Evan Magee & Tanneh Jah
Professor Jang
CmpSc 463 - 001
12/2/2025

Description

This project aims to create a tool for helping identify fake news or articles that contain misinformation using a classification style model. It also gives the user other tools for analyzing the input articles such as a word cloud, for providing visuals showcasing the weights and frequency of certain words, an article summarizer, and a tab for showing what keywords can cause an article to be deemed fake. The program utilizes a handful of libraries for helping do these tasks. This includes tkinter for GUI, newspaper for article parsing, wordcloud for word cloud creations, pickle for model storage, pandas for csv parsing, numpy for array functions, and sklearn for ML functions and performance analytics.

Significance of the Project

This project is meaningful because it helps to address the growing problem of fake news and misinformation which has taken the world and mainly the internet by storm. It has become harder and harder to identify what information and news sources provide truthful and trustworthy stories so the need for tools to help identify these sources or articles is dire. With this project people can input links to articles and receive a determination based on the word content of the article whether it is fake or not. It also provides analytical data and tools for further determination on whether a user should trust a source or if they should maybe look elsewhere for information.

Code Structure

This implementation of this program was done using a model training file, , and

1. model.py

This file handles all of the machine-learning parts of the project. It:

- Loads the dataset
- Builds a TF-IDF vectorizer
- Trains a **Logistic Regression** model
- Prints evaluation results
  - Roughly 99% for all given analytics.
- Saves the trained model and vectorizer into .pkl files so the GUI can use them

Files generated:

- model.pkl
- vectorizer.pkl
- X_train_tfidf.pkl
    - Used for misinform analytics

2. fake_news_app.py

This is the main GUI. It includes:

- The navigation bar
- Fake news detection screen
- Misinformation keyword analysis
- Word cloud tab
- Article summarizer

It loads the saved model and vectorizer and uses them to classify articles entered by the user.

Algorithms

- model.py
    - No inherit function but trains a logistic regression model for identifying potentially fake news.
    - The model used for this project was a logistic regression model that was trained on 44,898 articles and tested with 8980 articles. All functions within the model.py file were given the random state of 42 if it was one of their parameters to keep data the same through testing. The model results showed a 0.9901 accuracy and a 0.99 for precision, recall, and f1-score which indicates a strong model for predicting articles. This however doesn’t mean the model will get every article correct as misinformation is a constantly evolving metric but it is extremely strong at pointing out articles that may be misinformation or should more require external verification.
- fake_news_app.py
    - navigation_bar()
        - O(1)
        - This function creates the navigation buttons at the bottom of the

window that allow the user to switch between different analytics of functions for the article.

- predict_news(link, widget)
    - O(n log n + t)
        - n = number of vocabulary terms
        - t = number of words in article
    - This function downloads and parses an article from a URL, vectorizes articles using the TF-IDF vectorizer and then using the logistic regression model, it will classify if the news source is fake or real. More specifically it indicates if the article contains terms that could indicate it contains misinformation. It then lists the word contributions, showing the top 10 words in the article for classifying it as real and fake based on their weight followed by the article's text. It displays these results in an input tkinter scroll box.
- summarize_article(link, widget)
    - O(S^2 * V)
        - S = number of sentences
        - V = number of vocabulary in the TF-IDF
    - This function downloads and extracts the text from the input article. It then converts the sentence into a TF-IDF vector and computes similarity between every pair of sentences. It scores these sentences comparing its similarity to the others and picks the most important sentences. It then returns the top sentences depending on the article's length as the summary.
- clear_screen()
    - O(n)
    - This function takes the display window and iterates through all of its widgets or elements and destroys said elements. This clears the screen or produces a blank screen. It is used for mainly switching windows (this program does not do that) and ensuring the window is blank for when the program is run.
- fake_news_screen()

        - O(1)
        - This function clears the screen and displays the UI for the fake news detection tab. It creates an entry, a submission button, and the output textbox for the results of the article classification.
    - misinfo_screen()
        - O(v log v)
            - v = vocabulary size
        - This function clears the screen and displays the UI for misinformation analysis. This screen displays 2 textboxes which contain the top 50 most fake-influencing words and the top 50 most real-influencing words. It then lists other stats on those words such as its weight, average tf-idf, its max tf-idf, average tf, idf, and its frequency in the training articles.
    - article_word_cloud_screen()
        - O(T)
            - T = number of words in article
        - This function clears the screen and displays the UI for generating a wordcloud from a given URL. It contains an entry box and submission button. The generated wordcloud is then displayed on the blankspace in the window.
        - generate_word_cloud()
            - This function downloads the article and uses the wordcloud python library to generate a wordcloud for the article text.
    - article_summarizer_screen()
        - O(1)
        - This function clears the screen and displays the UI for the article summarizer tab. It creates an entry, a submission button, and the output textbox for the results of the article summary.

