Political Bias of Newspapers

Dave McRae

2022-11-22

Executive Summary and Introduction

This report presents an attempt to undertake supervised classification of newspaper articles covering the 2022 Australian Federal election using measures of their political bias to identify each article's publication outlet. To simplify the classification task, we define bias as coverage favouring either the Australian Labor Party (ALP) or its main opponent the Coalition (a grouping of the Liberal Party and the National Party), disregarding minor parties and independent candidates. We assemble a dataset of 1001 newspaper articles published during the election campaign period from 5 media outlets, which satisfy the additional condition that every article must contain at least 10 sentences that mention a political party, a party member or a candidate from these two political groupings.

As we lack resources to undertake manual coding of bias, we exclusively use computational measures, namely the average sentiment per article of sentences discussing the ALP and the Coalition, the average sentiment of the headlines of these articles, and full text classification using sentences that discuss the ALP and/or the Coalition. Classification employs three algorithms suited to multi-class classification: Support Vector machine, Naive Bayes and K-Nearest Neighbours.

We set the performance benchmark for these algorithms to be the prevalence of the most frequently occurring publication outlet in our train set, which is *The Australian* at 36.2%, which Kuhn and Johnson (2016) term the 'no information rate'. Using our most direct measure of political bias, only the K-Nearest Neighbours method exceeds the no information rate, and only modestly so. Visualisation of sentiment-based bias measures explains this modest performance, as with the exception of a few outliers, there is little separation between classes. SVM and Naive Bayes algorithms trained on the full text sentences mentioning the ALP and the Coalition perform better at classification, but we cannot demonstrate unequivocally that this classification relies on a measure of bias. Nevertheless, the SVM algorithm in particular performs very well on the political sentence text, achieving accuracy of 66.7%.

These results leave us with an ambiguous conclusion - it is clearly possible to use machine learning techniques on the text of articles discussing the federal election to identify their publication outlet, but we remain unsure whether this classification reflects differing biases. It may be that perceptions of distinctive political bias on the part of these media outlets are overstated. Equally, though, a combination of manual coding and computational measures may be needed to generate better features to enable classification based on political bias.

Background

There is a widespread perception that different print media outlets in Australia have different political orientations (Gans and Leigh 2012). Such bias, if present, may variously reflect the politics of the media's owners, their editors and journalists, or their intended readership. Table 1 summarises the presumed ideological bias of five major print media outlets in Victoria, Australia's second most populous state and the author's place of residence.[1]

[1] I describe these outlets as print media/newspapers, because they produce written text articles. *ABC News*, however, is the online-only print arm of the public broadcaster *ABC*, which also operates a television station and radio. *The Guardian* -

Table 1: Presumed ideological positioning of media outlets

Publication	Owner	Ideology
ABC News	Public Broadcaster	Centre Left
The Age	Nine Media	Centre Right
The Australian	News Corp	Right
The Guardian - Australia	Guardian Media Group	Left
Herald Sun	News Corp	Right

Perceived media bias is particularly contentious during election campaigns. News Corp-owned outlets in particular have fielded accusations of hostility to the Australian Labor Party (ALP), with former ALP Prime Minister Kevin Rudd petitioning in 2020 for a royal commission to investigate News Corp's attacks on its political opponents.[2] In this report, we seek to investigate political bias on the part of Victoria's print media during the 2022 Australian federal election campaign. We confine our investigation of bias to Australia's main political cleavage, between the left-wing ALP and the right-of-centre Coalition, a grouping of the Liberal Party and the rural-based National Party. The Coalition lost government to the ALP in this 2022 election: the ALP won 77 of 151 seats in the House of Representatives, the Coalition 58 seats, with minor parties and independent candidates winning the remaining 16 seats. Our chosen method is machine learning, specifically supervised classification. In plain terms, we investigate whether we can train an algorithm to predict which outlet an article was published in, based on measures of political bias of the article's text.

This report comprises four sections. In this initial section, we introduce our dataset of media articles and the measures of political bias we employ. Next, a methods section explains our approach to supervised classification. Third, a results section presents the outcomes of our tuning of our classification model parameters and the accuracy of our classification outcomes. Finally, in conclusion, we offer some brief observations on the effectiveness of this approach and possibilities for further enhancement.

We assemble our dataset for this report through an API query to obtain Guardian Australia articles, and a keyword search of the Factiva database for the remaining four publications. In each case we use "election" as our keyword, and collect articles published from Sunday 10 April - the day then Prime Minister Scott Morrison announced the election - until Friday 20 May - the day prior to polling day. After initial data-cleaning to remove duplicate articles, articles with fewer than 100 words, and live blog articles, we obtain an initial dataset of 3090 articles.[3]

Our approach to generating measures of bias relies in part in identifying sentences within these articles that discuss the ALP or the Coalition. We estimate a minimum of ten such sentences are required for us to be able to classify each article. This threshold also serves to eliminate a proportion of articles that contain the keyword "election" but do not in fact discuss federal politics. Manual examination of the initial dataset found our "election" search query had captured many articles discussing other topics that make only offhand mention of the federal election, or which discuss state politics or elections in other countries. We thus employ the SpacyR package to divide each article into constituent sentences and to extract so-called entities (names of people or organisations) from the text. SpacyR - which provides an R wrapper to the Python package Spacy - was preferred to alternative R package suchs as quanteda, because the authors of the two packages describe SpacyR as providing smarter tokenisation.[4]

After manual coding of relevant entities as either ALP-aligned or Coalition-aligned, 1001 articles remain that include at least ten sentences we can identify as discussing the ALP or the Coalition (Table 2). Entities were preferred to keywords or specific n-grams (e.g. bigrams or trigrams) because they both eliminate the

Australia is an online-only news website operating as the Australian edition of *The Guardian*.

[2]https://www.aph.gov.au/e-petitions/petition/EN1938

[3]Articles with fewer than 100 words are presumed too short to classify. Live blogs are removed because of an inconsistency in how different publications post live blogs to databases. Some publications post them as individual articles of fewer than 100 words, whereas others aggregate all live blog posts for the day into a single article. We delete this latter category of posts for consistency across publications.

