Project Write-up – Predicting Visits from topics of Digital Newspaper Articles

Abstract

The goal of this project was to combine topic modeling and regression approaches to forecast visits for newspaper articles published online to support editors in choosing which articles to publish. Additionally, the model may provide evidence for journalists which article topics and characteristics may be important for attracting readership. My analysis was based on articles of German publisher Frankfurter Allgemeine Zeitung published online 01/01/22 - 20/06/22 and their corresponding visit figures. In feature engineering, I leveraged article text data and meta information of articles itself to create a set of categorical and numerical features. In terms of topic modeling, I used LDA to feed linear regression, lasso, ridge, random forest regressor and gradient boosting regressor. Finally, model was integrated within a web app on streamlit enabling to pass article characteristics to generate visit predictions.

Design

The project topic constitutes a central question of current (data-driven) journalism. Feature data was scraped from a publically available newsticker website listing all historically published articles and from each articles' corresponding URL. The paywall was circumvented using cookie data of a digital subscription. Target data on article visits was provided from within the company. Extracting insights on article properties (such as topic, author, paid/free status or publication time) contributing to high reader numbers via machine learning models may support journalists and editors to reach larger audiences. It may help to evaluate article popularity ex-ante their publication online and thus assist in the choice of articles to be provided for-free or paid.

Data

The original dataset contains 26,538 articles with 106 features for each, 77 of which are categorical. A few feature highlights include "topics probabilities" derived from LDA, "free/paid", "author", "source weekday print", "publishing-time of day", "day of week" and "previous day visits". Approx. 75% of these features were created from specifications of more narrow categories (such as "author"). The feature set was refined by removing collinear and insignificant features with the final dataset consisting of 85 variables feeding the baseline model.

Algorithms

Feature Engineering

1. Create one-hot features for 15 authors with most and least visits and orders per article
2. Preprocess article texts (punctuation, lowering, stemming, stopwords), vectorize via TFIDF, fit an LDA model and choose number of topics
3. Creating bins reflecting time series features (such as time of day or day of week)
4. Creating bins reflecting article source (such as print weekday or news agency)
5. Converting categorical features to binary dummy variables (such as authors or department)
6. Log-transform count-data such as article visits to fulfill regression requirements of normal distribution of errors

Models

Linear regression, lasso, ridge, random forest regressor and gradient boosting regressor were used before settling on gradient boosting regressor as the model with strongest performance. Throughout, collinear and insignificant features were removed from the feature space.

Model Evaluation and Selection

The entire training dataset was split by publication date into individual months to conduct forward time series cross validation on individual months. As a result, all scores reported below were calculated with 4-fold cross validation. Predictions on articles with publication date in June were limited to the very end, so this split was only used and scores seen just once. Models were evaluated based on their generalization performance using R^2, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The gradient boosting regressor had a R^2 of 0.61 on the test sample versus a mean R^2 of 0.60 on the 4-fold CV sample.

