

# **Detecting AI-Generated Disinformation in News Media: A Comparative Analysis of different Machine Learning Models**

**Bill Park, Daphne He, Kacey Choi**

## **Introduction**

In today's fast changing digital era, the accuracy of news transmitted across multiple media platforms has become a top priority. Even News articles from big and well established journalism companies are not immune to the spread of disinformation(Schilit, 2024). Our approach addresses the essential issue of recognizing and categorizing news stories that may have been contaminated by artificial intelligence (AI)-generated material. This information falls into four categories: Human Real (HR), Human Fake (HF), Machine Real (MR), and Machine Fake (MF). Such distinctions are critical for preserving the integrity of news dissemination and assuring the public's access to credible information(Shu et al., 2020).

The issues posed by sophisticated AI technologies capable of producing news content that resembles true human writing serve as motivation for our research. These technologies can create news items with high levels of similarity to human generated articles, making it increasingly difficult to determine the credibility of the source(Tandoc et al., 2018). The consequences of such technology are far-reaching, altering public opinion, political landscapes, and even financial markets(Guess et al., 2020).

Our research is motivated by the need to find a strong solution for detecting AI-generated fake news effectively. This issue is underlined by examples of fake content being mistakenly pushed by credible news sites, resulting in extensive transmission of incorrect information. To address these challenges, we have explored several machine learning models: Logistic Regression, Naive Bayes, XGBoost, and Support Vector Machines (SVM). Each model has been used to be trained and tested to evaluate its effectiveness in accurately classifying the different types of news articles. Our approach leverages advanced algorithms to analyze textual data, extracting and learning from patterns that distinguish articles in each category.

Our primary research question is: "How effective are various machine learning models at differentiating between human-generated and AI-generated fake news, and which model provides the highest accuracy using our dataset?"

## **Related Work**

The research paper "Adapting Fake News Detection to the Era of Large Language Models" laid the groundwork for our initiative on detecting AI-generated deception in the media. This research examined the complicated landscape of fake news in the era of powerful AI technology, empha-sizing the importance of distinguishing between machine-generated false news and human-written information. It ex-posed a key flaw in earlier research, which either ignored the reality that not all machine-generated news is phony or assumed all human-written news is genuine. Inspired by this study, we used its innovative approach to categorize news stories into four categories—Human Real, Human Fake, Machine Real, and Machine Fake—to gain a more detailed knowledge of how various types of information in-teract. This classification strategy is critical for teaching our models to effectively recognize and label news, ad-dressing the complications provided by modern AI-powered content creation technologies.

The research paper “Truth, lies, and Automation” explores the capabilities and limitations of AI models like GPT-3 in producing deceptive content, which is crucial for developing effective detection mechanisms. The paper provides valuable insights into the potential linguistic patterns and narrative structures employed by AI-generated misinformation, which can inform the design of our detection framework. Building on their research, we can look into how TF-IDF vectorization can highlight strong features in machine text and the ability of different types of models in classifying human and AI-generated text as fake or real through textual patterns.

The research article, “In New Media & Society,” in detail explores the effects of knowledge about automated journalism on the evaluation of news generated by algorithms. They base the HAIL-TIME and Persuasion Knowledge Models, demonstrating that individuals with greater automated journalism tend to favor algorithmically generated news. The study also examines how different perceptions of automated journalism as either more machine-like or human-like can influence news evaluations based on user age and knowledge levels. This finding underscores the importance of considering user perceptions

in model development, as these perceptions could significantly affect the models' effectiveness in real-world applications. Consequently, their findings and work guided us on our approach to training and evaluating models, ensuring that models not only perform well technically but also align with realistic user expectations and experiences in distinguishing AI-generated content from human-written texts.

## Methods



Figure 1 :Project pipeline

**Data Acquisition:** Our dataset was sourced from the study titled "Adapting Fake News Detection to the Era of Large Language Models" by Su et al. This dataset includes news articles categorized into four distinct classes: Human Real (HR), Human Fake (HF), Machine Real (MR), and Machine Fake (MF). The dataset is derived from the GossipCop++ and PolitiFact datasets(Shu et al., 2020).

- **GossipCop++:** This dataset primarily focuses on celebrity gossip and entertainment news. This includes 4,084 human-written fake news (HF), 4,084 human-written real news (HR), and 4,169 machine-generated real news (MR) articles, along with 4,084 machine-generated fake news (MF) articles.
- **PolitiFact++:** This dataset is centered around political news and fact-checking, providing a diverse set of articles that cover various political topics and statements. It is a smaller dataset, containing 97 human-written fake news (HF), 97 human-written real news (HR), 132 machine-generated real news (MR) articles, and 97 machine-generated fake news (MF) articles.

### Features of the Dataset:

- **id:** A unique identifier for each article.
- **text:** The main body of the news article, which includes the text that will be analyzed.
- **title:** The headline of the news article.
- **description:** A brief summary or description of the news article.

**Text Cleaning:** Standardization was achieved by removing non-essential characters and formatting inconsistencies to ensure uniformity across varied

inputs, which is essential for maintaining model accuracy. This step involved stripping HTML tags, lowercasing text, and removing punctuation and special characters.

**Tokenization:** We converted texts into tokens to facilitate detailed linguistic analysis, which is pivotal for any text-based machine learning task.

**Removal of Common Words:** We filtered out stopwords, special characters, overly common, and non common words to emphasize more predictive words, enhancing the machine models' ability to distinguish between categories.

**TF-IDF Vectorization:** Term-Frequency-Inverse Document Frequency method was utilized to transform texts into a set of feature vectors. It helps in distinguishing the influence of frequently appearing words in the dataset, which might otherwise skew the model's learning phase.

