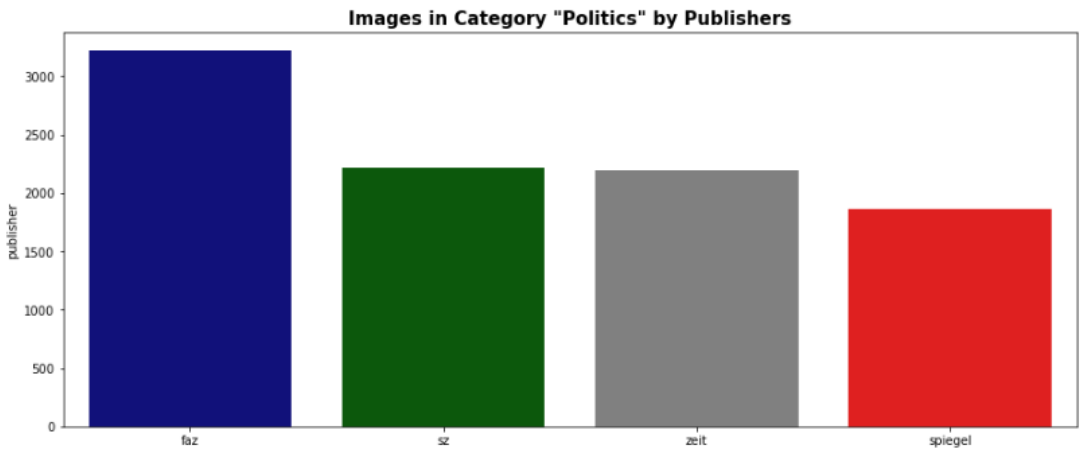
Tools

- BeautifulSoup for webscraping
- NumPy, pandas and keras for data manipulation

- NLTK for text processing
- Keras and sklearn for modeling
- Matplotlib, Seaborn and Bokeh for plotting
- Streamlit for creating a web application

Communication

In addition to the slides and visuals presented, the model is embedded in a dedicated streamlit app and project outputs are placed on my personal github.



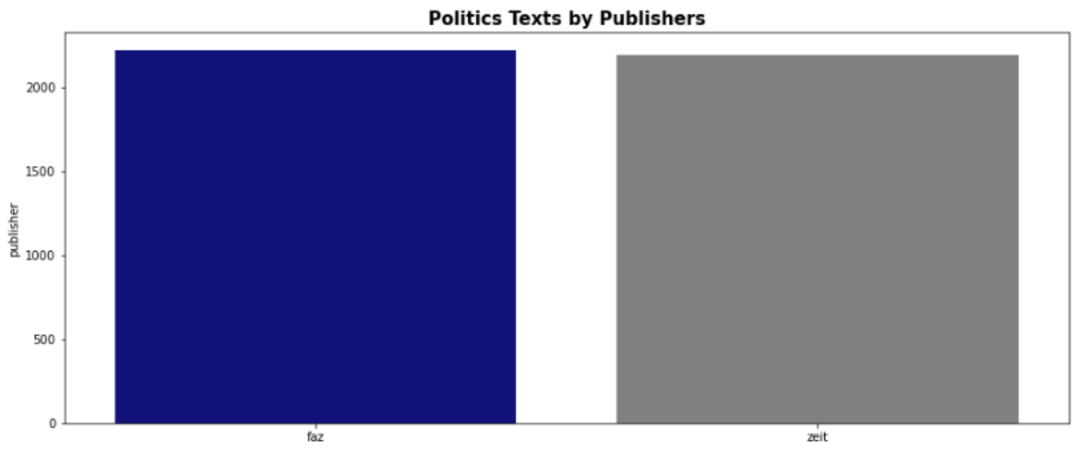
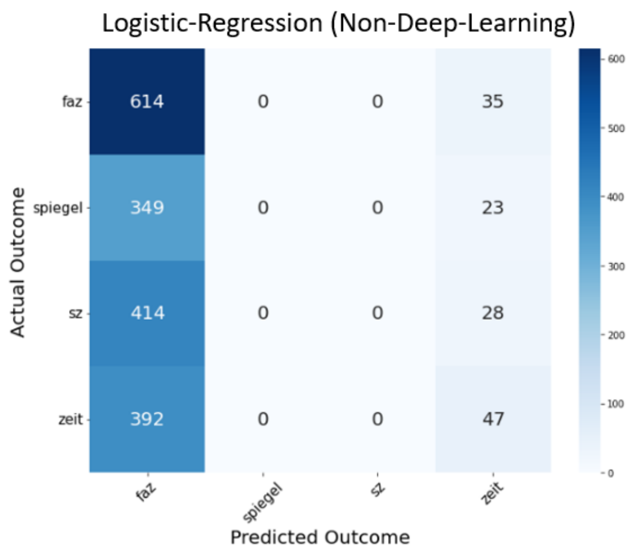
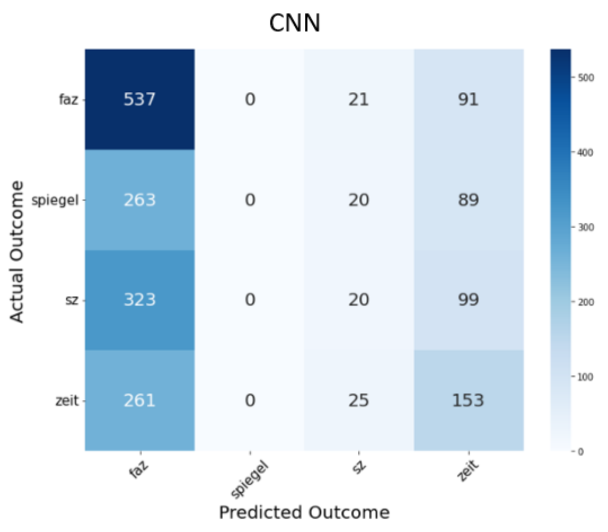