[4]https://spacyr.quanteda.io/articles/using_spacyr.html

ambiguity inherent in single word keywords and allow flexibility in the length of the string of words used to detect relevant sentences. As an example of the ambiguity problem of keywords, “Andrews” could denote either Coalition government Home Affairs Minister Karen Andrews (relevant) or Victorian State Premier Daniel Andrews (irrelevant).

Table 2: Articles from each media outlet in dataset

Publication	No of articles
The Australian	362
Guardian Australia	239
The Age	230
Herald Sun	87
ABC News	83

The following table shows the most frequently occurring ALP and Coalition entities.[5] For the ALP< we see different forms of the party name, different forms of party leader and now prime minister Anthony Albanese’s name, his deputy Jim Chalmers, Albanese’s predecessor as party leader Bill Shorten, and former prime minister Kevin Rudd. For the Coalition, half of the top ten entities are different forms of party names, with the remainder being different forms of then prime minister Scott Morrison’s name, then treasurer Josh Frydenberg’s name, as well as former Coalition prime minister John Howard.

Table 3: Most frequently occurring ALP and Coalition Entities

Alp Entities	No	Coalition Entities	No
Labor	8594	Morrison	4448
Albanese	3300	Scott_Morrison	1839
Anthony_Albanese	1360	Liberals	810
ALP	536	Frydenberg	507
Albo	365	the_Liberal_Party	435
the_Labor_Party	213	LNP	362
Bill_Shorten	170	Liberal	308
Kevin_Rudd	154	Josh_Frydenberg	304
Jim_Chalmers	151	Howard	278
Opposition	131	Nationals	274

Methods

We choose overall accuracy as our metric to evaluate model success. Accuracy is a suitable measure as there is no imbalance in importance between false positives and false negatives, as would be the case in disease detection for example.[6] Kuhn and Johnson (2016, p. 255) propose using a ‘no information rate’ equal to the frequency in the training set of the largest class as a minimum benchmark for model adequacy. If we can achieve greater than 36% accuracy in classification using justifiable measures of political bias, therefore, we will conclude the publications are distinguished by differing political bias to more than a negligible extent.

Our stages of analysis are as follows:

1. We partition the 1001 articles in our dataset into a 90 percent train set and 10 percent test set, and then repeat the partition just on the train set to obtain a 81% train set and 9% probe set. The test set is set aside for final testing only of our chosen classification model.

[5] We insert the “_” during entity extraction, these are not present in the original document.
[6] Kuhn and Johnson 2016, p. 255

2. We first investigate the feasibility of classification by publication outlet using a Naive Bayes and a Support Vector Machine algorithm on the full text of the articles in our dataset. These two algorithms are implemented in the quanteda.textmodels package as textmodel_nb and textmodel_svm.[7] For Naive Bayes, we run two versions of the model: an unadjusted model that assumes all classes are equally distributed, and a model tuned with the "prior" parameter to adjust predictions based on the actual prevalence of each publication in our train set. Each operates on a quanteda document frequency matrix (dfm), a data object that records the frequency of individual words in each document and the dataset as a whole, irrespective of word order. Stopwords and punctuation are removed prior to generating the dfm; so too are urls and a list of keywords corresponding to the publication names, lest our models use these to classify the articles. The 'bag of words' approach is not a measure of political bias, but a result above the 'no information rate' would demonstrate that classification of the articles based on some measure derived from their text is possible.

3. We next attempt classification by the first (and most direct) of our political bias measures, the sentiment of sentences discussing the ALP and the Coalition respectively. We first calculate sentiment per sentence using the sentimentR package,[8] preferred because it takes account of negators when calculating sentiment (Naldi 2019). For classification, we prepare three sets of sentiment scores for each document in the dataset:

   i) Average sentiment of sentences mentioning ALP and mentioning Coalition in each document - sentences mentioning both sides are included in the calculation of both averages. As our prediction algorithms cannot handle missing values, if a document does not discuss one side, its score is set as 0.
   ii) A date-adjusted average of sentences mentioning ALP and mentioning Coalition in each document - we divide the dataset into weekly date bins, calculate the weekly average of the sentiment scores in (i), and then generate a date-adjusted sentiment scores for each document calculated as: date-adjusted ALP sentiment score = (average ALP sentiment score of all documents published the same week) - (document's ALP sentimente score). This measure aims to capture media who are outliers from time-dependent trends in sentiment during the campaign (e.g a publication that maintained positive reportage about the ALP when other publications were running negative coverage of a campaign development.)
   iii) A combination of both the raw and date-adjusted sentence sentiment scores

We then use three algorithms suited to multi-class classification on these sentiment scores: Naive Bayes, Support Vector Machine, and K Nearest Neighbours. We implement all of these algorithms using the Caret package,[9] using the "svmLinear", "naive_bayes" and "knn" methods respectively. We specify the following tuning of these models:

i) For the Support Vector Machine, we tune the cost parameter, trying values between 0.1 and 1 in 0.1 increments.[10]
ii) For K Nearest Neighbours, we tune the values of k. We anticipate the optimum value will be the odd number closest to the square root of the number of documents in our training set[11] - 28.4 - and consequently trial odd numbers ranging from 5 to 31.
iii) For Naive Bayes, we do not use Laplace smoothing, but try the kernel set as True and False respectively, and try adjust values between 0.5 and 5 in 0.5 increments.

4. We next include a less direct measure of political bias - the sentiment of the headlines of the articles. We again use the same classification algorithms with the same range of tuning paramaters as for the sentiment of sentences, training these algorithms on the sentiment scores for the headlines in combination with the raw sentence sentiment scores.

[7] https://github.com/quanteda/quanteda.textmodels The package provides a tutorial on the requirements to implement these models, using the Naive Bayes model as an example: https://tutorials.quanteda.io/machine-learning/nb/

[8] https://cran.r-project.org/web/packages/sentimentr/index.html

[9] https://cran.r-project.org/web/packages/caret/vignettes/caret.html

[10] Originally we trialled 0 to 1, but 0 throws up erro messages as no support vectors are found.