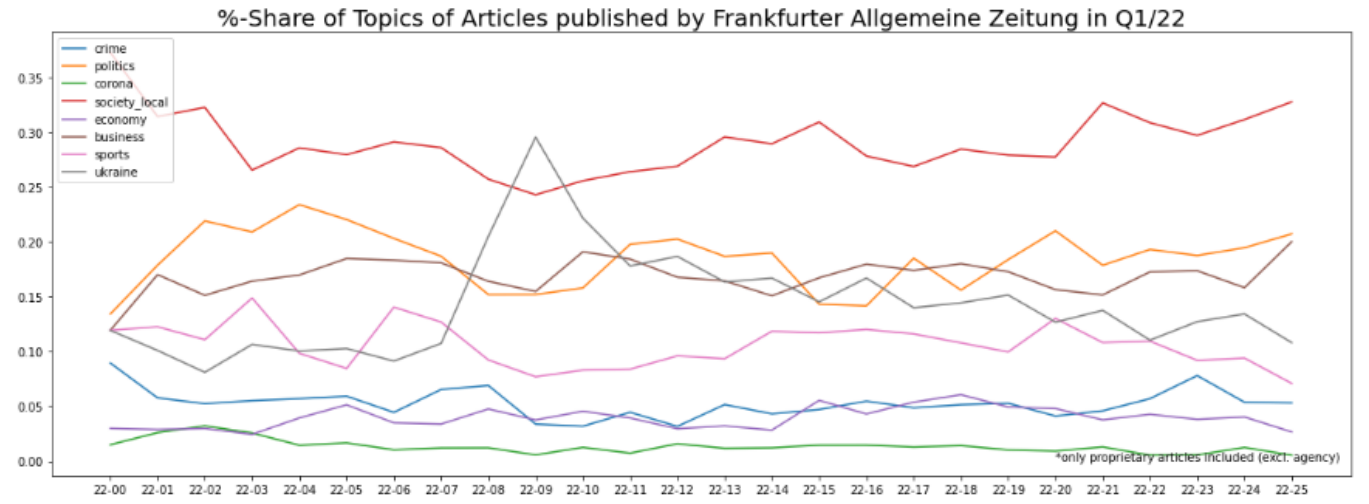
Tools

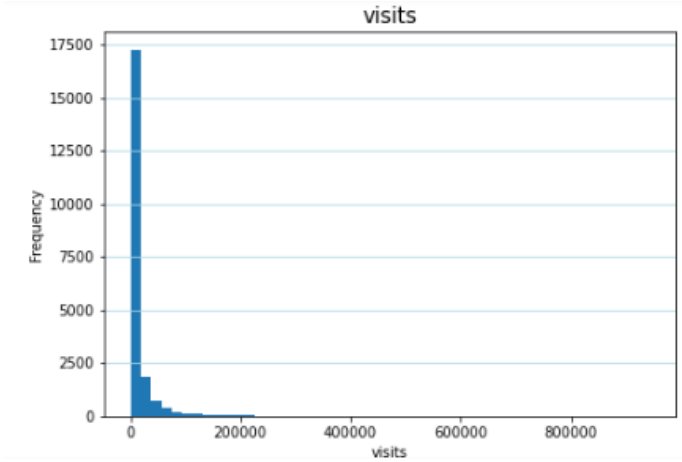
- BeautifulSoup and Selenium for webscraping
- NumPy and pandas for data manipulation

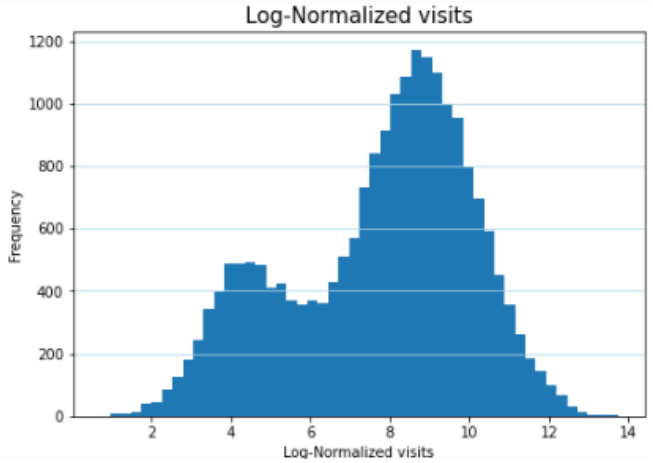
- NLTK and genism for text processing and topic modeling
- Pillow for image feature generation
- Statsmodels and scikit-learn for modeling
- Matplotlib and Seaborn for plotting
- Streamlit for creating a web application

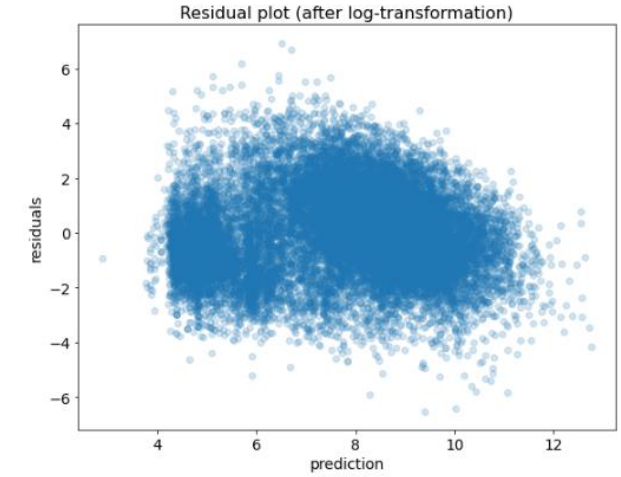
Communication

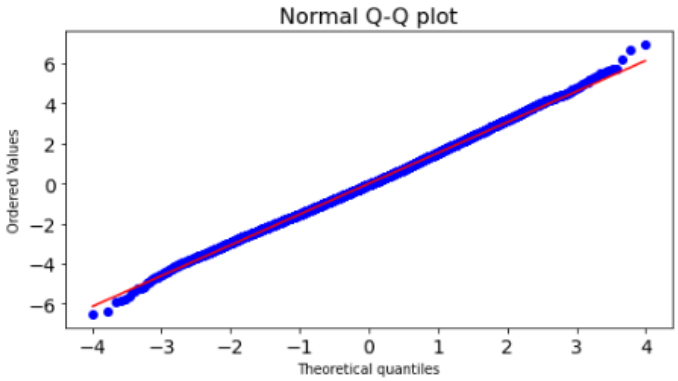
In addition to the slides and visuals presented, the model is embedded in a dedicated streamlit app and project outputs are placed on my personal github.