Image Multiclass classification – Baseline Model



	faz	spiegel	sz	zeit
faz	614	0	0	35
spiegel	349	0	0	23
sz	414	0	0	28
zeit	392	0	0	47



	faz	spiegel	sz	zeit
faz	537	0	21	91
spiegel	263	0	20	89
sz	323	0	20	99
zeit	261	0	25	153

Image Multiclass classification – Evaluation & Top Performing Model

	accuracy	recall	precision	f1	run_time
mobilnetv2_trainable_array.h5	0.341220	0.250000	0.085305	0.127205	12.658987
cnn_model.h5	0.387487	0.336131	0.358029	0.321290	3.814294
vgg16_model_array.h5	0.436909	0.412385	0.408664	0.406802	47.056915
Xception_model_array_trainable.h5	0.417981	0.401676	0.400933	0.397134	21.157846
Xception_model_array.h5	0.410620	0.368455	0.427191	0.345746	21.232836
mobilnetv2_base_array.h5	0.400105	0.397005	0.402469	0.389216	8.095628

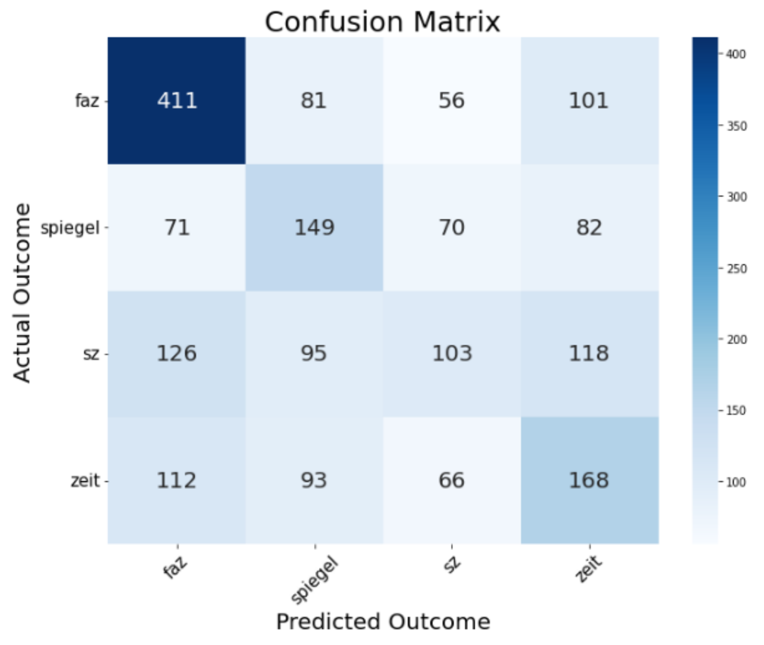
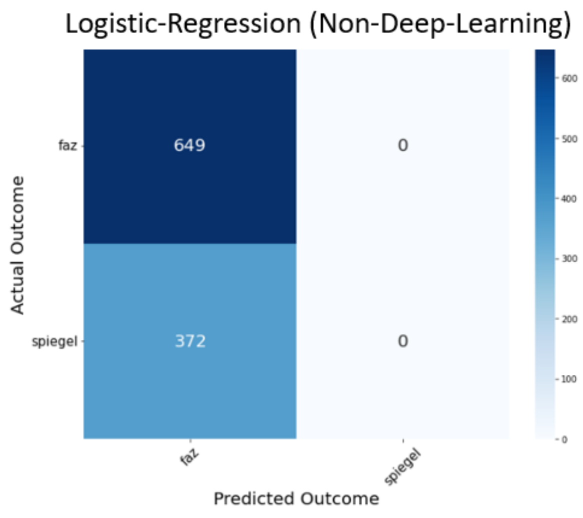


Image Binary classification – Baseline Model



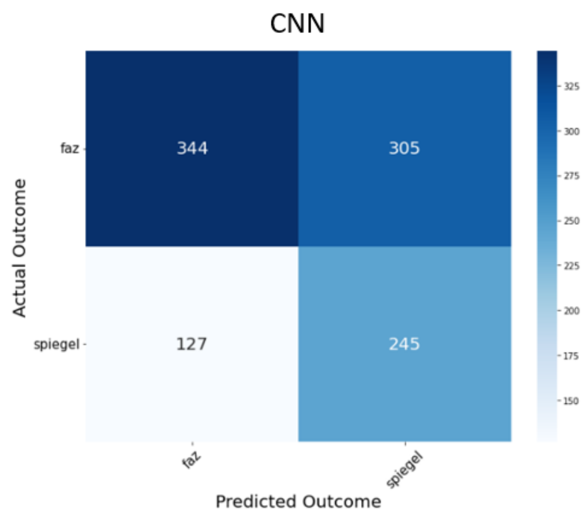
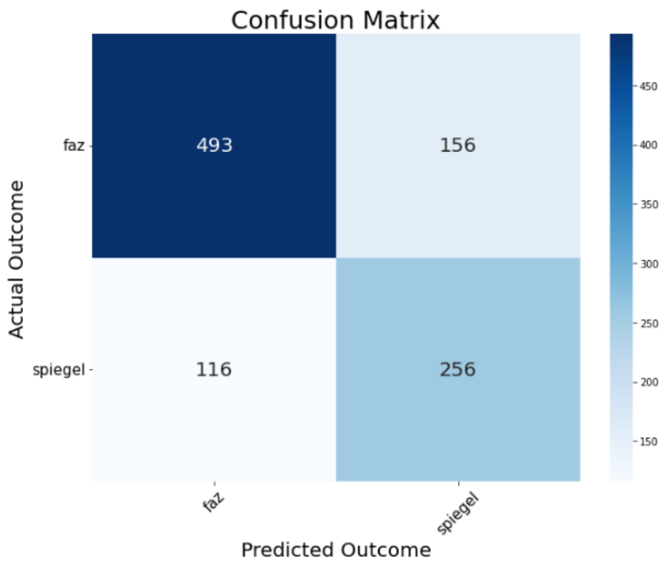
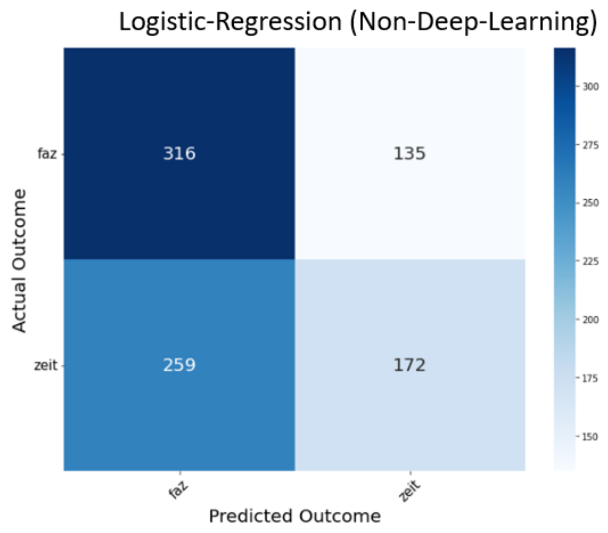