## Machine Learning Models

### Logistic Regression:

The Logistic Regression model uses the logistic function  $\sigma(z) = 1/(1+e^{-z})$  where  $z$  is the linear combination of the TF-IDF vectorized features and their corresponding weights. The logistic function transforms any real-valued number into the range  $[0, 1]$ . In our case of binary classification, for the first step classifying human vs AI news, values under 0.5 are predicted as human news, while values over 0.5 are predicted as AI news. In the second step, for real vs fake in each category, model output values under 0.5 are predicted real and values over 0.5 are predicted fake. We chose to use logistic regression because it is great for binary classification and also allows for fine-tuning with scikit-learn's GridSearchCV and RandomizedSearchCV utilities. GridSearchCV has values of different hyperparameters to tune, such as the regularization penalty, the regularization strength (C), the solver algorithm, and the maximum number of iterations. Another option is RandomizedSearchCV, which is a faster alternative that samples a fixed number of parameter settings from the specified parameter distributions instead of trying all possible combinations. These utilities can help find the

optimal hyperparameters for the logistic regression model.

**Naive Bayes:**

The Naive Bayes model is a probabilistic classifier based on the Bayes' theorem and the assumption of feature independence. In the context of text classification, the Naive Bayes algorithm calculates the probability of a document belonging to a particular class (e.g., human news or AI news) based on the prior probabilities of each class. We chose to use the Naive Bayes classifier because it is a simple and efficient algorithm that can handle high-dimensional data, such as text data represented as TF-IDF vectors. Additionally, the Naive Bayes classifier is known to be robust to irrelevant features and can provide a good baseline for more complex models. While the Naive Bayes classifier does not have as many hyperparameters as logistic regression, it can still benefit from tuning certain parameters, such as the smoothing parameter (alpha) or the class prior probabilities. In scikit-learn, the MultinomialNB class used for Naive Bayes text classification provides an alpha parameter for smoothing, which can be adjusted to prevent zero probabilities and improve the model's performance.

**XgBoost:**

XGBoost (eXtreme Gradient Boosting) is a machine learning algorithm best known for its performance in supervised learning tasks, particularly classification and regression. It falls under the ensemble learning paradigm, which combines the predictions of numerous weak learners to create a robust and accurate model. XGBoost's novelty resides in the sequential creation of a powerful ensemble of decision trees using boosting and gradient boosting approaches. XGBoost uses boosting to successively train numerous weak learners, frequently shallow decision trees, to correct the mistakes of their predecessors. It constructs each consecutive tree using the residuals (errors) of the previous trees, focusing on areas where the model performed poorly. Furthermore, XGBoost uses gradient boosting to iteratively optimize model performance, minimizing a loss function by greedily adding weak learners. It computes the gradient of the loss function in relation to the model's predictions, and then trains each succeeding tree to minimize this gradient

**SVM (Support Vector Machine):**

The Support Vector Machine is well known to be a learning algorithm that uses supervised learning models to solve complex classification, regression. For our project, the model was chosen for their effectiveness in handling high-dimensional text data. For this particular model testing, SVMs with a linear kernel were employed to perform binary and multi-class classification tasks. We focused on the application of Support Vector Machines

(SVM) for the classification of text data in different categories: Human Real, Human fake, Machine Real, and Machine Fake. This approach was part of a broader collaborative effort where different team members explored various models using the same datasets. If you do not follow the above requirements, it is likely that we will be unable to publish your paper.

**Why these models were chosen:**

We chose these methods based on their established effectiveness in text classification tasks and their ability to handle high-dimensional data typical of textual datasets. Logistic Regression and Naive Bayes were selected for their simplicity and interpretability, providing a strong baseline for comparison. XGBoost was chosen for its robust performance in various machine learning competitions and its ability to handle complex data structures through boosting. SVM was included for its efficiency in high-dimensional spaces and its robustness in classification tasks(Kille, 2020).

**Results**

Model	Accuracy	Rank
Logistic regression	HR vs HF: 0.76887 MR vs MF: 0.49740 H vs M: 0.46539	3
Naive bayes	HR vs HF: 0.67007 MR vs MF: 0.49740 H vs M: 0.43888	4
Xgboost	HR vsHF:0.63917 MR vsMF:0.88209 H vs M:0.87884	2
SVM	HR vs HF:0.79661 MR vs MF:0.89130 H vs M:0.93269	1

Figure 2 : Summary of Accuracies for distinguishing different types of Text for each models

In a comparative evaluation of four machine learning models—Logistic Regression, Naive Bayes, XGBoost, and SVM—across three classification tasks: Human Real vs. Human Fake, Machine Real vs. Machine Fake, and Human vs. Machine generated content, the performance varied significantly across two phases: Training and Testing, and Validation. During the Training and Testing phase, Logistic Regression showed strong performance, particularly in distinguishing Machine Real from Machine Fake with an accuracy of 0.99014 and Human from Machine at 0.977562. Naive Bayes excelled in the Machine Real vs. Machine Fake task with an accuracy of 0.9809, but XGBoost and SVM showed variable results, with XGBoost peaking at 0.93095 for Machine Real vs. Machine Fake, and SVM excelling in the same category at 0.95888.

However, during the Validation phase, SVM emerged as the top performer with overall highest accuracies, notably 0.79661 in Human Real vs. Human Fake and 0.93269 in Human vs. Machine. XGBoost, ranked second, had its best showing in Machine Real vs. Machine Fake at 0.88209 accuracy. Logistic Regression and Naive Bayes, however, saw a decline in performance, particularly evident in Logistic Regression's drop to 0.76887 in Human Real vs. Human Fake and Naive Bayes falling to the last rank with its highest validation accuracy at 0.67007 in Human Real vs. Human Fake. These results underline the necessity for further tuning and validation to enhance model stability and performance across different phases.

## Experiment Evaluations

### Logistic Regression:

#### Questions Addressed by Analysis:

- 1.How accurate is the logistic regression model at distinguishing between human vs machine-generated text? What are the best options for fine tuning?
- 2.What are some features that strongly contribute to model performance and what are some of the feature patterns?
- 3.How well does the model perform on new unseen data in a different category of news?

The logistic regression model performed very well in classifying human vs machine-generated text with around 97% accuracy. Human real vs fake classification had approximately 92%, and machine real vs fake classification performed even better with approximately 99% accuracy.

