Project Write-up – Increase publisher website traffic by article & publication features

Abstract

Goal of the project is to perform an analysis of article content of two leading German news publishers Frankfurter Allgemeine Zeitung (F.A.Z.) and Süddeutsche Zeitung (SZ) and give recommendations to journalists and editorial board on how to adjust the article publication behavior and article characteristics itself to increase website traffic and ultimately subscribers. The analysis is performed by identifying weeks with high website traffic and comparing article features of these weeks to those with lower website traffic. Furthermore, a direct comparison of article features between both publishers was conducted. Also, a regression model was trained and significantly correlated variables identified using daily website traffic as. In terms of modelling I used linear regression, lasso, ridge, polynomial regression, random forest regressor and gradient boosting regressor.

Design

The project topic tackles a central question of publisher F.A.Z. which faces lower paid content subscription growth than main competitor SZ. The hypothesis is that identifying drivers of traffic (readership) will help journalists and editors increase readership by adjusting article and publication characteristics. Feature data was scraped from a publicly available newsticker website listing all historically published articles and from each articles' corresponding URL. Target data (low and high website traffic) was provided by industry association website which regularly publishes visit data. Extracting insights on article properties (such as %-paid articles, length, publication time, or keywords) contributing to high reader numbers may support journalists and editors to reach larger audiences.

Data

The dataset contains 37,485 articles, 119 days of website traffic and 187 numerical features for each day. However, descriptive comparisons were conducted based on 9 categories of features which include "article departments", "article source", "Weekday", "Time of Day", "Paid vs Free", "Comment Article", "Length", "Author" and "Keywords". For the regression model, each feature was investigated more closely with 6 of them forming baseline models.

Algorithms

Feature Engineering

1. Removing duplicate article data
2. Generating department information from URLs
3. Extracting 20 most used keywords from keywords published with each article
4. Creating bins reflecting time series features (such as time of day or day of week), article source (such as print weekday or news agency), paid/free or article length (from its indicated reading time)
5. Log-transform visit-data

Descriptive

Weeks with low traffic were compared to those with high traffic for (a) before start of Ukraine war; (b) after start of the war; (c) low/high traffic weeks identified by difference-in-differences approach between F.A.Z. and SZ. Additionally, general article and publication characteristics were compared between the two publishers.

Models

Linear regression, lasso, ridge, elastic net, polynomial regression, random forest regressor and gradient boosting regressor were used. Throughout, linear regression was used to check on p-values of each feature to ensure significance.

Model Evaluation and Selection

The training dataset of 119 days was split into 90/10 train vs. test, and all scores reported below were calculated with 10-fold cross validation on the training portion only. Predictions on 10% test 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). Simple linear regression had a R^2 of 0.63 on the test sample versus a mean R^2 of 0.43 on the 10-fold CV sample.

Tools

- BeautifulSoup for webscraping and Selenium for automatic download of files
- Microsoft Excel for Exploratory Data Analysis (EDA)
- Microsoft Excel for generating descriptive statistics
- Tableau and Excel for visualization
- NumPy and pandas for data manipulation
- Statsmodels and scikit-learn for modeling
- Matplotlib and Seaborn for plotting

Communication

In addition to the slides and visuals presented, project outputs will be embedded on my personal github and dashboard pages on Tableau Online.