Classification of News Article Political Leaning Using Various Machine Learning Algorithms

Nawal Valliani NVALLIA1@JH.EDU
Whiting School of Engineering
Johns Hopkins University

Abstract

Commonly used machine learning algorithms are used to create classification models to predict the political leaning of news articles sourced from a variety of publishers across the bias and reliability spectrum, as defined by AdFontes Media. Examples of machine learning algorithms employed herein include Decision Trees, Logistic Regression, and Support Vector Machines. The news article data was sourced by scraping the websites of select publishers over a period of two weeks. The data collection effort culminated in 1,494 unique news articles. Three approaches for making predictions were investigated - (1) using only data such as headline length, article content length, predicted news topic, and polarity, (2) using only the headline text, and (3) using only the article content text. All text data is vectorized using TF-IDF vectorization. Results show that the Random Forest classification is the highest performing of all of the tested models in each of the three classification approaches. The results also show that the highest quality predictions may be achieved by using the article content text to make predictions of political leaning.

1. Introduction

Curating desired content on the internet can be done in various ways. For example, users on Reddit can subscribe to subreddits relevant to their interests and see posts they would like to see. Similarly, users of StumbleUpon could select topics of interest and view random webpages relevant to those interests. This curation of content has been done in the realm of news articles as well, through Reddit or StumbleUpon, but also through Really Simple Syndication, or RSS, feeds. Users can subscribe to the RSS feeds of different news publishers and have content delivered to them.

Keeping the idea of a curated news feed in mind, is there a way a user can have content delivered to them without having to explicitly specify the type of news content or publisher desired? For example, if a user desires only left leaning political news delivered from any publisher, can a news aggregator differentiate between a left or right leaning news article without taking into account the political bias of the publisher? In other words, is it possible to classify a news article as left or right leaning solely based off of the headline or the content of the news article? The goal of this project is to develop classification models to reliably and accurately predict if a given news article is left or right leaning. This is a binary classification task.

2. Relevant Literature

A fair amount of research and testing has been done on the topic of text classification. Aggarwal and Zhai (2012) have noted that some of the most commonly used and successful

models for text classification are decision trees, pattern and rule based classifiers, Support Vector Machine (SVM) classifiers, and neural network classifiers. Ikonomakis et al. (2005) provide a framework for performing classification on text documents. The framework consists of the following steps in order: tokenization/normalization, stemming/lemmatization, stop word removal, vectorization of text, feature selection, and process through the learning algorithm. Zhou, Resnick, and Mei (2011) studied the classification of the political leaning of news articles. Zhou, Resnick, and Mei used semi-supervised learning methods such as Random Walk and local consistency-global consistency. In cross-validation, their best algorithm achieved 96.3% accuracy on the articles and user comments sourced from the social media website Digg.

3. Hypothesis and Approach

Hypothesis

It is hypothesized that left vs. right leaning articles may be separated using a machine learning classifier with high accuracy and precision. That is, a binary classification task of classifying news data into left or right leanings will be quite successful. In personal experience, the presence of certain words and phrases between left-leaning and right-leaning news media allow insight into the true political nature of the article in question. Right-leaning articles tend to have more negative words and resort to harsh criticisms and name-calling of public figures, while left-leaning news articles tend to avoid the negative verbiage and instead employ more neutral or positive terminology.

Data Acquisition Process

An extensive search for the data required for this project was unsuccessful, for the most part. While there are data sets with publisher data and news article headline/content data, many of the news articles are not directly politically-oriented and thus would add noise into the classification process. For example, a data set titled "All The News" was found published on Kaggle which contained roughly 143,000 news articles from 15 publishers across the political spectrum (left-leaning and right-leaning publishers, as well as those in the "middle" such as Reuters and Associated Press). The data was simply acquired from the front page of many of these publishers, resulting in the data being across a variety of news topics, such as finance, technology, entertainment, etc. Data sets such as "All The News" were ultimately not used in the project herein, as the effort required to scrub the data of non-politically oriented news would be tremendous.