### Feature Analysis:

Since we used TF-IDF as our optimal method of vectorization, we also looked into the top features or words with the largest positive coefficients indicating each class.

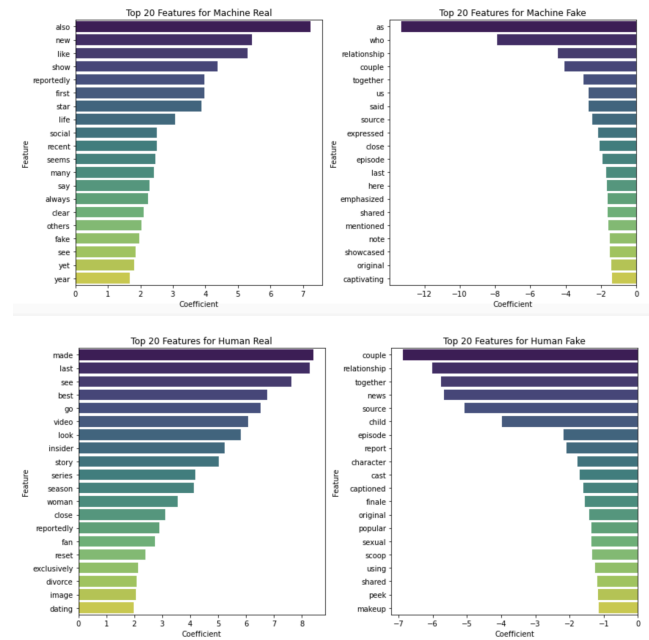
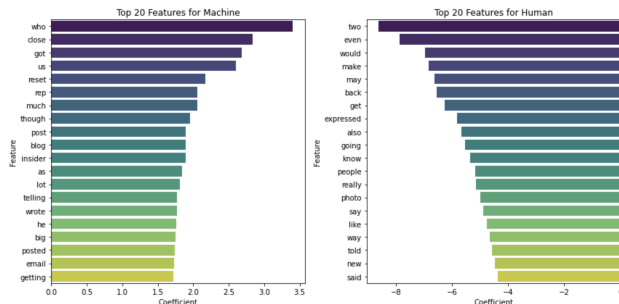


Figure 3 :Logistic Regression Top 20 Features for each Comparison

Unigrams overall are stronger feature indicators than bigrams for all classifications. The logistic regression model's top features reveal distinct patterns between human and machine-generated content. Machine-generated content tends to focus on online activities and communications, evidenced by words like "post," "blog," "email," and "posted," and adopts a more neutral or formal tone with pronouns like "who," "he," "us," and "as." In contrast, human-generated content is characterized by conversational and opinion-based expressions, such as "would," "make," "may," and "like," and uses words like "people," "really," "say," "told," and "said" to convey a more personal and emotive communication style.

Within the categories of real and fake content, machine-generated real content focuses on current or timely events, highlighted by words like "new," "recent," "first," and "year." Fake machine-generated content, on the other hand, emphasizes personal relationships and interactions, using words such as "relationship," "couple," "together," "close," and "shared." Similarly, real human content often refers to insider perspectives and multimedia elements with words like "video," "image," "look," and "insider," while fake human content, much like its machine counterpart, centers around personal relationships and social dynamics, using terms like "couple," "together," and "child." These patterns suggest that fake content, regardless of its origin, often seeks to engage and provoke through emotionally charged topics.

### Fine-tuning:

We used GridSearchCV to systematically search through a specified parameter grid and evaluate the model's performance using cross-validation to determine the optimal hyperparameters. Results showed that the regularization strength (C) of 10, penalty of 'l2', the solver algorithm (solver) of 'liblinear,' and the maximum number of iterations (max\_iter) of 100 provided the best performance.

### Cross Validation on New PolitiFact Data:

Following cross validation on new data from Politico, accuracy for human vs machine news fell to approximately 47%, human real vs human fake prediction accuracy fell to approximately 76%, and machine real vs machine fake prediction accuracy fell to approximately 50%. Based on the new data cross validation confusion matrix for human vs machine-generated classification, there appears to be a lot of false negatives, where machine news is mistakenly classified as human news. There is also a tendency for fake news to be mistakenly classified as real for the human real vs fake and machine real vs fake models. The drastic drop in performance may indicate overfitting of the model on the training data.

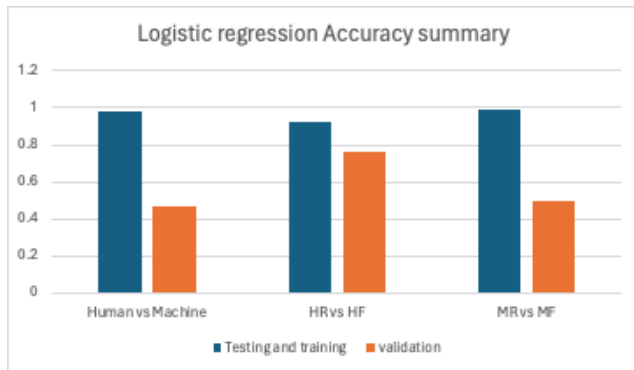


Figure 4 :Logistic Regression Accuracy Summary Graph

### Naive Bayes:

#### Questions Addressed by Analysis:

- 1.How accurate is the Naive Bayes model at distinguishing between human vs machine-generated text?
- 2.What are some features that strongly contribute to model performance and what are some of the feature patterns?
- 3.How well does the model perform on new unseen data in a different category of news?