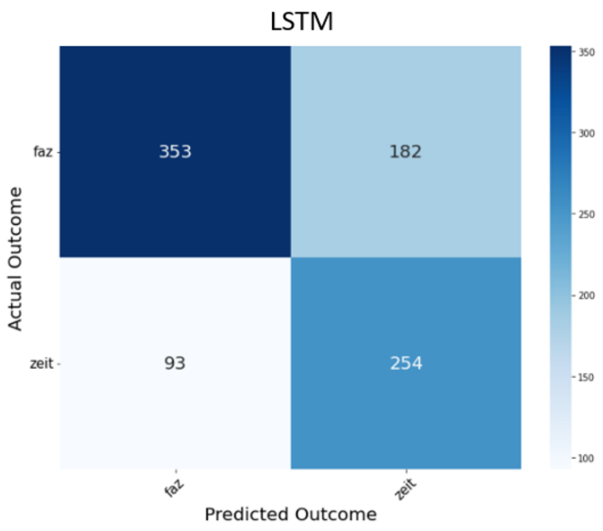
Image Binary classification – Evaluation & Top Performing Model

	accuracy	recall	precision	f1	run_time
cnn_model_fazsp.h5	0.676787	0.604066	0.648574	0.602658	2.178975
vgg16_model_array_trainable_fazsp.h5	0.732615	0.723704	0.714754	0.717792	24.136954
vgg16_model_array_fazsp.h5	0.733595	0.723901	0.715442	0.718423	24.378965



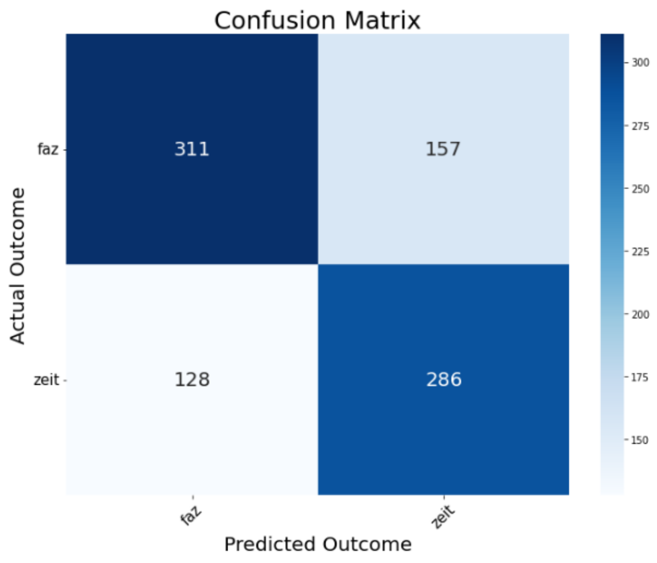
Text Binary classification – Baseline Model





Text Binary classification – Evaluation & Top Performing Model

```
accuracy : 0.6768707482993197
recall : 0.6770132200722965
precision : 0.6776755852842808
f1 : 0.6766109309287673
```


Confusion Matrix
faz
zeit
Actual Outcome
Predicted Outcome
faz
zeit