In order to have data specifically suited for this project, it was opted to scrape data from select news sources across the political spectrum. The "spectrum" is defined by the AdFontes Media Bias Chart. Teams of analysts score popular news publishers in terms of bias and reliability. The bias scale spans the range -42 to +42, where more negative scores indicate extremely left-leaning publishers and more positive scores indicate extremely right-leaning publishers. The range -6 to +6 defines publishers in the middle or with balanced-bias. The reliability scale spans the range 0 to 64, where higher scores indicate the publisher is more reliable. Each document is classified as "Left," "Middle," or "Right" depending on

the bias score of the corresponding publisher. Ideally, each document would be classified at the document level; but the only available metrics are at the publisher level.

The scraping effort was rather involved. Nine news sources were selected (shown below) across the bias and reliability spectrum. For each source, the HTML source for the "Politics" front page (or, in some cases, the page closest to politics - e.g., "World News" instead of "Politics") was examined to determine where and how headlines and the paragraphs which constitute the article content are located and formatted (e.g., within div tags, paragraph tags, header tags, etc.). The appropriate HTML tags were utilized in a Python program leveraging the Requests and BeautifulSoup libraries to acquire the text data. The Python script created to scrape the selected news sources was executed several times a day over a period of two weeks. The scraping effort resulted in a total of 1,494 unique news articles acquired over a range of bias/reliability and nine publishers.

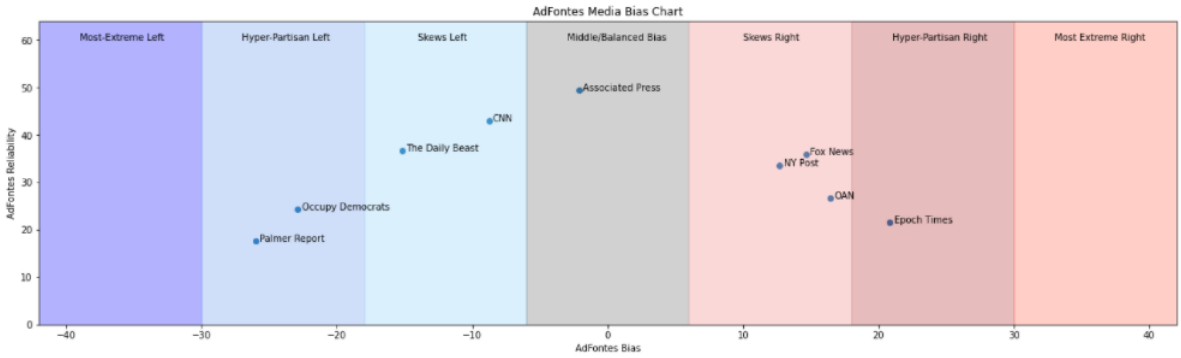

Figure 1: Bias and Reliability Chart for Utilized Sources

Data Handling

The scraped news data are saved and imported using the Pandas data analysis library within Python. Prior to building and tuning the classification models, the required data are stored as NumPy arrays to aid in computational complexity, as repeatedly querying/fetching data from a DataFrame can get expensive from a time complexity standpoint. Any categorical and ordinal variables are OneHot and ordinally encoded, respectively, as needed. Any null data points are dropped from the data prior to building classifiers.

Experimental Methods

Prior to attempting any classification tasks, an extensive Exploratory Data Analysis (EDA) was performed on the data which examined the data in several ways. As mentioned previously, the classification process followed herein is similar to that outlined by Ikonomakis et. al (2005). The data is first normalized (tokenization, stemming/lemmatization, stop word removal), then vectorized using TF-IDF vectorization. Feature selection and transformation may then be performed, if desired. Lastly, the learning algorithm is fed the training data. These steps will be discussed in detail in the following sections.

4. Distribution of Acquired Data and Exploratory Data Analysis

The distribution of the data may be viewed in several ways. First, the distribution of the number of articles per publisher is examined:

Table 1: Number of Articles by Publisher

Publisher	**# of Articles**	**Percentage**
NY Post	338	22.62
CNN	281	18.81
Fox News	225	15.06
OAN	196	13.12
Associated Press	167	11.18
Palmer Report	152	10.17
The Daily Beast	56	3.75
Occupy Democrats	40	2.68
Epoch Times	39	2.61
Total	**1494**	**100.0**

The differences in the number of news article per publisher may be attributed to several factors:

- CNN, Fox News, and NY Post were among the first news sources to be scraped
- The number of news articles available on a page to be scraped (varies per publisher)
- Some publisher's pages were more prone to having duplicates URLs present/scraped. Duplicates are removed.