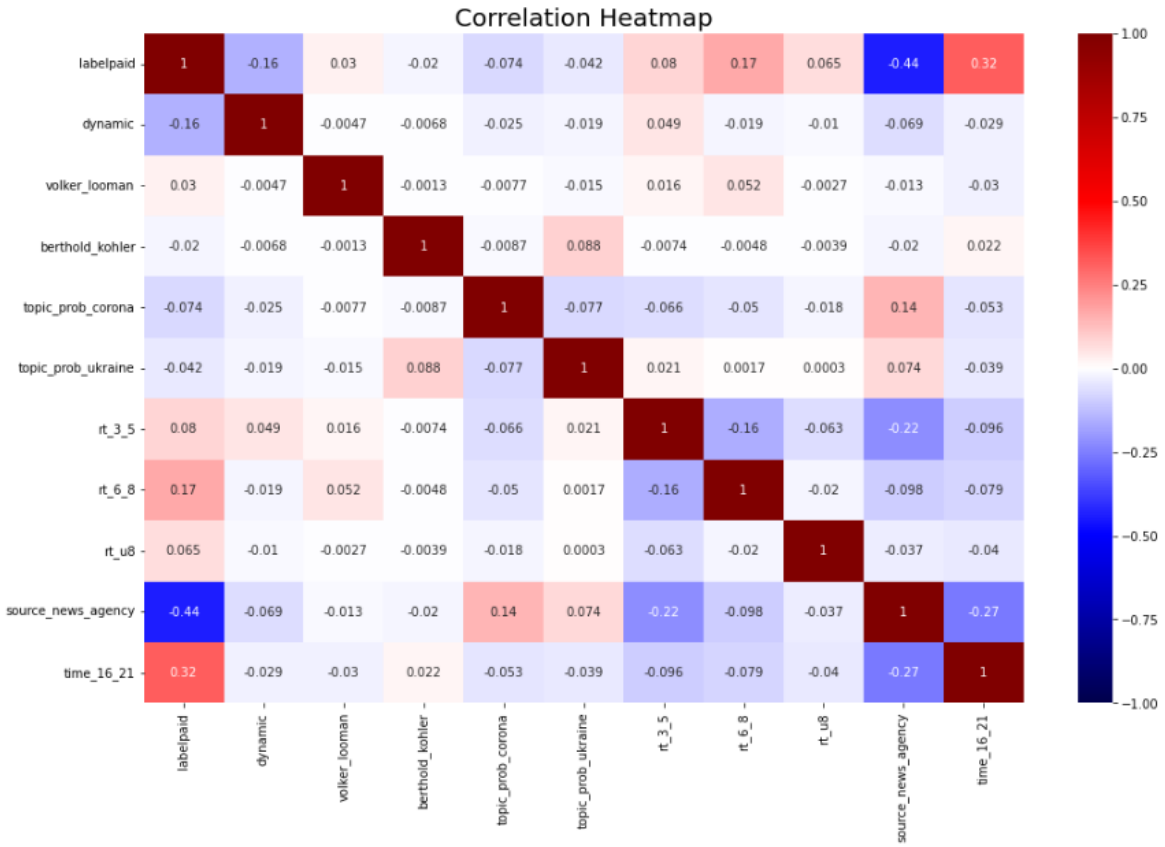


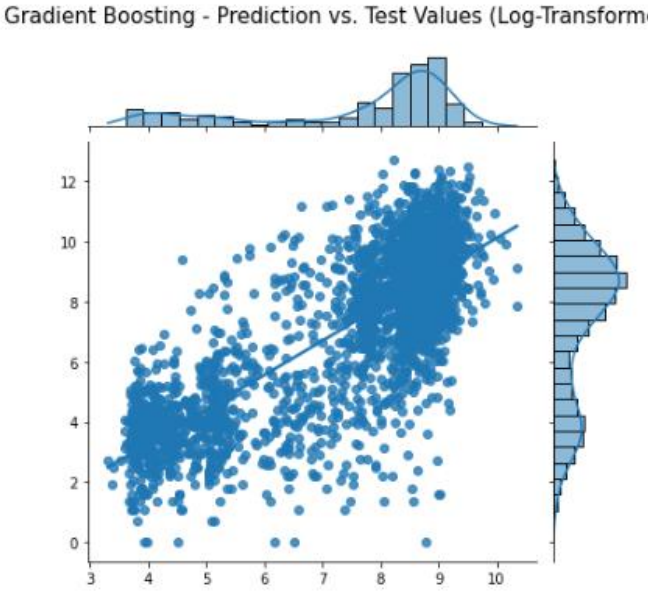













```
Linear Regression Mean Average Error (MAE): 1.3721
Linear Regression Root Mean Squared Error (RMSE): 1.7547
Linear Regression R2 Score (R2): 0.5258

Lasso MAE: 1.3860
Lasso RMSE: 1.7712
Lasso R2: 0.5168

Ridge MAE: 1.3762
Ridge RMSE: 1.7610
Ridge R2: 0.5224

Random Forest Regressor MAE: 1.2775
Random Forest Regressor RMSE: 1.6702
Random Forest Regressor R2: 0.5704

Gradient Boosting MAE: 1.2040
Gradient Boosting RMSE: 1.5890
Gradient Boosting R2: 0.6111
```