### Detailed Analysis and Observations:

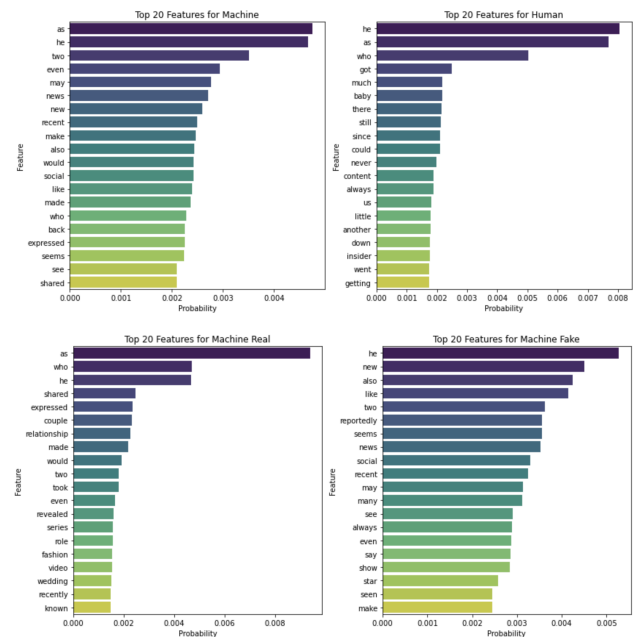
The Naive Bayes model performed very well in classifying human vs machine-generated text with around 97% accuracy. Human real vs fake classification had approximately 92%, and machine real vs fake classification

performed even better with approximately 99% accuracy. Overall, there was no significant difference in Logistic Regression and Naive Bayes model prediction performance.

### Feature Analysis:

The Naive Bayes machine learning model's top features reveal notable patterns in differentiating between human and machine-generated content, as well as between real and fake content. Machine-generated content often includes words such as "news," "also," "social," and "shared," indicating a focus on news, updates, and the sharing of social and current events. In contrast, human-generated content is marked by words like "still," "since," "could," and "always," which suggest temporal statements and reflect on past events or continuous states.

For both real and fake content, there are observable differences in focus. Real content, whether machine or human-generated, tends to center on personal relationships, using words like "couple," "relationship," "wedding," "together," and "shared." On the other hand, fake content from both origins is more focused on reporting timely events, with terms such as "recent," "reportedly," "new," and "insider" being prevalent. This indicates that the Naive Bayes model considers real content to be associated with personal and emotional narratives, whereas fake content seeks to capture attention through the immediacy and novelty of current events.





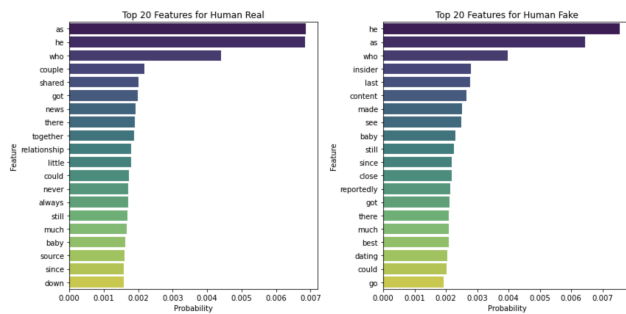


Figure 5: Naive Bayes Top 20 Features for each Comparison

### Cross Validation on New PolitiFact Data:

Following cross validation on new data from Politico, accuracy for human vs machine news fell to approximately 44%, human real vs human fake prediction accuracy fell to approximately 67%, and machine real vs machine fake prediction accuracy fell to approximately 50%. Similarly, to the Logistic Regression model, the Naive Bayes model performance drastically decreased, with overwhelming false negatives in classifying human vs machine news. Despite having similar drops in performance, cross validation results showed that Naive Bayes prediction accuracy was overall still lower than that of Logistic Regression in each category, which led us to decide not to proceed with the Naive Bayes model.

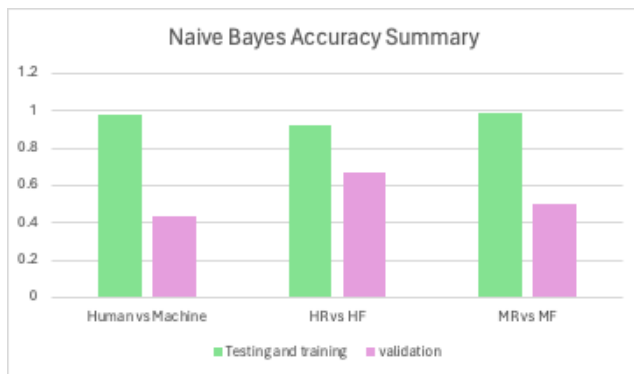


Figure 6: Naive Bayes Accuracy Summary Graph

## XgBoost

### Questions Addressed by Analysis:

1. How accurate is the XgBoost model at distinguishing between Human Real vs. Human Fake and Machine Real vs. Machine Fake news articles?
2. How does the XgBoost model's accuracy on machine-generated text compare to its accuracy on human-generated text?
3. Can XgBoost maintain high performance levels when scaled to larger, more diverse datasets?

XGBoost demonstrated a strong capability in distinguishing between Machine Real and Machine Fake news with an accuracy of approximately 93%, showing particular effectiveness in identifying nuances in machine-generated texts. However, its performance in differentiating Human Real from Human Fake news was less robust, with an accuracy around 77%. This highlights an area for potential improvement, especially in enhancing the model's sensitivity to variations in human-written content.

When tested on new, unseen data, XGBoost maintained a respectable level of accuracy, achieving around 87% in distinguishing all human texts from machine-generated articles, suggesting reasonable generalization capabilities. Nonetheless, the accuracy for Human Real versus Human Fake dropped to approximately 64%, indicating challenges in generalizing to human-written discrepancies in an unseen dataset. This suggests that while XGBoost is adept at identifying patterns in machine-generated texts, it struggles more with the subtleties of human-generated content.

The model's higher accuracy on machine-generated texts compared to human-generated texts may be due to clearer systematic differences in machine-generated content that the model can recognize more easily. However, maintaining high performance with larger and more diverse datasets remains a challenge, as increasing complexity and variability can impact the model's effectiveness without additional tuning and adaptation.

