Fake News Detection Project Final Report

Abstract

This project combines three datasets (ISOT Fake News, Fake News Classification, and Horne 2017) to classify news articles as fake or true. We used data science techniques like TF-IDF for feature extraction and PCA (Principal Component Analysis) for dimensionality reduction, followed by visualizations to identify patterns in the data. After cleaning and merging the data, we focused on feature engineering using Word2Vec embeddings to capture the semantic meaning of words and one-hot encoding of the article's source. We then trained several models, i.e., Logistic Regression, XGBoost, Random Forest, Naive Bayes, SGD Classifier, Passive Aggressive Classifier, and Neural Network, on the combined dataset. The final objective is to predict the authenticity of news articles based on these features.We found that XGBoost and Neural Networks performed the best with an accuracy of around 95 percent on our custom fake news dataset.

Background

Fake News Definition and Significance:

Fake news refers to content that is intentionally false or misleading, designed to deceive or provoke controversy. It has become a growing problem, especially with the rise of AI tools that can generate fake content more easily and convincingly. Fake news undermines public trust in media, manipulates opinions, and can influence critical events like elections. Tackling this issue is essential for maintaining the integrity of information online.

Impact of Fake News Classification:

With the surge in AI-generated content, detecting fake news has become harder. Automated detection tools based on machine learning can help spot misleading information quickly, ensuring online content remains trustworthy and accurate. These tools are vital in stopping misinformation from spreading and improving the quality of information available to the public.

Existing Research:

Studies have shown that machine learning can effectively classify news articles as fake or real. Traditional models like Support Vector Machines (SVM) and Naive Bayes have had success, but newer approaches using deep learning, like LSTM and BERT, provide even better accuracy by understanding more complex patterns in text. However, challenges remain around dataset biases and ensuring the models work across different topics.

Our Approach:

In this project, we’re building on these methods by experimenting with different models using a custom dataset drawn from multiple fake news sources. Our dataset combines political and general news articles to improve the model’s accuracy and reduce biases from previous studies. The aim is to create a more reliable fake news classifier that can generalize across various types of content.

Dataset

The dataset used in the project is the final fake news dataset, which contains news articles and a corresponding label indicating whether each article is "fake" or "true". The dataset includes various features, such as the article text and the source of the article.

We have combined three different fake news datasets

1. ISOT Fake News detection dataset

   Topic: World and US political new

   Size: 23502 fake news instances, 21417 real news instance
2. Fake News Classification

   Topic: General news not strictly political

   Size: 37,106 fake news instances, 35,028 real news instances
3. Horne 2017 Fake News Data

   Topic: Random Data

   Size: Around 100 fake news and true news (but each text segment is quite large)

The final dataset has three columns

1. Text: the news segment
2. Source: which of the three original dataset this text is from
3. Label: fake or real

Method

The approach involves:

1. Data Collection: A final dataset combining various features, including the text of the articles and their labels (fake/true), is used.
2. Feature Extraction: The text data is converted into numerical form using *TF-IDF* (Term Frequency-Inverse Document Frequency) vectorization, which captures the significance of words in relation to the entire corpus.
3. Data Standardization: The feature vectors are standardized using *StandardScaler* to ensure that the data is on the same scale.
4. Dimensionality Reduction: *PCA* is applied to reduce the high-dimensional feature space to two components for visualization.
5. Visualization: A scatter plot is generated to visualize the separation of fake and true news based on the first two PCA components
6. Model Training and Testing: After our preprocessing steps we will train multiple ML classification models and compare results to see which performed the best.