[11] https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb

5. We then proceed to our final - and least direct - measure of political bias is to perform full text classification on the sentences that we have identified as discussing the ALP and the Coalition. We produce three different datasets by collapsing all of the political sentences from each document into a single text:

   i) ALP corpus - only the sentences coded as mentioning the ALP are included
   ii) Coalition corpus - only the sentences coded as mentioning the Coalition are included
   iii) Political sentences corpus - all sentences coded either as mentioning the ALP and the Coalition are included.

We first again use the quanteda.textmodels package to perform Naive Bayes and SVM classification on this text, to maximise comparability with our full article text measures. The authors of quanteda.textmodels have implemented these algorithms in a way tailored to the fast and accurate classifcation of text, when compared to more generic implementations available through the Caret package.[12] For our purposes, however, a key limitation of quanteda.textmodels package is that it lacks a method to extract the most important features the algorithms have used to classify the articles, which might enable us to make a judgment on how directly this classification derives from political bias.[13]

6. We train our best performing model on the full train set and test it on the previously set aside test set. As this model employs an indirect measure of political bias, we also use keyness frequency from the quanteda.textstats[14] and quanteda.textplots[15] packages to attempt to approximate the features that may be contributing to classification to check for their relevance to bias.

Results

Full text classification

As detailed in methods, we train a Support Vector Machine and a Naive Bayes algorithm on the full text of our train set articles, and use these models to classify the probe set. Both models perform well beyond the 'no information rate'. The SVM model achieves an accuracy of 77.2% on the probe set. Naive Bayes performs less well, but is still well above the 'no information rate threshold', both when used unadjusted at 58.7% and when adjusted to take account of the prevalence of publication outlets in our dataset, at 58.7% accuracy. Note that in both cases we apply these algorithms only after removing publication names and URLs from the bag of words - the SVM performed significantly better and Naive Bayes a little bit worse before these features were removed, suggesting that the algorithms were making use of them in their classification prior to their removal. We do not attempt to tune either model further, as these results are sufficient to show that it is possible to classify the articles based on information derived from their text.

Sentence-wise sentiment classification

We train a Naive Bayes, Support Vector Machine and K-Nearest Neighbours algorithm on our train set, and attempt to use these algorithms to classify the probe set. We run these algorithms on three different sets of features: i) the raw average sentiment scores for sentences mentioning the ALP and the Coalition. ii) the date adjusted sentiment scores, which subtract the raw scored from the weekly average. iii) a combination of the raw and date-adjusted sentiment scores

Table 4 summarises our results:

[12]See answer by Kenneth Benoit on this point in this Stack Overflow post: https://stackoverflow.com/questions/54427001/naive-bayes-in-quanteda-vs-caret-wildly-different-results

[13]See https://github.com/quanteda/quanteda.textmodels/issues/10 for a user request for this feature

[14]https://cran.r-project.org/web/packages/quanteda.textstats/index.html

[15]https://cran.r-project.org/web/packages/quanteda.textplots/index.html

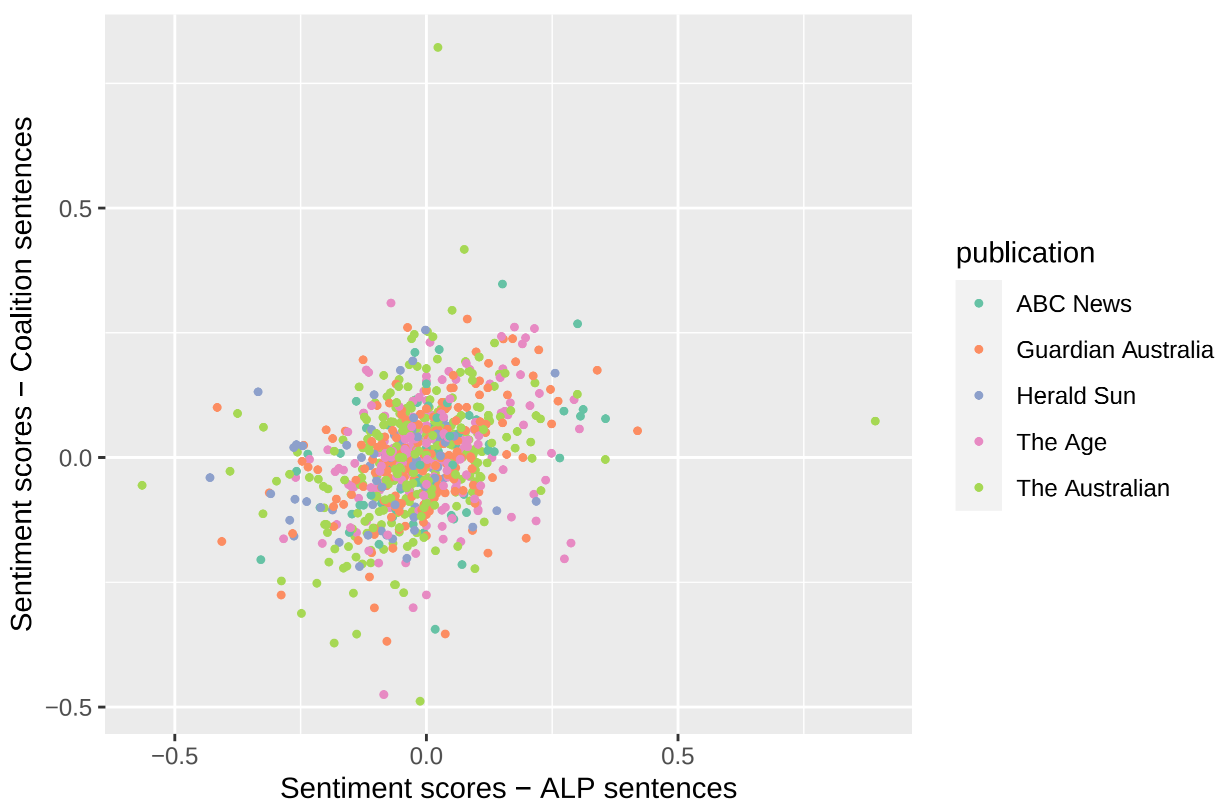
Table 4: Prediction results using sentence sentiments

Features	K-Nearest Neighbours	Support Vector Machine	Naive Bayes
Raw sentiment scores	0.4130435	0.3586957	0.3586957
Date-adjusted sentiment scores	0.3586957	0.3586957	0.3586957
Combined	0.3913043	0.3586957	0.3478261

Only the K-Nearest Neighbours predictor performs better than the 'no information rate', and only modestly so. Observe that the Support Vector Machine and Naive Bayes algorithms are very close to the no information rate. This is no accident - with the exception of Naive Bayes applied to the combined sentiment scores, these models are predicting *The Australian* for every article and so show its prevalence in the probe set.