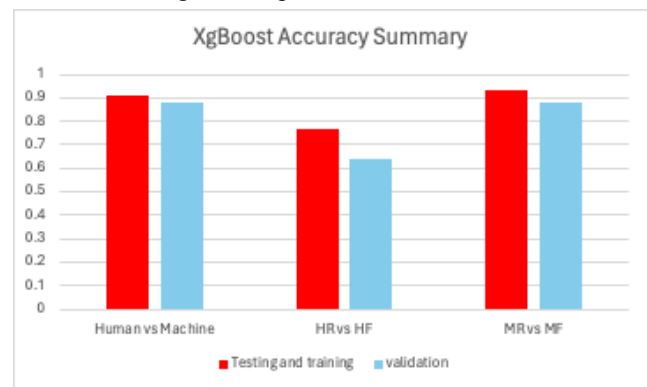


Figure 7: XgBoost Accuracy Summary Graph

### Feature Analysis:

Feature analysis for XGBoost revealed significant insights. For Human vs. Machine classification, top features included "told," "however," "fan," "expressed,"

and "despite," indicating that human-generated content often involves narrative elements, emotional engagement, and complex sentence structures. In distinguishing Human Real (HR) from Human Fake (HF) news, features such as "source," "insider," "pitt," "gossip," and "report" were crucial, suggesting that celebrity names and gossip-related terms are key indicators of fake human articles. For Machine Real (MR) vs. Machine Fake (MF) classification, top features included "source," "rumor," "close," "many," and "recent," helping capture stylistic patterns and the repetitive nature of machine-generated content. These insights into feature importance aid in understanding the model's decision-making process and highlight areas where further refinement could enhance performance.(See Appendix A)

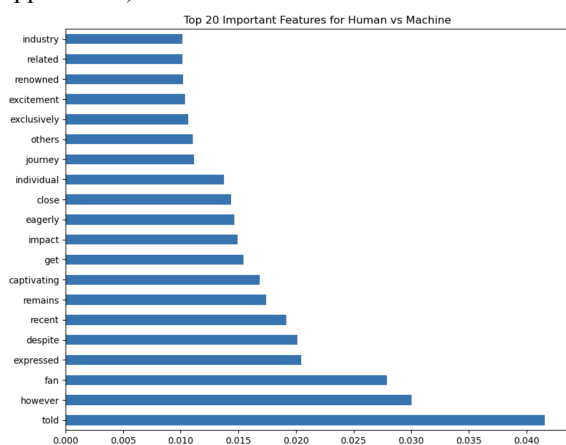


Figure 8 :Top 20 features for Human vs Machines for XgBoost

## SVM (Support Vector Machine):

### Questions Addressed by the Analysis:

- 1.Can SVM models effectively differentiate and classify between human-generated and machine-generated texts?
2. How well does SVM distinguish between real and fake texts within human and machine categories?
3. Does the model demonstrate signs of overfitting, and how does this impact its predictive accuracy on unseen data?

### Detailed Analysis and Observations:

Overall the SVM model demonstrated a robust performance across various text classifications. The accuracy for Machine Real vs. Machine Fake was the highest with approximately 95% accuracy, followed by Human vs. Machine with 92% accuracy, and lastly Human Real vs. Human Fake with 79% accuracy. However, especially high accuracy led to cautious overfitting, which was also indicated by learning curves.

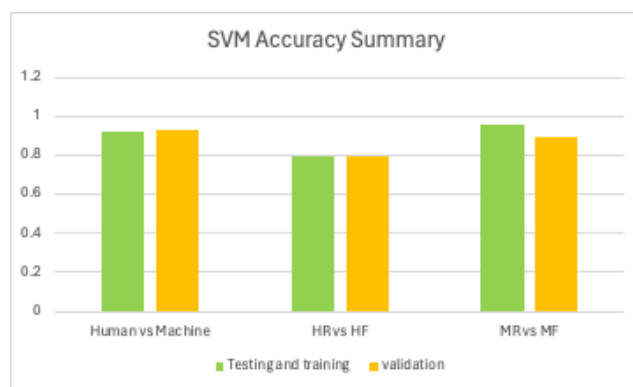


Figure 9 :SVM Accuracy summary graph

These results were initially encouraging but the skeptically high accuracy rates of the model raised questions regarding the model's ability to generalize beyond the training data. Therefore, we graphed the learning curve to take a deeper analysis of the training and validation trends of each of the model runs.

Overall, the learning curves indicated signs of potential overfitting, as training scores were consistently high across all categories while validation scores were comparatively lower, though they improved with an increase in training samples. This discrepancy suggested that while the model is learning effectively, it might be too tailored to the training dataset. To assess the model's generalizability and robustness to new environments and datasets, we decided to further test the SVM model using a new dataset, PolitiFact.

The model's performance on the PolitiFact dataset generally mirrored the results from the original dataset, showing high overall accuracies for all categories. Specifically, the accuracy for the Human vs. Machine all text comparison was notably high at 93.2%. This was followed by Machine Real vs. Machine Fake at 89.1% and Human Real vs. Human Fake at 79.6%. Given these results, we decided to run the learning curves again for this dataset to further investigate the model's performance across these different categories. Furthermore, to address the issue of overfitting, we fine-tuned the model by reducing the number of features, determining the optimal 'C' parameter value using GridSearchCV, and increasing the fold size for cross-validation. And even after these fine-tuning processes the accuracy level for SVM remained decently high across all the different categories.

### Feature Analysis:

From the feature extraction using the SVM model, the results indicated distinct linguistic patterns across different categories. In the Human vs. Machine classification, the model identified narrative connectors such as "expressed," "however," and "stating" as more prevalent in machine-generated texts. For the Machine Real vs. Fake classification, terms like "many," "source," and "rumor" were commonly associated with machine-generated fake texts. Lastly, in the Human Real vs. Fake classification, key terms including "source," "insider," and "exclusively" demonstrated the model's capability to authenticate human content(See Appendix A).

## Discussion

Our project assessed the efficacy of several machine learning models—Logistic Regression, Naive Bayes, XGBoost, and SVM—in detecting fake news generated by humans or AI across four distinct categories: Human Real (HR), Human Fake (HF), Machine Real (MR), and Machine Fake (MF).