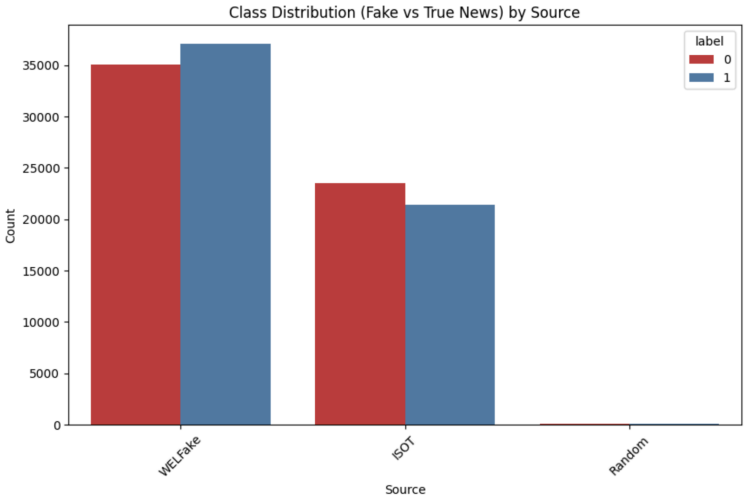
Analysis

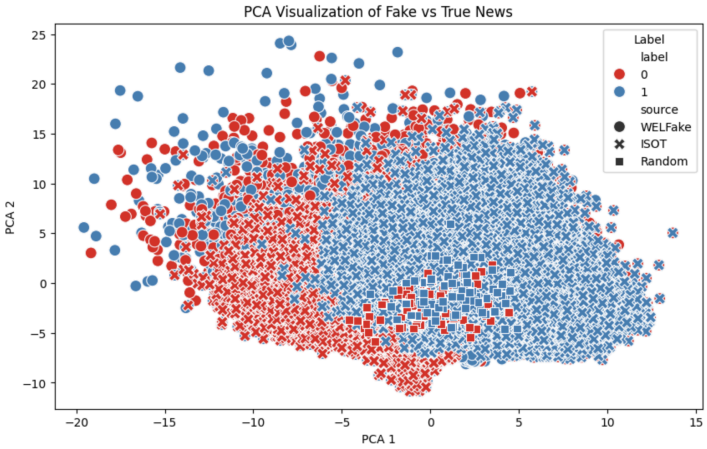
<u>Data Collection and Cleaning:</u>

Description: All 3 of our datasets (ISOT Fake News, Fake News Classification, Horne 2017) have been combined into one larger set. Only the top 100 words from Horne were used as each text segment in the set is very large. We added the ‘source’ column into the combined dataset. This inclusion was integral to the accuracy of our model, outlined later in the Training and Testing Phase.

Data Visualization

Description: The visualizations below were generated in order to determine next steps for building our model. The distribution of the dataset was very balanced, albeit with a lower amount of samples from the Horne dataset.

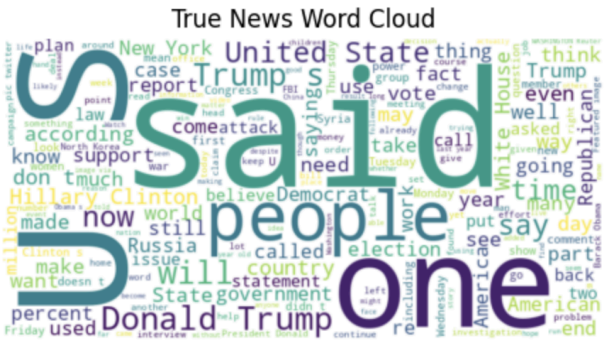




The PCA indicated that the data is not linearly separable and that Linear Regression wouldn’t be sufficient. This also may explain why Logistic Regression eventually performed worse than other models. There is overlap between the True and Fake News Word Clouds below, necessitating the use of Word2Vec to learn the semantic meaning of the words.

Feature Engineering

We converted every news article into a Word2Vec embedding trained on the full corpus to capture semantics beyond word counts. These embeddings were then concatenated with a one-hot encoding of each article’s source to create a unified feature set.



With this combined dataset, we ran several different classification models to see which would perform the best: Logistic Regression, XGBoost, Random Forest, Naive Bayes, SGD Classifier, Passive Aggressive Classifier, and Neural Network.