Data visualisation of our two features reveals why our classification algorithms perform so poorly. There is barely even partial separation between classes of the sort that would enable a decision boundary to be established between them (Figure 1). The contrast is clear if we compare this plot Irizarry (2022) uses to illustrate multi-class classification, where for instance there is perfect separation in the proportion of different fatty acids in olive oil produced in different regions of Italy, enabling 100 % accuracy in classification.

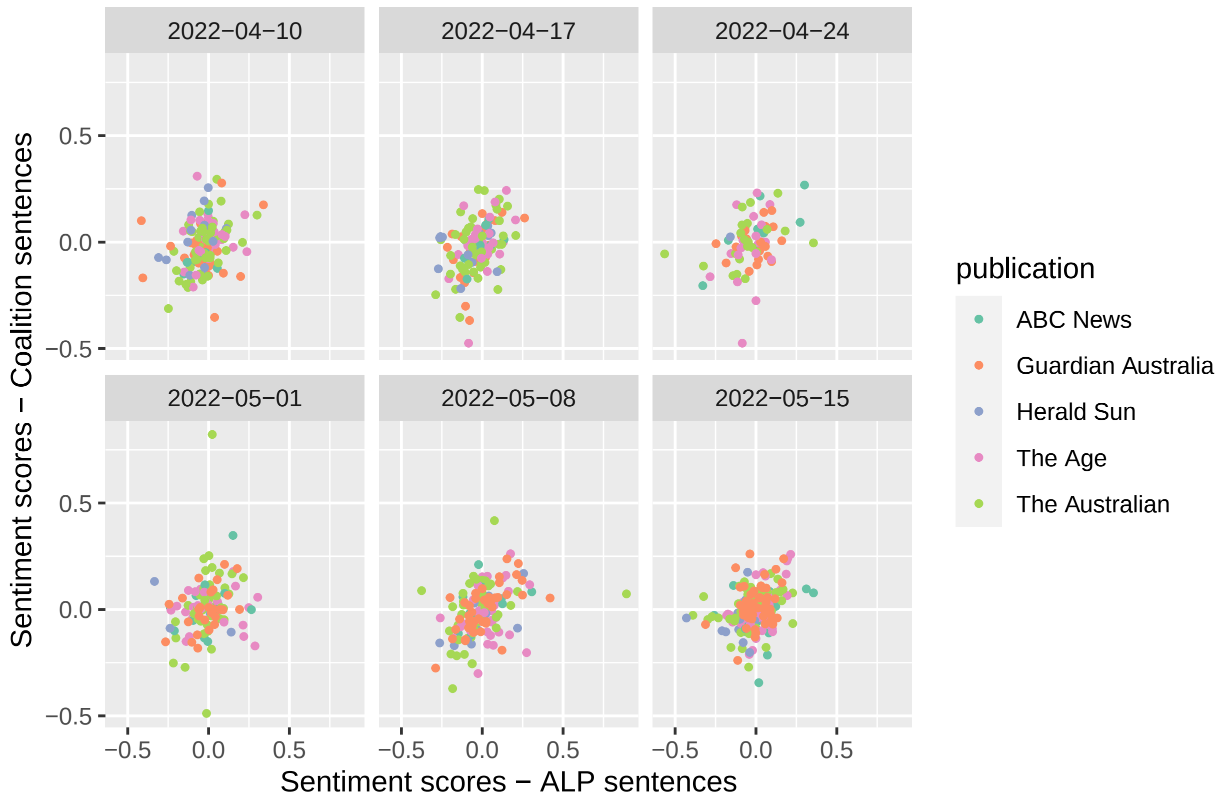
Figure 1: Distribution of Articles by Sentiment Scores



This distribution persists even when we divide our sentiment data into six date bins spanning the six weeks of the campaign, explaining why the addition of date-adjusted sentiment data does not add predictive power to our algorithms. These visualisations suggest that rather than further tuning of our algorithms or a search for more sophisticated classification algorithms, different or additional features are required to further investigate the feasibility of classifying the articles based on political bias (Figure 2).[16]

[16] As a further test of this conclusion, we trialled a polynomial SVM algorithm and a random forest algorithm on the raw sentiment scores. As results were not superior to the algorithms presented here, we did not attempt refinement of these algorithms. Code is presented in the accompanying R script.