The distribution of the political leaning is also examined:

Table 2: Number of Articles By Leaning - Three Classes

Leaning	**# of Articles**	**Percentage**
Right	798	53.41
Left	529	35.41
Middle	167	11.18
Total	**1494**	**100.0**

A significant portion of the data is right-leaning when the above distribution is examined. Ultimately, however, the classification task explored herein is a binary classification. That is, the "Middle" category will be absorbed into the left-leaning articles. The only publisher categorized as "Middle" is the Associated Press, which, speaking strictly according to the bias scores, is slightly left-leaning and is treated as such within machine learning models. The distribution by leaning is adjusted to reflect this:

Table 3: Number of Articles By Leaning - Two Classes

Leaning	**# of Articles**	**Percentage**
Right	798	53.41
Left	696	46.59
Total	**1494**	**100.0**

Headline and Article Content Length

The length of each headline and article is determined by tokenizing the text data and recording the length of the tokenized data. For this, no stop word removal or lemmatization is performed. The average headline length is about 11.32 words per headline, whereas the average content length is approximately 598 words per article. The distribution of these lengths is analyzed through histograms:

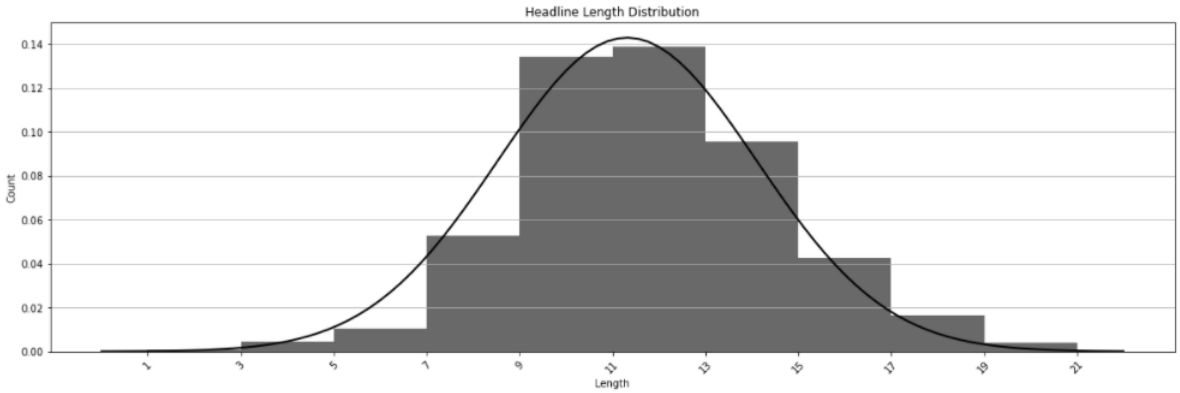

Figure 2: Headline Length Distribution

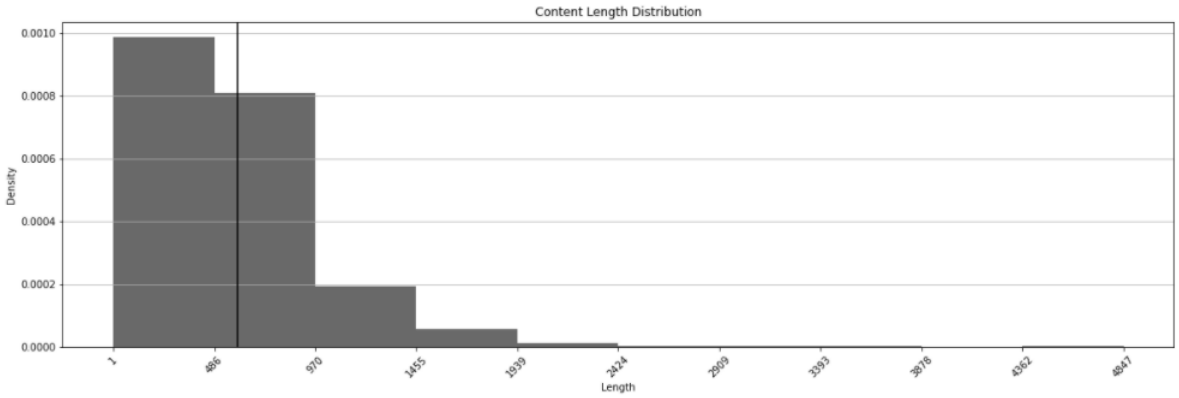

Figure 3: Content Length Distribution

The headline length distribution appears to follow a distribution similar to that of the Gaussian/Normal distribution, whereas the content length distribution appears to represent a Gamma distribution. The mean content length is skewed by the presence of a small amount of long articles, since it appears that the bulk of the data has less words per article than average.