Verification of Algorithms

The algorithms were verified using articles found that were known to be fake or not. For testing of the model itself, the model was split using the test split in a 80/20 split. Using this split the data was found to provide a 99.01% accuracy, 99% precision, 99% recall, and an f1-score of 99%. Articles used in this project were also compared to other fake news identifiers that exist online to cross examine whether or not the article is being detected correctly. Through this testing, it found that most articles were correctly labeled like other examples but these labelings were oftentimes a result of it containing words that were oftentimes present in fake articles. This results in political and more opinion focused news articles being flagged more often than not. (sports articles as well). This could be due to the opinionated sides of these subjects which at times aren't necessarily fake but aren't true statements either.

- Predict News Testing

Predict news was found to be valid in its ability to collect article data and parse it. This was done using the newspaper python package which provides parsing tools for articles. When the provided link fails an error is displayed in the output stating so.

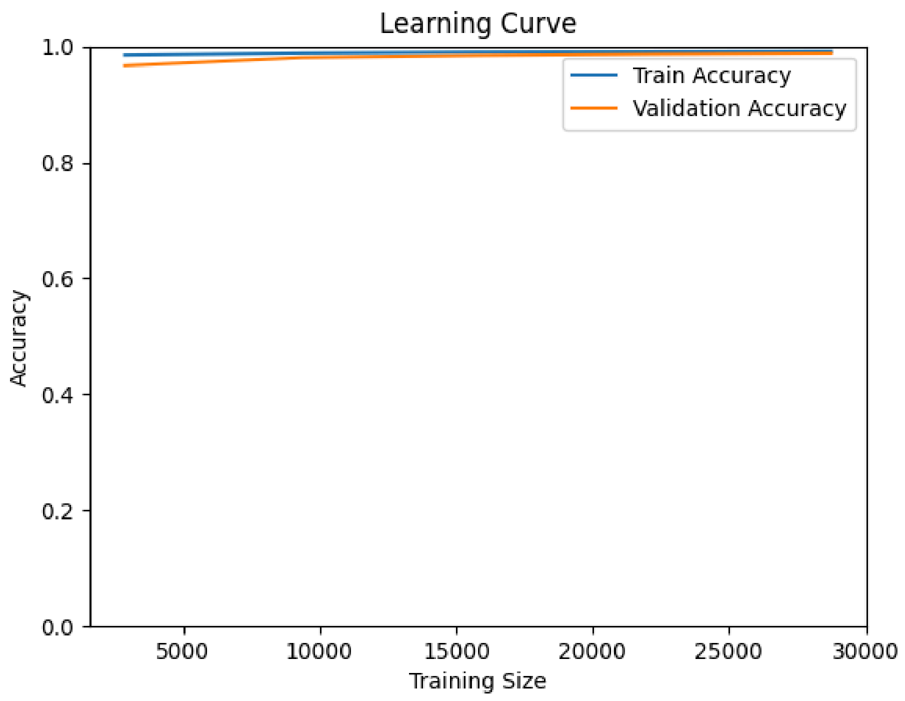
- Misinfo Analysis Testing

The output for the misinformation analysis tab is the same in all cases as its an overview of our project's model and not something the user creates when they run the fake_news_app.py file.

- Word Cloud Testing

Similar to predicting news, this tab takes in an article link input and will return a word cloud based on the words in said article. This article will use the newspaper python package to parse through an article and enter the resulting text into the wordcloud package functions. The outputting wordcloud is then displayed in the window for the user to see.

- Summarize Article



While this data is very promising, it is to note that this doesn't 100% reflect into the real world as accurately as misinformation is a constantly evolving metric. Also the method of using a TF-IDF can put bias onto certain words that appear in fake articles even if the word may not be a true indicator of something being fake (such as the presence of an image or quote). This contributed to the changed approach of saying this model can find if an article is fake or not to an approach of being able to identify potentially misinforming articles.

Functionalities

Functionalities

Fake News Detection

- Displays prediction (real or fake)
- Shows top words contributing toward a fake classification
- Shows top words contributing toward a real classification
- Displays the full article text

Misinformation Keyword Analysis

- Lists the top 50 fake-influencing words

- Lists the top 50 real-influencing words
- Shows weight, average TF-IDF, max TF-IDF, IDF, and document frequency

Word Cloud

- Generates a word cloud visualization from the article text

Article Summarizer

- Creates a concise summary using Newspaper3k’s NLP functions

All features are accessible from the navigation bar.

Execution Results

During testing with real-world news websites such as AP News, Reuters, BBC, and Politico, we observed a significant limitation of the model:

The classifier frequently labeled legitimate articles as fake.