Figure 2: Distribution of Articles by Sentiment Scores, by Week of Campaign



Adding headline sentiment to classification

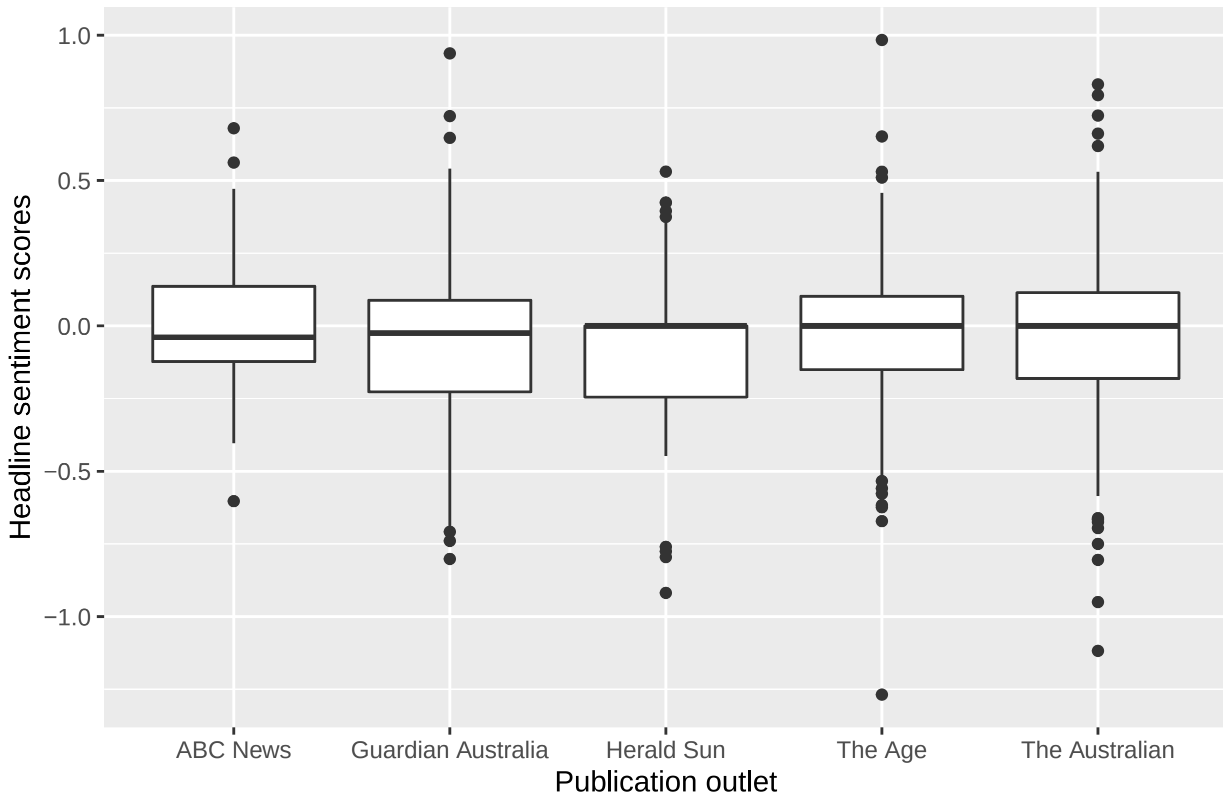
We next add the sentiment score for headlines as an additional predictor, and rerun our algorithms. As the raw sentiment scores perform the best of our three sentence sentiment measures, we combine the headings as a third predictor with our raw sentence sentiment scores. The addition of the headline sentiments does not improve model performance for any of our algorithms.

Table 5: Prediction results using headline sentiments

Features	K-Nearest Neighbours	Support Vector Machine	Naive Bayes
Raw sentiment scores	0.4130435	0.3586957	0.3586957
Date-adjusted sentiment scores	0.3586957	0.3586957	0.3586957
Combined	0.3913043	0.3586957	0.3478261
Headlines	**0.3586957**	**0.3586957**	**0.3586957**

Similar to the previous sub-section, a boxplot suggests why the headings are not adding predictive power. Although there are some outliers, we again see substantial overlap in scores between different publications (Fig-

Figure 3: Distribution of Articles by Headline Sentiment Scores



ure 3).

Sentence text classification

Our final - and least direct - measure of political bias is to perform full text classification on the sentences that we have identified as discussing the ALP and the Coalition. We produce three different datasets by collapsing all of the political sentences from each document into a single text:

i) ALP corpus - only the sentences coded as mentioning the ALP are included
ii) Coalition corpus - only the sentences coded as mentioning the Coalition are included
iii) Political sentences corpus - all sentences coded either as mentioning the ALP and the Coalition are included.

For (i) and (ii) respectively, we are forced to remove 37 documents from our dataset which do not contain any sentences coded as mentioning the ALP, and 17 documents with no sentences mentioning the Coalition. We again use the Support Vector Machine and Naive Bayes classifiers implemented through quanteda.textmodels, and achieve accuracy well above our sentiment-based classification, although somewhat below the full articles text scores.

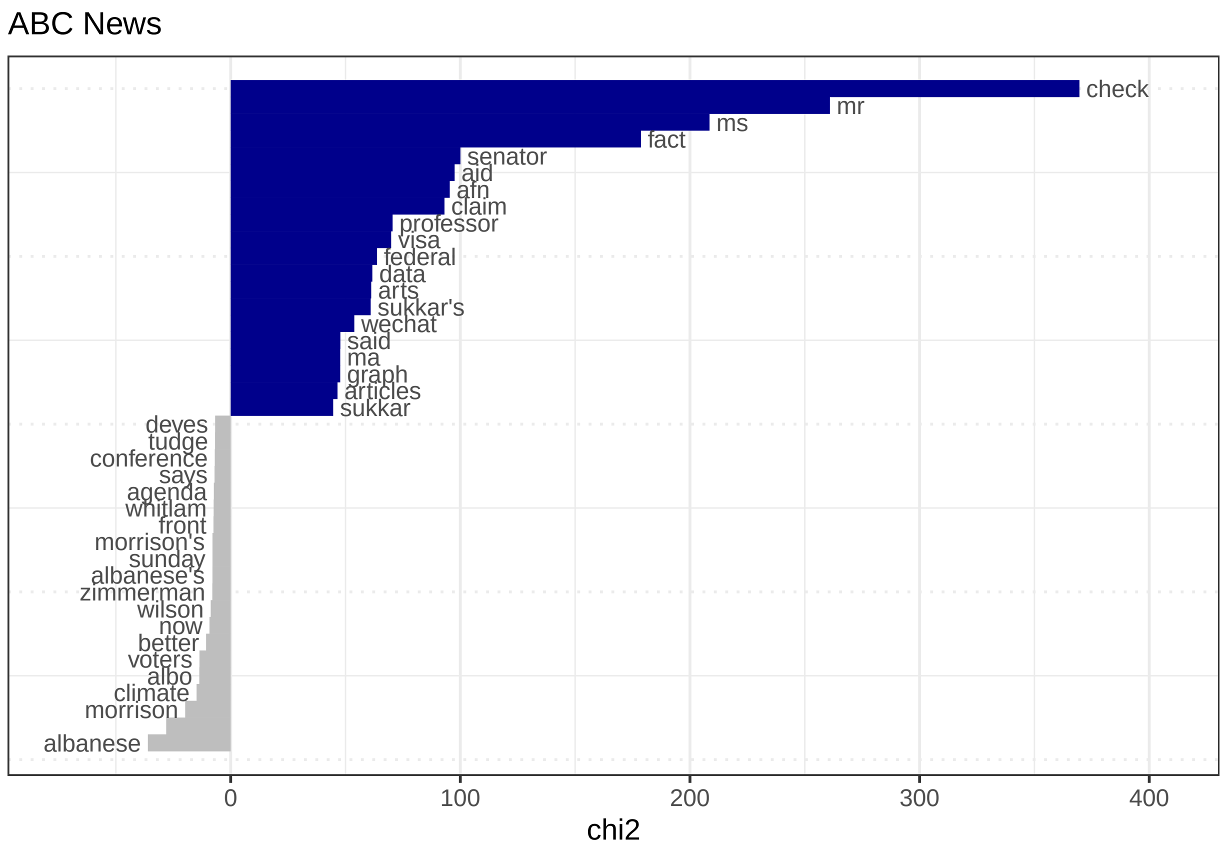
Table 6: Prediction results using text of political sentences

