

Creating Natural Language Processing Models for Detecting Fake News with Limited Data

Introduction

Fake news is information spread with the intent to manipulate and misinform. It has affected the trust in democratic processes and the mitigation of the COVID-19 pandemic. As such, it must be automatically moderated on the social media sites where it is spread.
Research Goal:
Create a successful machine learning model that can identify an article as fake or real solely based on the text or title of said article.

Materials & Methods

One training dataset of just titles, one testing dataset split into titles and articles.

Preprocessing:

- Lowercase and remove punctuation
- Tokenize text
- Remove common stop words
- Lemmatize text
- Visualize most common words and add any “useless” words back to stop words list

Training:

- Reprocess with new stop words list
- N-gram analyze - store every sentence as every set of consecutive words within
- Vectorize data
- TF-IDF transform - provide rare items with more importance than common items
- Save models locally

Testing:

- Load models
- Test on 20% of training data subset
- Test on titles from testing dataset
- Test on texts from testing dataset
- Save confusion matrices with results for each subset

Results

Every test output a binary confusion matrix that was processed in order to get the accuracy, negative predictive value, recall, and precision scores.

- The best performing algorithm overall was the passive aggressive classifier model.
- All models performed far better on the texts than the titles.

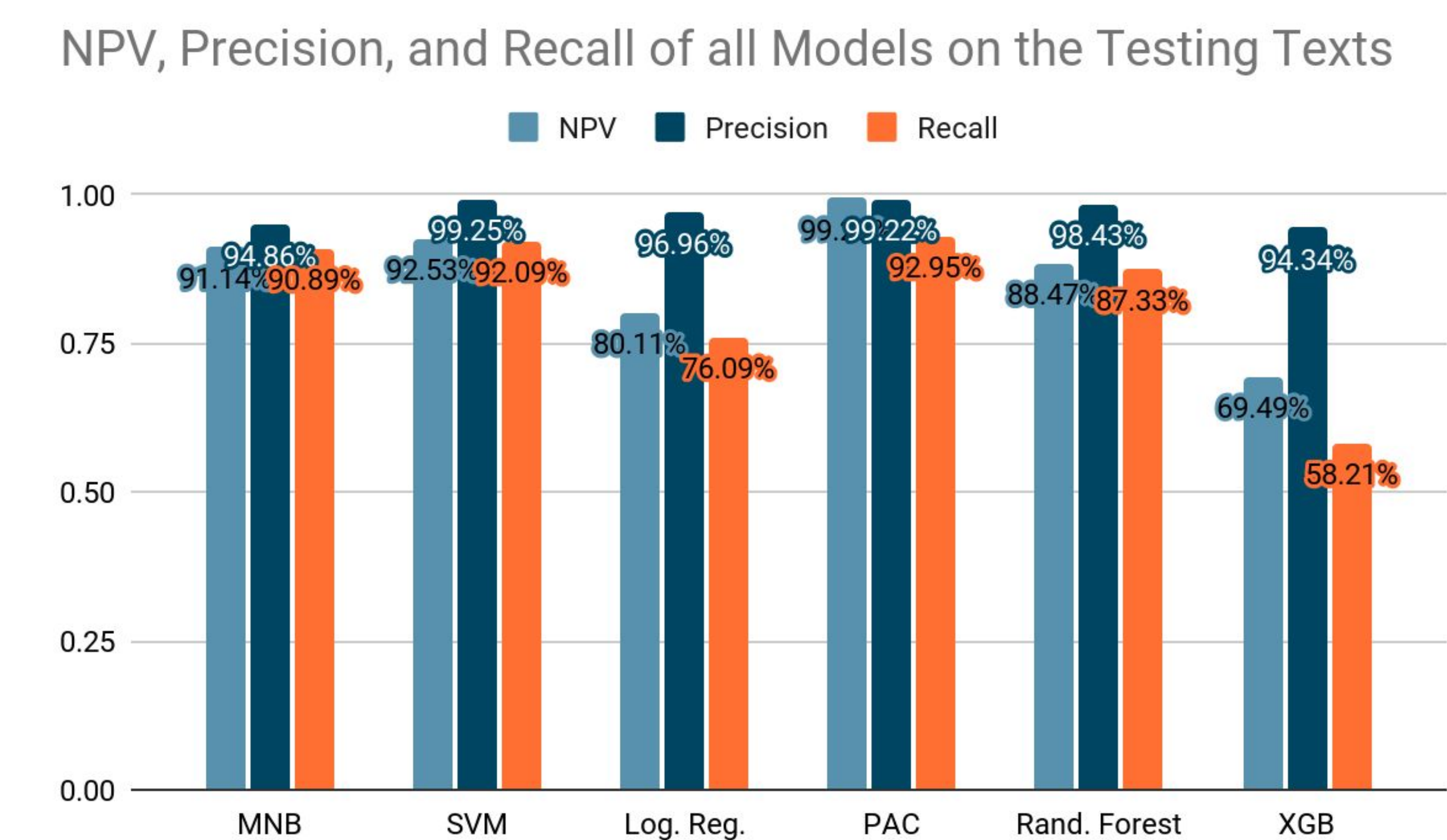
Performance on the training data subset is less important than the performance on the titles and texts.