Data Normalization

The EDA tasks in the following sections require the data to be normalized. The data is normalized in several ways. All punctuation is removed, and any uppercase letters present are case-folded. The English stop word list present in the Natural Language Toolkit (NLTK) Python library is used to perform stop word removal. Lemmatization is performed using the WordNet lemmatizer available in NLTK. The effects of normalization on the headline and article content data are presented in the below table, as well as the percent change in the total number of unique terms after each normalization. As expected, the number of unique terms in both the headlines and the content significantly decreases after the normalization step.

Table 4: Corpus Terms Before and After Normalization

Feature	**Unique Terms**	**% Change**
Headline	5944	-
Normalized Headline	4120	-30.69
Content	69537	-
Normalized Content	29365	-57.77

Topic Modeling

Two common approaches to topic modeling are explored - Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF). Using both approaches, 10 topics are derived from the data and each document is tagged with two topics, one topic derived from LDA and one from NMF.

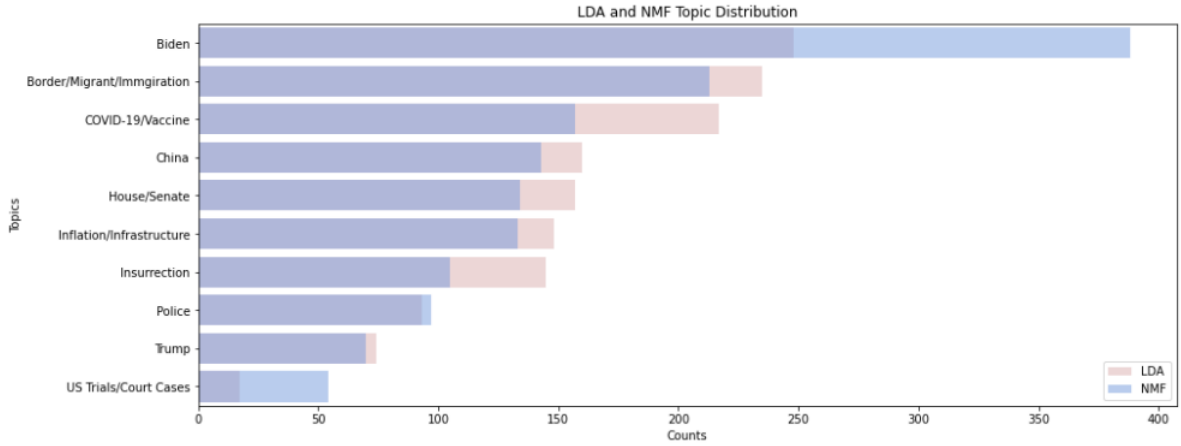

Figure 4: LDA and NMF Topic Distribution

There is considerable overlap between the topics derived from each approach. For either approach, a significant number of news articles are relevant to the Biden Administration and ongoing issues at country borders. At the bottom of the list for both LDA and NMF are articles relevant to Trump/the Trump Administration and on going trials and court cases in the United States.

Sentiment Analysis - Polarity

Sentiment analysis is performed on each headline and article to determine it's polarity. The polarity score spans the range -1.0 to 1.0, where negative polarity indicates the presence of more words with negative connotations and positive polarity indicates the opposite. Polarities of and around zero indicate a neutral sentiment. The headline and content polarities are visualized with histograms with the different leanings shown:

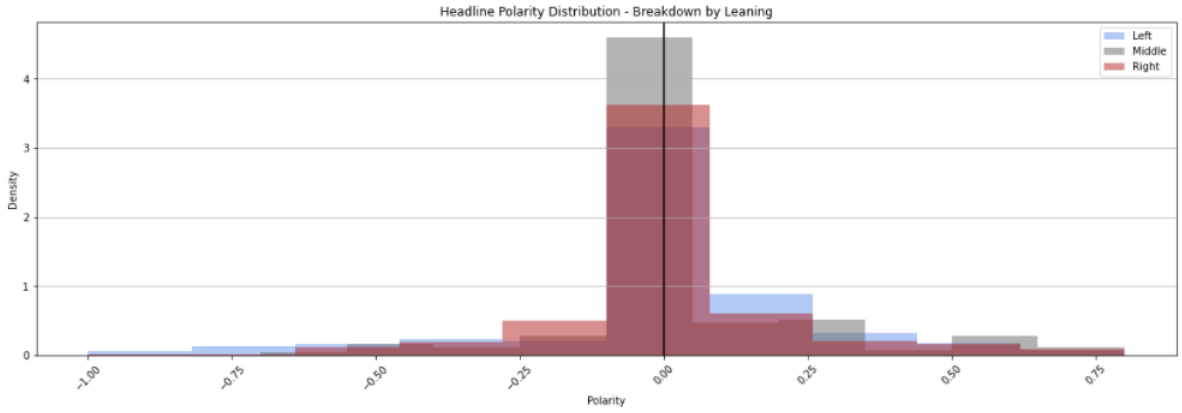

Figure 5: Headline Polarity Distribution

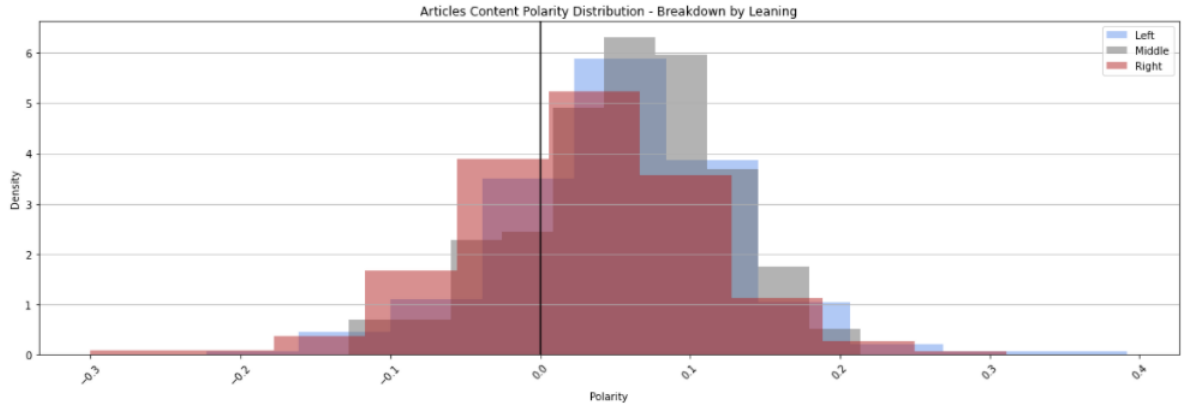

Figure 6: Content Polarity Distribution

The solid vertical bar indicates a polarity of zero. While a significant portion of left and right articles have neutral, or near-neutral polarity in headlines, news articles in the middle are more abundant in neutral headlines. Left-leaning headlines appear to have more density in the negative extreme of polarity.

The content polarity distribution is more telling. Left-leaning articles and articles in the middle tend to be more positive, while right-leaning articles tend to be more negative. This trend is seen on both sides of the neutral point and is more evident in the negative

polarity. The density of right-leaning articles is higher than the left-leaning and middle articles, indicating that the right-leaning articles, as a whole, tend to have more negative verbiage than left-leaning or middle articles.

TF-IDF Vectorization

The headline and article content text are vectorized as TF-IDF vectors using the available scikit-learn functions. The inputs to the machine learning models are these TF-IDF vectors.

5. Classification Models and Results

A variety of algorithms/models are explored herein:

- k-Nearest Neighbors
- Logistic Regression
- Support Vector Machines
  - RBF Kernel
  - Linear Kernel
  - Sigmoid Kernel
  - Polynomial Kernels (3rd and 5th degree polynomials)
- Decision Trees
- Random Forest

The classification performance of all machine learning models is examined through 5-Fold Cross Validation. 5-Fold Cross Validation is performed to ensure that each data point in the total dataset is present at least once in a training set. This aids in producing a more robust metric for the classification or regression metric at hand. The data set is split into five groups (folds). Each fold is used once as a test set, with the remaining folds being combined to use as a training set. As such, only 20% of the data is being tested upon each time.