Through our testing and evaluation, distinct performance patterns emerged among these models. SVM outperformed in accuracy compared to other models, particularly in terms of stability and accuracy during validation scenarios.

This suggests that SVM is capable of handling the complexities of high-dimensional text data, making it a robust tool for this application.

Logistic Regression and Naive Bayes, while initially showing promise in training phases, experienced significant performance drops during validation, underscoring potential issues of overfitting and a high sensitivity to the specific distributions of training data. These models' declines highlight the critical need for models that not only learn effectively but can also generalize well to new, unseen data sets. XGBoost demonstrated a strong ability to differentiate between MR and MF articles, with an accuracy of around 93%. However, it faced challenges with the HR vs. HF distinction, pointing to areas where further refinement is needed. This finding underlines the necessity of tailoring model selection and tuning to the specific challenges and characteristics of the data being analyzed.

From these results, we analyzed several factors of SVM that might explain the outperforming performance of the model. First, we found that SVM's robustness in handling

high-dimensional data, crucial for text data with numerous word and phrase features, was a great fit for our dataset. Next, its ability to find the optimal hyperplane and maximize the margin between different classes ensured more effective separation in complex datasets, surpassing the capabilities of other models. Additionally, the kernel trick allows SVM to manage complex patterns by transforming data into higher-dimensional spaces, unlike logistic regression and Naive Bayes, which assume feature independence and struggle with intricate patterns. While XGBoost can handle non-linearity, it often requires extensive tuning. And lastly, the effective regularization parameter  $C$  found via hyperparameter tuning, balanced the trade-off between low training and testing errors, reducing overfitting and ensuring generalization to new data.

Building on our findings, our best performing model definitely suggested its potential as a robust tool for detecting various categories of AI-generated articles. While our model is still undergoing the experimental phase, if further validated, we believe this approach could significantly enhance our ability to identify and mitigate disinformation. Helping to preserve the integrity of news dissemination and ensuring the public's access to reliable information.

## Conclusion and Future work

### Final Product Status

#### Try the SVM Classifier

Enter your text or article below to check if it is written by a human or a machine, and further, whether it is real or fake.

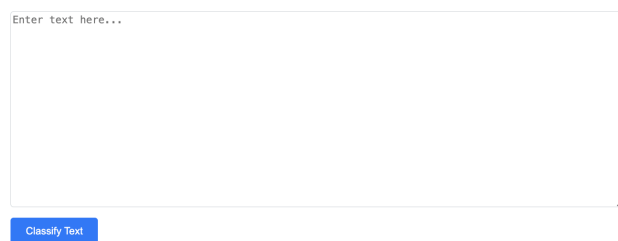
A screenshot of a web application interface for a text classifier. It features a large, empty text input area with a placeholder text "Enter text here...". Below the input area is a blue button with the text "Classify Text" in white.

Figure 10: Web app classifier prototype

We have developed a web application using Flask that allows users to input text and classify it into one of four categories: Human Real (HR), Human Fake (HF), Machine Real (MR), and Machine Fake (MF). This application leverages our best-performing SVM model, providing a user-friendly interface for real-time text classification. This tool is designed to help users identify the authenticity and



origin of news articles, thereby enhancing the public's ability to access credible information.

our models to develop reliable tools capable of effectively identifying and mitigating the spread of misinformation.

### **Future Work**

In the future, we aim to enhance the capabilities and robustness of our models and the web application. Firstly, we plan to diversify our training data by collecting a more extensive and varied dataset from multiple news sources. This will include different genres and languages to ensure our models are not biased towards specific types of content and can generalize better across various contexts.

We will also explore more sophisticated Large Language Models (LLMs) and advanced vectorization techniques. Testing these with our existing models will help us evaluate their impact on accuracy and identify the most effective methods for distinguishing between real and fake, as well as human and machine-generated, content. By incorporating state-of-the-art vectorization methods, we aim to capture more nuanced patterns in the text.

Conducting a deeper analysis of linguistic features that differentiate human from machine-generated texts is another priority. We will delve into the specifics of these features and integrate them into our models to enhance their precision. This detailed analysis will help us understand better the subtleties that can indicate the authenticity of a news article.

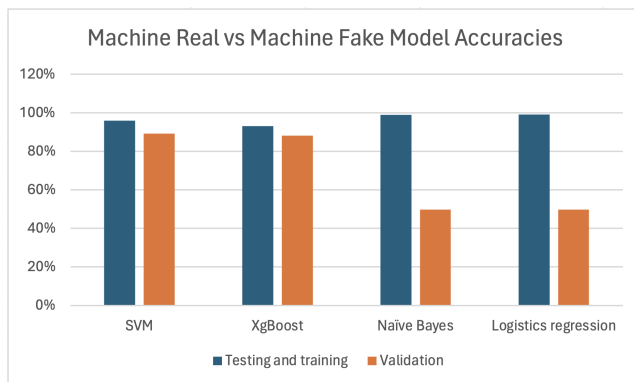
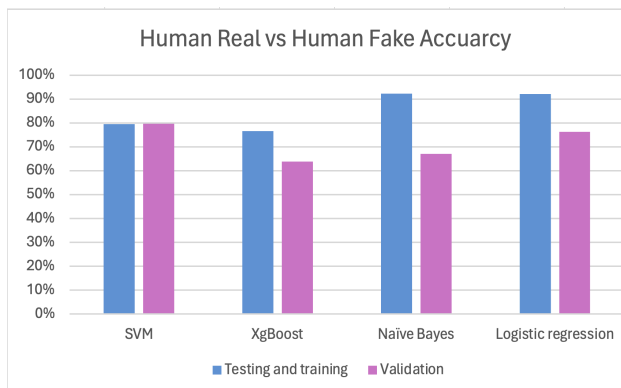
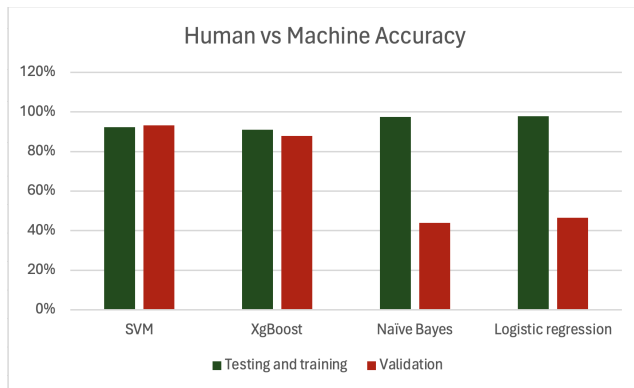
Additionally, we plan to explore advanced model architectures by combining traditional machine learning techniques with deep learning approaches. For instance, integrating logistic regression with neural networks could allow us to capture more complex patterns in the text, leading to improved performance. Hybrid models may offer the best of both worlds, leveraging the strengths of each approach to tackle the intricacies of text classification more effectively.