Features	Support Vector Machine	Naive Bayes	Naive Bayes Adjusted for Prevalence
ALP sentences	0.4456522	0.4565217	0.4782609
Coalition sentences	0.4565217	0.5108696	0.5108696
ALP and Coalition Sentences	0.5543478	0.5434783	0.5326087

Final model

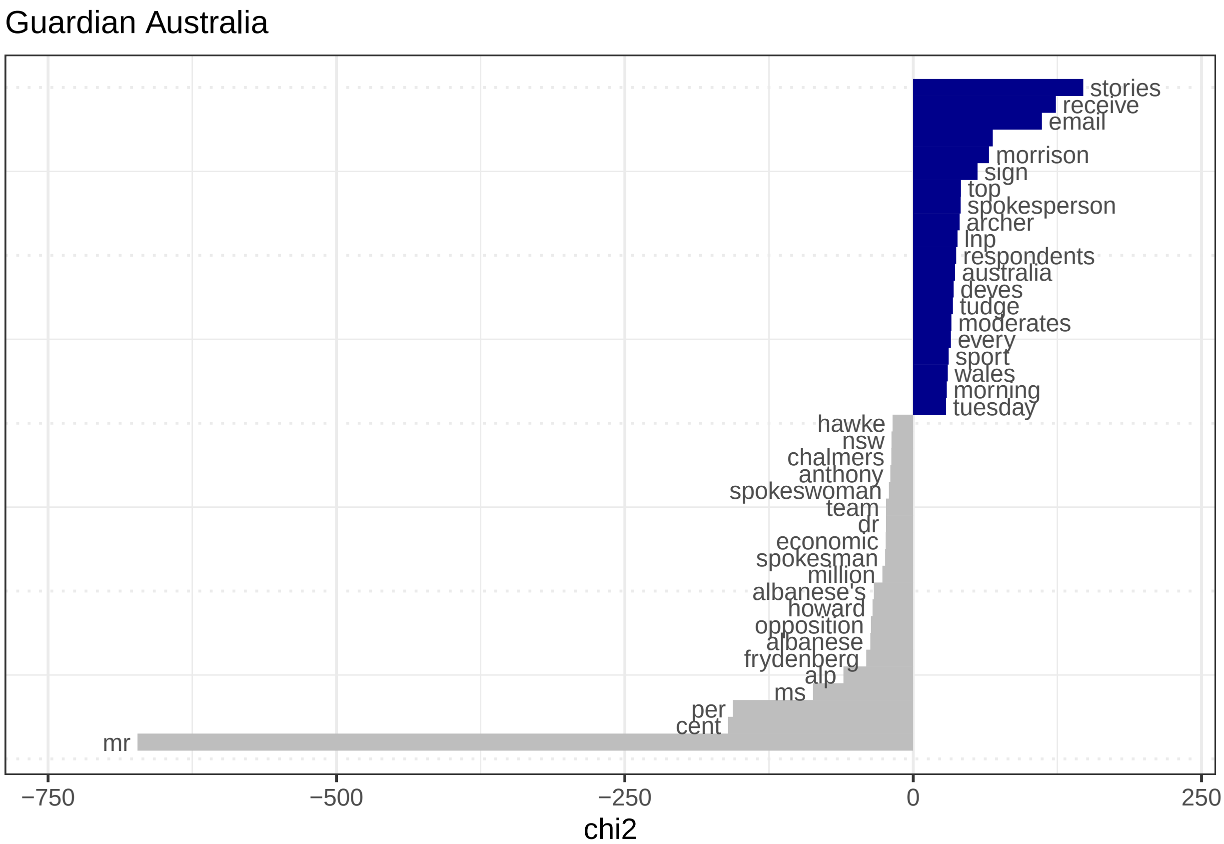
Of the various models we have tested on the probe set, the best performed are our three algorithms trained on the text of sentences mentioning the ALP and/or the Coalition. This presents us with the option of using an ensemble of the three algorithms, or simply selecting the Support Vector Machine as the best performing individual algorithm. Inspection of the actual predictions of Naive Bayes and the adjusted Naive Bayes algorithm makes it clear that an ensemble is inappropriate. The predictions of these two algorithms differ in only one instance, meaning they would outvote the SVM algorithm despite having inferior accuracy.[17] We hence train the SVM algorithm on the full train set, and test it on our test set. Our final classification accuracy is 66.7%.

In lieu of a formal measure of feature importance, we use quanteda.textplot's textplot_keyness function to visualise which words tend to occur more in each publications compared to the other four.[18] In the plots below, the blue columns at the top of each plot indicate words that are more likely to appear in the designated publication, whereas the grey columns are words that are more likely to appear in the others. Using this proxy measure, we see little evidence that these words are direct measures of bias. Many are entities (names) or nouns.