Baseline Model - Majority Vote Classifier

All machine learning models explored herein are compared to the Majority Vote classifier. In this simple classification model, each test data point is assigned the majority class label from the training data. The machine learning models are compared to this simple classifier to ensure that the models will perform better than simply picking a majority class and applying it to the test data. If machine learning models perform worse than the majority vote classifier, there is essentially no use in developing such models for predictions. The classification metrics for the majority vote classifier are summarized below:

Table 5: Majority Vote Classifier Metrics

Metric	Value
Avg. Accuracy	0.5338
Avg. Precision	0.5338
Avg. Recall	1.0000
Avg. F1	0.6960

Basic Classification on Headline/Content Metrics and LDA/NMF Topics

First, classification is performed using the headline/content length, polarity, and topics generated from LDA and NMF with a small subset of machine learning models. The goal of developing a model using these features is to determine if headline and article content would aid or harm classification performance and to see if a quick classification can be done using the available data (i.e., without vectorizing a large amount of text).

Table 6: Headline/Content Metrics and LDA/NMF Topics Only - Classification Metrics

	NMF	LDA
Model	**Precision**	
Majority Vote	0.5338	0.5338
k-Nearest Neighbors	0.5773	0.5773
Decision Tree	0.6745	0.7623
Random Forest	0.7314	0.7676

K-Nearest Neighbors slightly outperforms the majority vote classifier in precision, but it is the least favorable of the machine learning models explored. Decision Tree and Random Forest classification perform much better than K-Nearest Neighbors. Additionally, it appears that the topics generated using LDA result in better classification performance than the topics generated using NMF. Random Forest classification grows many different Decision Trees, which are individually used to create multiple predictions, and the majority prediction is then selected. This approach works very well because Decision Trees are able to determine which features of the data are most pertinent and indicative of the target class (using metrics such as Information Gain), and perform a kind of feature selection to create their predictions. Below is an example of a Decision Tree that was grown for this classification task. Note that Decision Tree hyperparameters (such as maximum depth) were iterated to determine the best configuration to grow the tree, which is why the following tree is shorter than most fully-grown Decision Trees with a large depth.

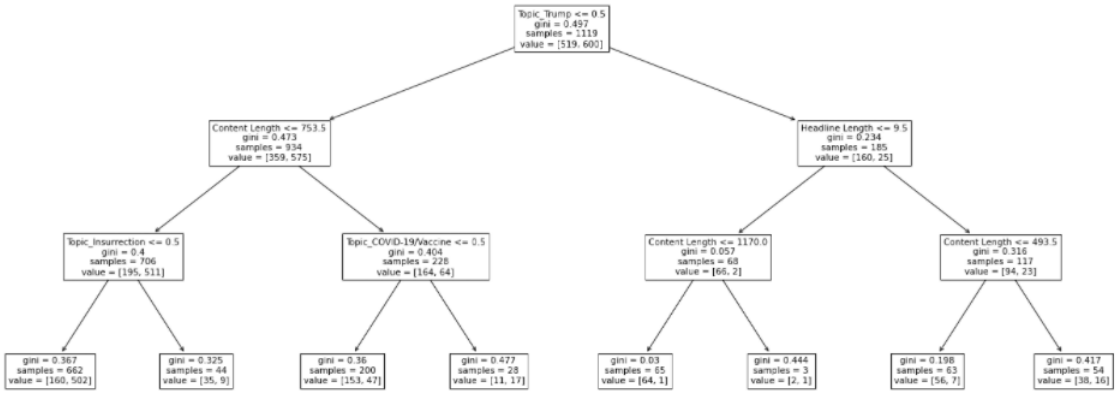

Figure 7: Example Decision Tree

Classification on Documents as TF-IDF Vectors

Headline Only

Classification models using only the text from each document's headline are created. It is expected that classification using only the headline data will outperform the basic/quick classification discussed above, but underperform classification using the article content.