Collaboration will be a key component of our future work. We intend to collaborate with other research groups to share insights and advancements. Furthermore, we plan to publish our code and models as an open-source project. This will not only encourage community contributions and improvements but also promote transparency and collective progress in the field of AI-generated content detection.

In conclusion, our research has established a solid foundation for using machine learning to detect fake news, with the SVM model showing particularly promising results. However, we recognize the challenges such as overfitting, sensitivity to training data distribution, and ethical concerns regarding AI applications. Addressing these issues is crucial, and we are committed to refining

## Appendix A)

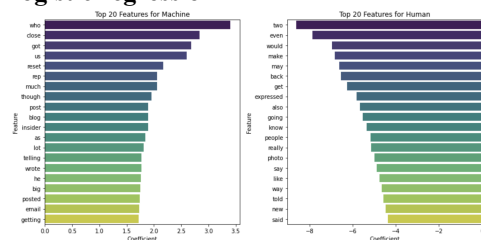
### Accuracy graphs for each Comparisons



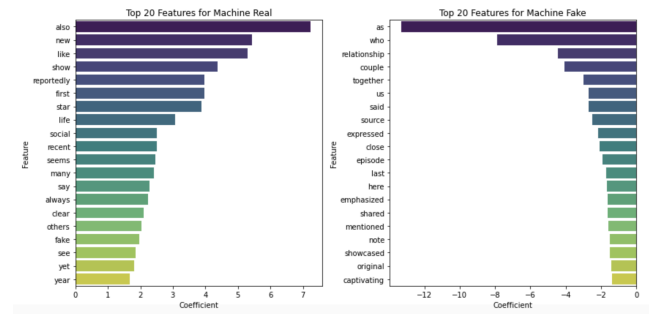
## Appendix B)