Table 1

Test Subset	Algorithms					
	MNB	SVM	Log. Reg.	PAC	Rand. Forest	XGB
Training	.91305	.95578	.93412	.96028	.93186	.91670
Titles	.69997	.53245	.53117	.55195	.49760	.49888
Texts	.92925	.95668	.86765	.96084	.92919	.77206

The table shows the accuracy scores for the various models on the various testing sets.

Figure 1



Since SVM has only a marginally better precision, but a worse NPV compared to PAC, PAC can be considered the strongest algorithm.

Discussion & Conclusion

Strong NLP models can be trained with less data than that for predictions.

Deployment:

- Less time can be spent training the models, so they can be retrained more frequently as the subject of fake news articles changes.
- Requiring only the text of an article allows the models to be able to analyze as many articles as possible, since there is no other characteristics to be labeled manually.

Ethical Considerations:

- Raw text input allow for “fairer” classifications, since the models can’t make generalizations based on subject or party affiliation directly, and the text must act as proxy.
- Censorship associated with automated content moderation must be addressed, but with focus on understanding on how companies can manipulate models with biased training data.

Future Directions

Test the PAC model with data scraped directly from the Internet. Implement model optimizations such as grid-search and cross-validation. Experiment with feature extraction on the texts (eg. capital letter frequency, word/sentence length, etc.).

References

George, J. A. (2020, June 14). *Fake News Detection using NLP techniques* / by Joyce Annie George / Analytics Vidhya. Medium.
<https://medium.com/analytics-vidhya/fake-news-detection-using-nlp-techniques-c2dc4be05f99>

Jillani. (2022). *Fake or Real News*. Kaggle.
<https://www.kaggle.com/datasets/jillanisoftech/fake-or-real-news>

Kumari, R., Ashok, N., Ghosal, T., & Ekbal, A. (2022, January). What the fake? Probing misinformation detection standing on the shoulder of novelty and emotion. *Information Processing & Management*, 59(1), 102740. Science Direct.
<https://doi.org/10.1016/j.ipm.2021.102740>

Nagy, K. S., & Kapusta, J. (2023, May 25). Twldw—A Novel Method for Feature Extraction from Unstructured Texts. *Appl. Sci.*, 13(11), 6438. <https://doi.org/10.3390/app13116438>

Shetty, B., & Koss, H. (2022). *NLP Machine Learning: Build an NLP Classifier*. Built In. Retrieved July 18, 2023, from <https://builtin.com/machine-learning/nlp-machine-learning>

Refer to the paper for full list of references.