Table 7: Headline Only - Classification Metrics

Model	Avg. Accuracy	Avg. Precision	Avg. Recall	Avg. F1
Majority Vote	0.5338	0.5338	1.0000	0.6960
K-Nearest Neighbors	0.9250	0.9488	0.9120	0.9284
Logistic Regression	0.8882	0.8717	0.9378	0.9002
SVM - RBF	0.9350	0.9246	0.9609	0.9408
SVM - Linear	0.9143	0.9144	0.9301	0.9205
SVM - Sigmoid	0.8748	0.8654	0.9128	0.8861
SVM - Polynomial (3)	0.8968	0.8937	0.9335	0.9049
SVM - Polynomial (5)	0.8854	0.8771	0.9354	0.8949
Decision Tree	0.9357	0.9446	0.9369	0.9395
Random Forest	**0.9404**	**0.9467**	**0.9446**	**0.9443**

The Random Forest classifier outperforms all other classifiers, based on the average F1 score; however, the SVM with a Radial Basis Function kernel comes in a close second. Seeing as how the results presented in Table 7 are vastly higher than those in Table 6, utilizing the headline data to make classifications enhances the classification ability and outperforms the basic classification method discussed in the previous section.

Article Content Only

Classification models using only the article content from each document are created. These models are expected to perform the best out of all of the outlined approaches; however, are also expected to be the slowest, since article content can get lengthy and creating vectors of large documents may become expensive in terms of the time cost.

Table 8: Article Content Only - Classification Metrics

Model	Avg. Accuracy	Avg. Precision	Avg. Recall	Avg. F1
Majority Vote	0.5338	0.5338	1.0000	0.6960
K-Nearest Neighbors	0.9410	0.9571	0.9332	0.9442
Logistic Regression	0.9290	0.9285	0.9442	0.9344
SVM - RBF	0.9612	0.9613	0.9685	0.9640
SVM - Linear	0.9672	0.9729	0.9672	0.9694
SVM - Sigmoid	0.9544	0.9638	0.9524	0.9573
SVM - Polynomial (3)	0.9256	0.9238	0.9491	0.9320
SVM - Polynomial (5)	0.8962	0.8893	0.9384	0.9049
Decision Tree	0.9692	0.9725	0.9700	0.9709
Random Forest	**0.9719**	**0.9758**	**0.9735**	**0.9739**

Once more, the Random Forest classifier outperforms all other classifiers based on the average F1 score. The Decision Tree classification is in a close second, unlike seen previously with the headline only predictions where SVM/RBF outperformed Decision Tree classification. Comparing the scores above to those in Table 7 (headline only classification), the article content helps in creating more reliable and accurate predictions.

These results are to be expected. Headlines are written to capture the reader's attention and get them to engage with the link and read the full news article. While the headlines may provide some insight into the topic and the ensuing discussion, that is not always the case. Headlines must be short and attention-grabbing, so they will almost always lack the appropriate context when compared to the article content. As such, the article content provides much more context and a large volume of information to infer the political leaning of the article. That said, it is still impressive that the classification of news articles performs with such high metrics using only the headline data.

6. Conclusions

The results of the experiments conducted herein demonstrate that accurate and reliable classification models can be created to classify news article political leaning. The experiment results also confirm Aggarwal and Zhai's (2012) claims. Decision Trees and SVMs, which are pattern/rule based classifiers, consistently provided high classification metrics and performance. Aggarwal and Zhai's claims also apply to Random Forest classification, which is simply many random Decision Trees, as discussed previously. Since neural network

classification was not explored herein, a suggestion for future work would be to include neural network classification to gauge the performance.

The framework described by Ikonomakis et al. (2005) proved to be a suitable set of guidelines to follow to build classification models for news article texts. Ikonomakis et al. (2005) discussed various ways of vectorizing the documents to be classified. The vectorization herein was on a word basis. A suggestion for future work would be to observe the impact of vectors consisting of character n-grams on the classification performance.

Zhou, Resnick, and Mei (2011) stated that their best model for classifying news article political leaning achieved 96.3% accuracy. Of the three different classification approaches explored herein, the article content only classification achieved accuracy above 96.3% (see Table 8). The Random Forest classification achieved 97.19%, slightly outperforming Zhou, Resnick, and Mei's best semi-supervised learning model.

References

C. C. Aggarwal and C. X. Zhai. *Mining Text Data, Chapter 6 - A Survey of Text Classification Algorithms*, 2012.

D. X. Zhou, P. Resnick, Q. Mei. Classifying the Political Leaning of News Articles and Users from User Votes, *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media)*, 2011.

M. Ikonomakis, S. Kotsiantis, V. Tampakas. Text Classification Using Machine Learning Techniques, *WSEAS Transactions on Computers, (Vol. 4, Issue 8)*, 966-974, 2005.