### Feature Extraction for each models

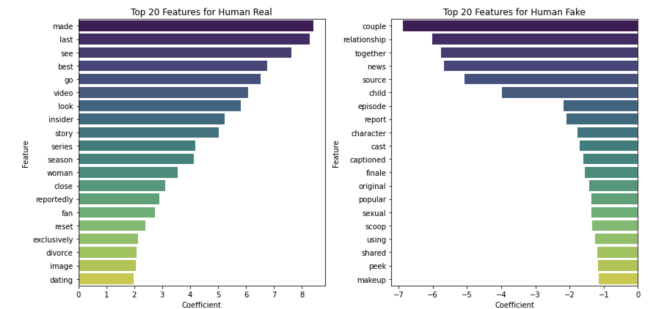
#### Logistic regression



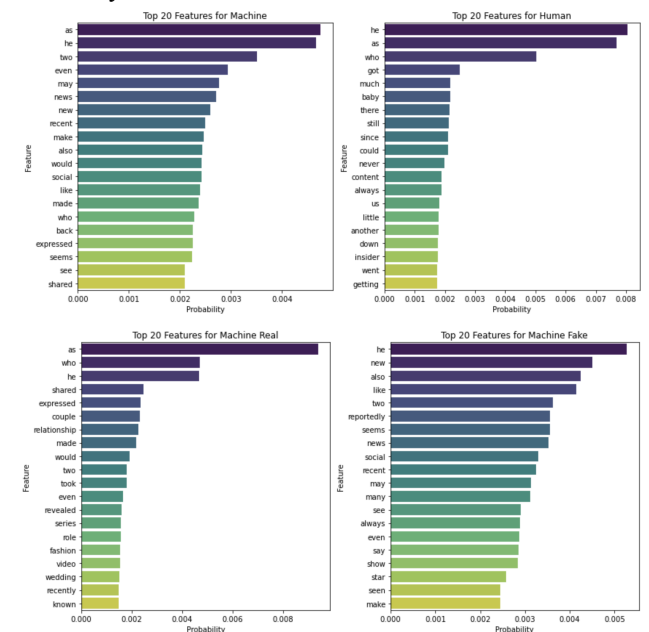
#### Machine vs Machine fake

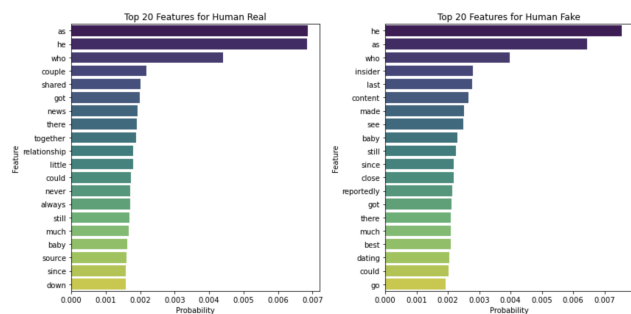


#### Human real vs Human fake

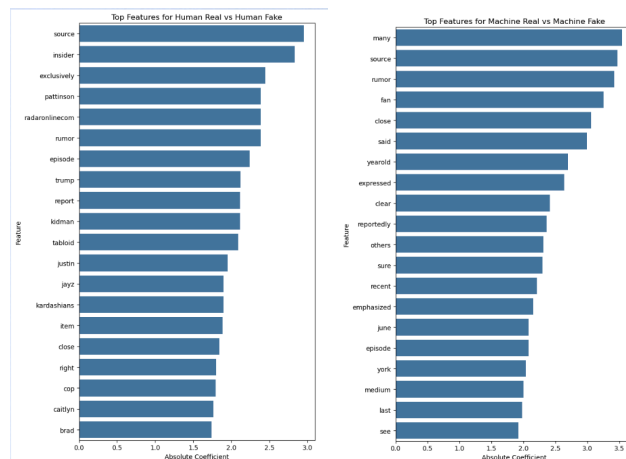


#### Naive Bayes

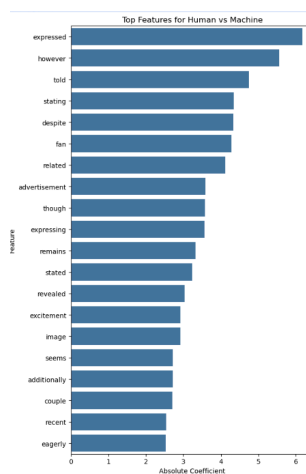
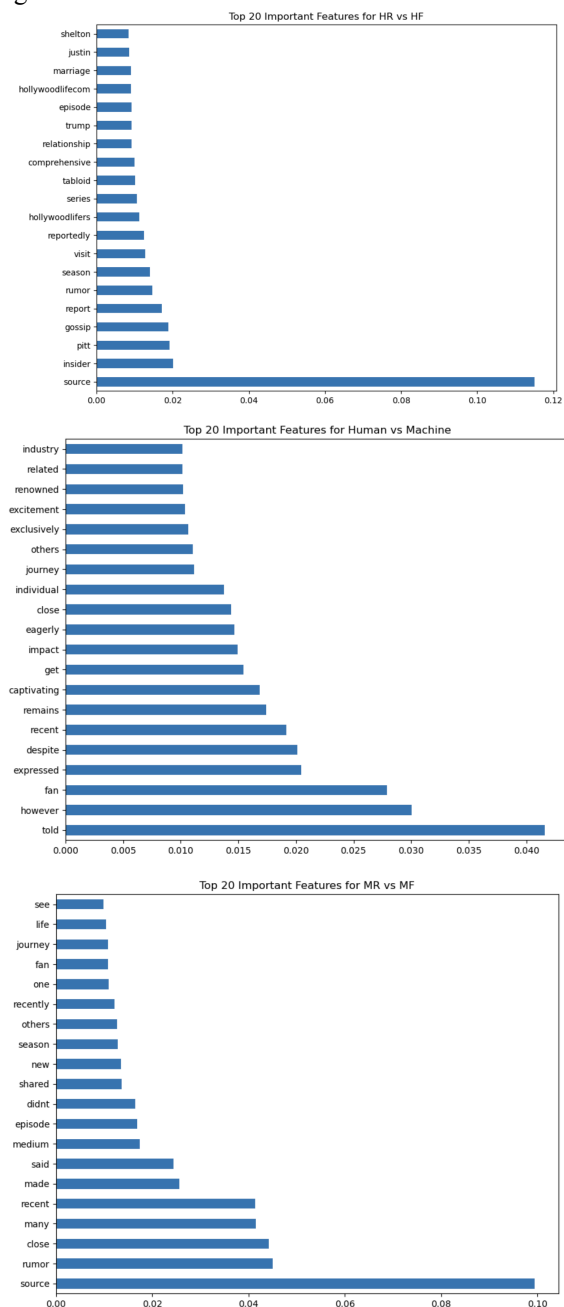




## SVM



## XgBoost



## References

- Akram, A. (2023). An empirical study of AI-generated text detection tools. *Advances in Machine Learning & Artificial Intelligence*, 4(2).  
<https://doi.org/10.33140/amlai.04.02.03>
- Amy Mitchell, M. J. (2021, February 22). 3. misinformation and competing views of reality abounded throughout 2020. Pew Research Center.  
<https://www.pewresearch.org/journalism/2021/02/22/misinformation-and-competing-views-of-reality-abounded-throughout-2020/>
- Buchanan, B., Lohn, A., Musser, M., & Sedova, K. (2023, August 30). Truth, lies, and Automation. Center for Security and Emerging Technology.  
<https://cset.georgetown.edu/publication/truth-lies-and-automation/>
- Guess, A. M., Nyhan, B., & Reifler, J. (2020). Exposure to untrustworthy websites in the 2016 US election. *Nature Human Behaviour*, 4(5), 472–480.  
<https://doi.org/10.1038/s41562-020-0833-x>
- Jiang, S., & Wilson, C. (2018). Linguistic signals under misinformation and fact-checking. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW), 1–23.  
<https://doi.org/10.1145/3274351>
- Jang, W. (Eric), Kwak, D. H., & Bucy, E. (2022). Knowledge of automated journalism moderates evaluations of algorithmically generated news. *New Media & Society*, 146144482211425.  
<https://doi.org/10.1177/14614448221142534>
- Kille, L. W. (2020, December 8). Committee of Concerned Journalists: The principles of journalism. The Journalist's Resource.  
<https://journalistsresource.org/home/principles-of-journalism/>
- Monteith, S., Glenn, T., Geddes, J. R., Whybrow, P. C., Achtyes, E., & Bauer, M. (2023). Artificial Intelligence and increasing misinformation. *The British Journal of Psychiatry*, 224(2), 33–35.  
<https://doi.org/10.1192/bjp.2023.136>
- Schilit, N. (2024, May 15). Leading international editors share how they address disinformation's harm to democracies. International Journalists' Network.  
<https://ijnnet.org/en/story/leading-international-editors-share-how-they-address-disinformations-harm-democracies>
- Tandoc, E. C., Jenkins, J., & Craft, S. (2019). Fake news as a critical incident in journalism. *Journalism Practice*, 13(6), 673–689. doi:10.1080/17512786.2018.1562958
- Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380), 1146–1151. <https://doi.org/10.1126/science.aap9559>