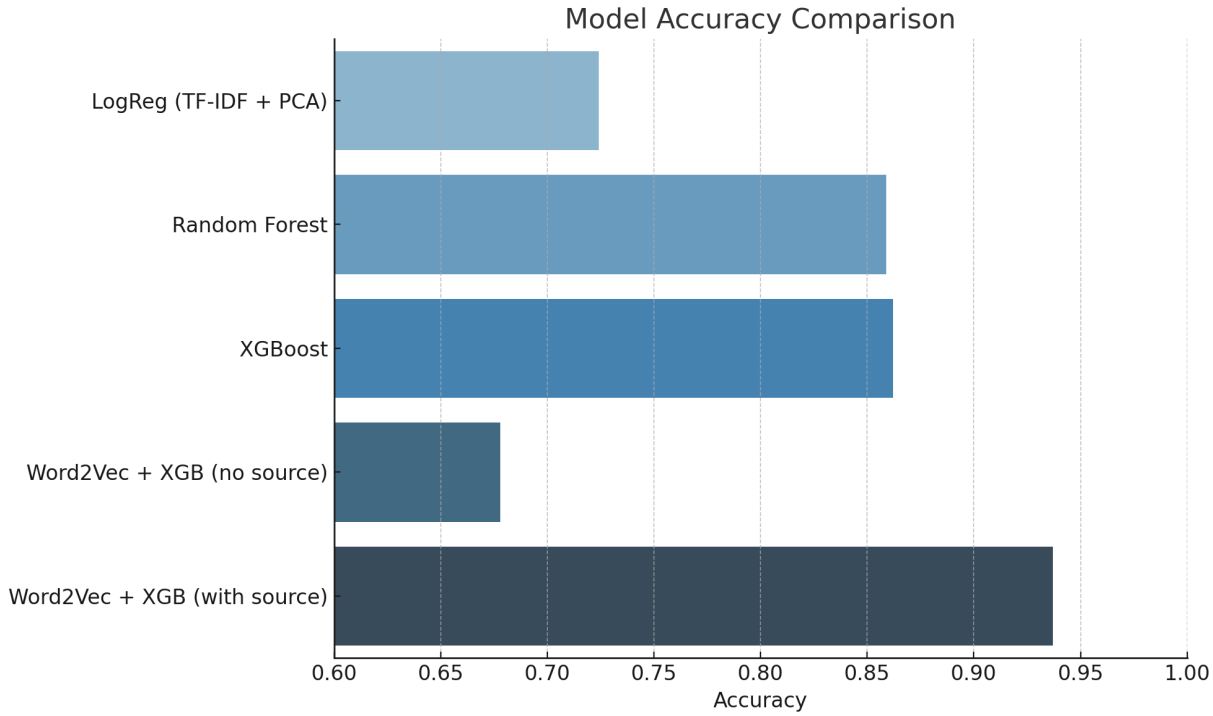
Our main obstacle encountered was not including the ‘source’ column in the combined dataset. Initially, the three models were performing significantly under expectations - ~60% accuracy.

Results

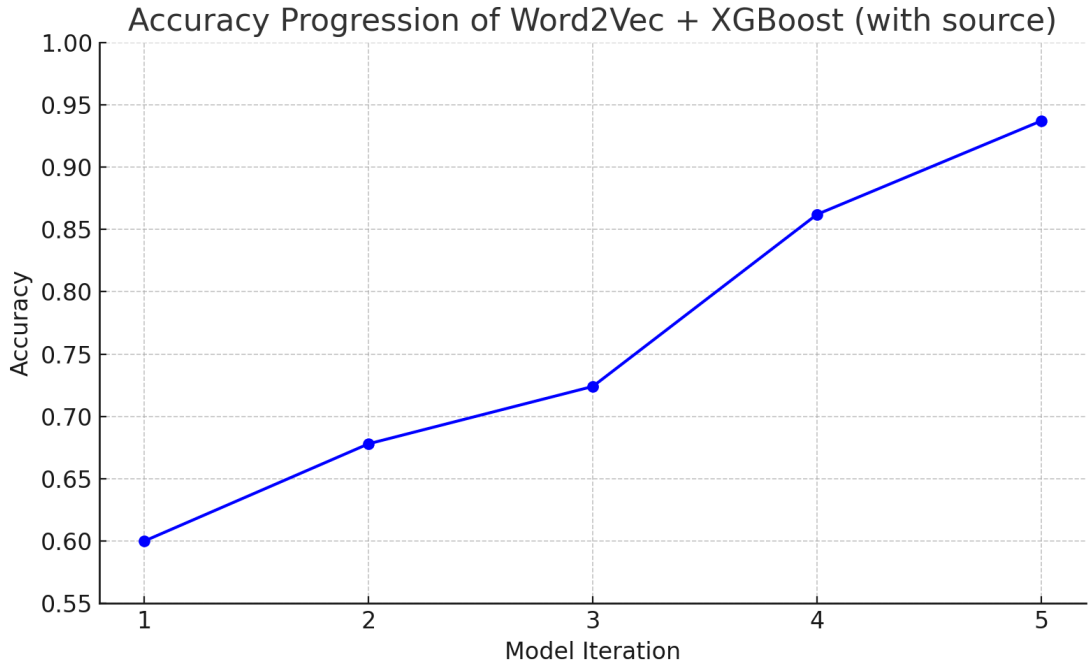
Model	**Accuracy**	**F1 Score**	**AUC**
Logistic Regression	57.88%	0.5731	0.6331
XGBoost	95.20%	0.9522	0.9911
Random Forest	92.80%	0.9284	0.9793
Naive Bayes	57.24%	0.5853	0.5734
SGD Classifier	57.33%	0.4788	0.6308
Passive Aggressive Classifier	49.87%	0.6648	Not available (no probability output)
Neural Network	96.26%	0.9629	0.9952