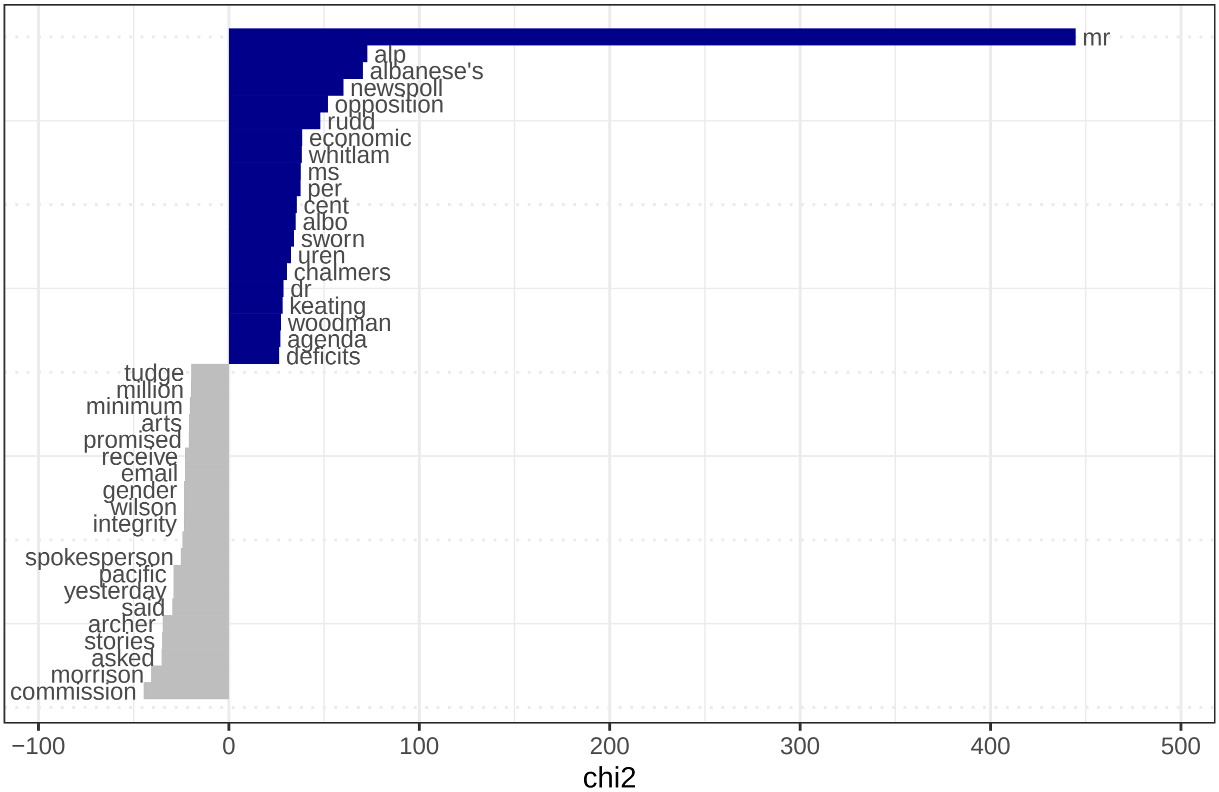


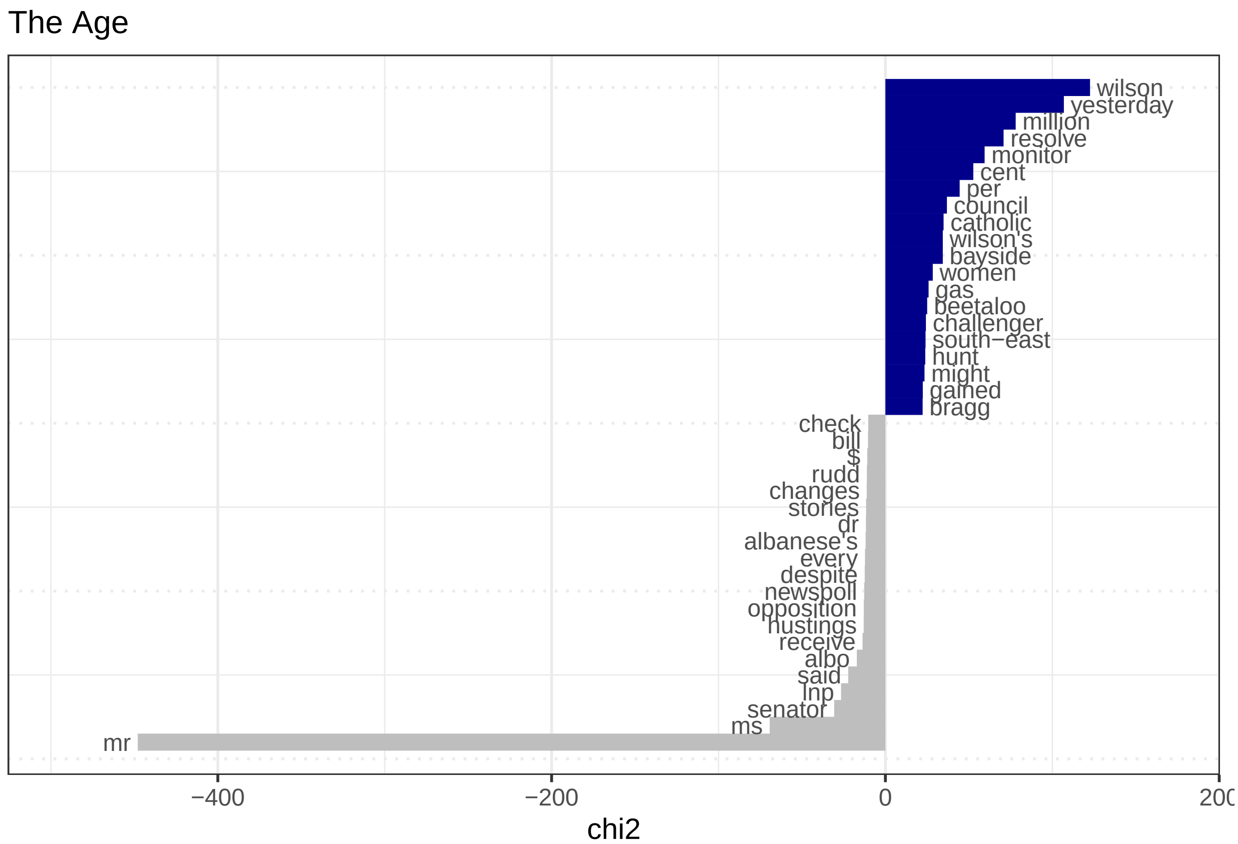
[17]Predictions are not listed here, but code to produce them is provided in the accompanying R script