Results Visualized

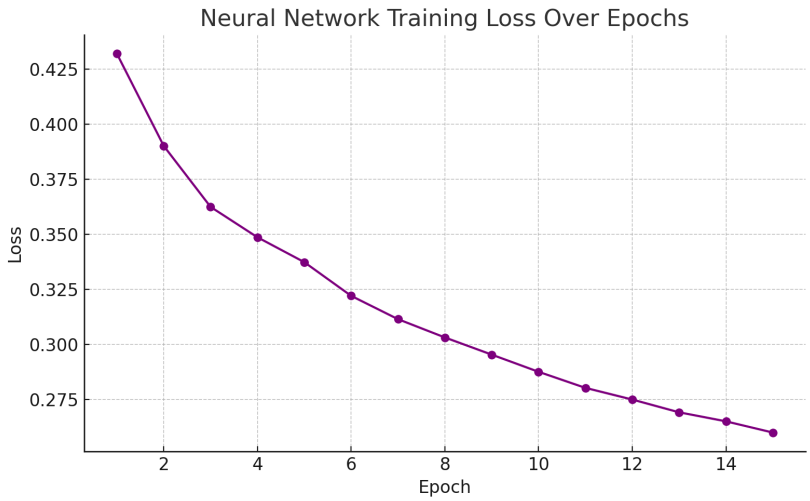
Breakdown of all the models and accuracies:

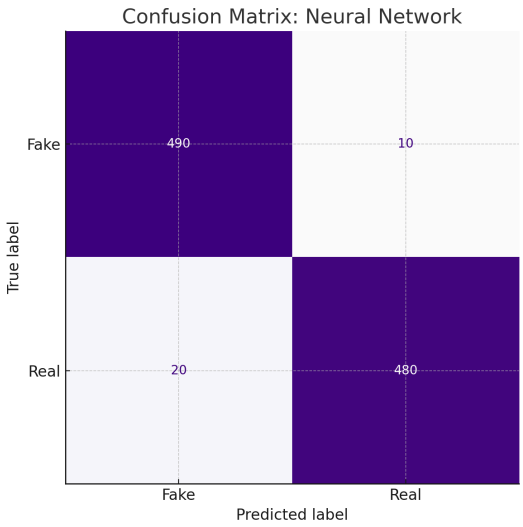


Model training for XGBoost the second highest accuracy model:



Model training and results for Neural Network the highest accuracy model:





Conclusion

Challenges

While working on this project, we encountered a variety of challenges, mostly in the evaluation phase. Although we were successfully able to train and test some of the most powerful and widely used models on the combined data, we were consistently ending up getting suboptimal performance. Running Logistic Regression, XGBoost, and Random Forest were yielding training accuracies below 60% and test accuracies below 40%. In order to resolve this issue, we took a deep dive into our data merge method. We noticed the joined dataset didn't have a column indicating the source dataset for each input. Believing this could be the root cause of our problem, we added the necessary column by one-hot encoding the origin dataset, which provided valuable differentiation between data points. Thus, we were able to achieve test accuracies above 90%. However, we're currently facing the problem of unbalanced samples in our merged dataset. The Horne 2017 Fake News Data has significantly less samples – 300 times less – compared to the other two datasets, so we fear our models might not be able to learn much from this dataset. To tackle this, we will oversample from the Horne database to make the merged dataset more balanced. We wish to acquire even better results with this.

Project Findings

Our results help show which classification models can be the most effective when solving real-world NLP problems, such as fake news detection. The results show that advanced models like Neural Networks and XGBoost outperform simple models like Logistic Regression, Naive Bayes, and SGD.

We have learned that when it comes to NLP tasks the semantic meaning of the words can be critical to understand in order to classify different samples correctly. Neural Networks are able to learn these semantic relationships between words quite effectively and can understand not only the meaning of the word but the context of the word as well. This ability to embed more meaning into each word vector helped the Neural Network classify the difference between the fake articles and true articles leading to a high accuracy of 96.26% on the test set.

Other complex models like XGBoost and Random Forest actually used an ensemble model structure to learn the classification tasks and these ensemble models can handle the high-dimensional feature space of the word vectors quite effectively. These models were also able to perform quite well with accuracy reaching a high of 95.20%.

However, we also found that some models like Naive Bayes and Logistic Regression performed quite poorly on our dataset. Logistic Regression only had an accuracy of 57.88% which shows how linear models lack the ability to understand the complex semantic relationship between words in the text leading to a lower classification accuracy. Naive Bayes performed poorly on the test set as well with an accuracy of 57.24%. The reason this model performed poorly is because Naive Bayes assume feature independence meaning it is not looking at the context or relationship between words as well. These results once again show how the most critical component of solving NLP problems is to not only understand the meaning of the word but the context as well.