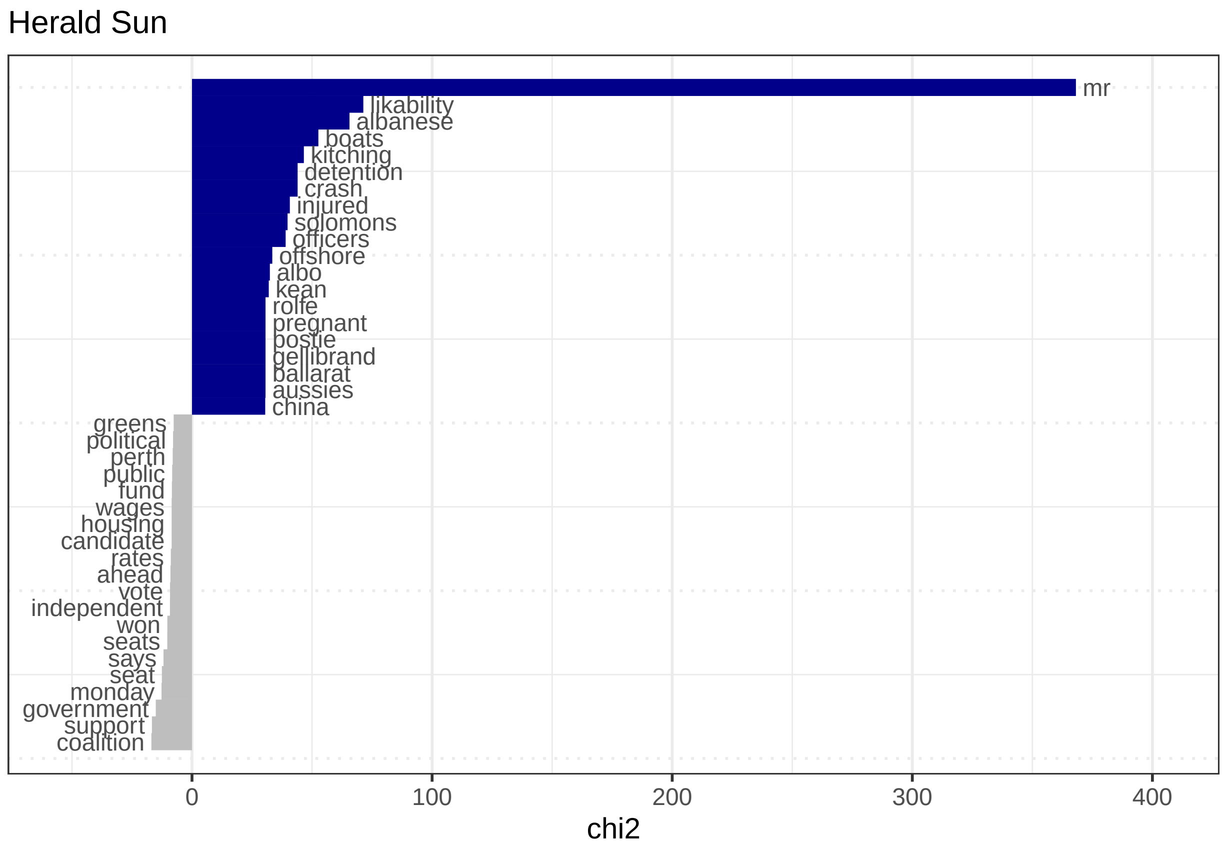
[18]https://tutorials.quanteda.io/statistical-analysis/keyness/


Guardian Australia
stories
receive
email
morrison
sign
top
spokesperson
archer
lnp
respondents
australia
deyes
tudge
moderates
every
sport
wales
morning
tuesday
hawke
nsw
chalmers
anthony
spokeswoman
team
dr
economic
spokesman
million
albanese's
howard
opposition
albanese
frydenberg
alp
ms
per
cent
mr
−750
−500
−250
0
250
chi2

The Australian




The Age
wilson
yesterday
million
resolve
monitor
cent
per
council
catholic
wilson's
bayside
women
gas
beetaloo
challenger
south−east
hunt
might
gained
bragg
check
bill
$
rudd
changes
stories
dr
albanese's
every
despite
newspoll
opposition
hustings
receive
albo
said
lnp
senator
ms
mr
−400
−200
0
200
chi2



Conclusion

These results leave us with an ambiguous conclusion. It is clearly possible to use machine learning techniques on the text of articles discussing the federal election to identify their publication outlet. The Support Vector Machine algorithm performs very well in classifying articles - far above the 'no information rate' - even based just on the sentences we identify as mentioning the ALP and/or Coalition. Without the ability using the quanteda.textmodels package to extract the most important features, however, we remain unsure whether this classification reflects differing biases between publications or some other systematic difference in these outlets' use of language. Identifying or developing a method to extract these features from quanteda.textmodels, or re-implementing the algorithms using a different package that would enable feature extraction, is one area for further work.

Equally, the modest performance of our sentiment-based measures does not support a definitive conclusion regarding bias. It may be that perceptions of distinctive political bias on the part of these media outlets are overstated. Equally, though, further work on developing better features may yield clearer results. This might include using other packages or approaches to generate sentiment scores, or developing a different computational measure of the political stance of each text. More likely, in the author's estimation, some combination of manual coding and computational measures may be needed to generate better features to enable classification based on political bias.

References

Gans, Joshua S. and Leigh, Andrew. 2012. 'How Partisan is the Press? Multiple Measures of Media Slant'. *The Economic Record*, 88(280): 127–147. http://andrewleigh.org/pdf/MediaSlant.pdf

Irizarry, Rafael A. 2022. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R* https://rafalab.github.io/dsbook/

Kuhn, Max and Johnson, Kjell. 2016. *Applied Predictive Modelling.* Springer: New York. https://link.springer.com/book/10.1007/978-1-4614-6849-3

Naldi, Maurizio. 2019. *A review of sentiment computation methods with R packages.* arXiv eprint. https://doi.org/10.48550/arXiv.1901.08319