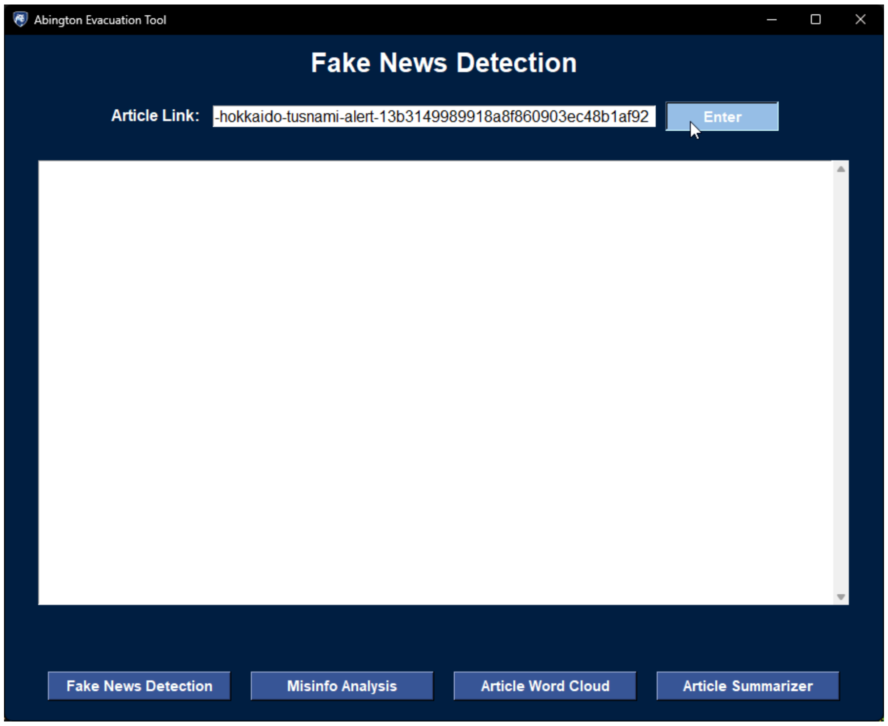
Examples include:

- AP News articles incorrectly predicted as fake
- Reuters articles incorrectly predicted as fake
- BBC articles resulting in low-confidence predictions or “fake” labels

This indicates that the trained model has **dataset bias.**
The dataset it was trained on contains language patterns that differ from real journalism, leading the model to misinterpret normal reporting language as signs of misinformation.

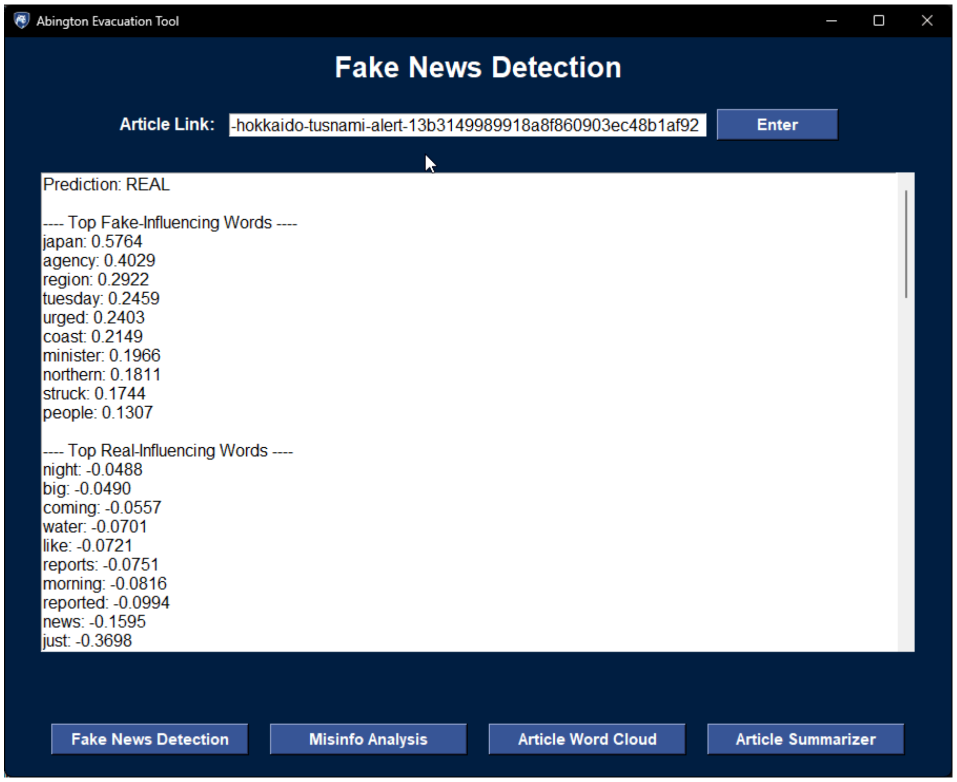
The analysis tools (word cloud, keyword tables, summarizer) still function correctly, but prediction accuracy on real news websites is not reliable.

- Fake News Detection
    1. Real
        1. Input a link to an article into the entry box and click the enter button



Article Link: -hokkaido-tusnami-alert-13b3149989918a8f860903ec48b1af92

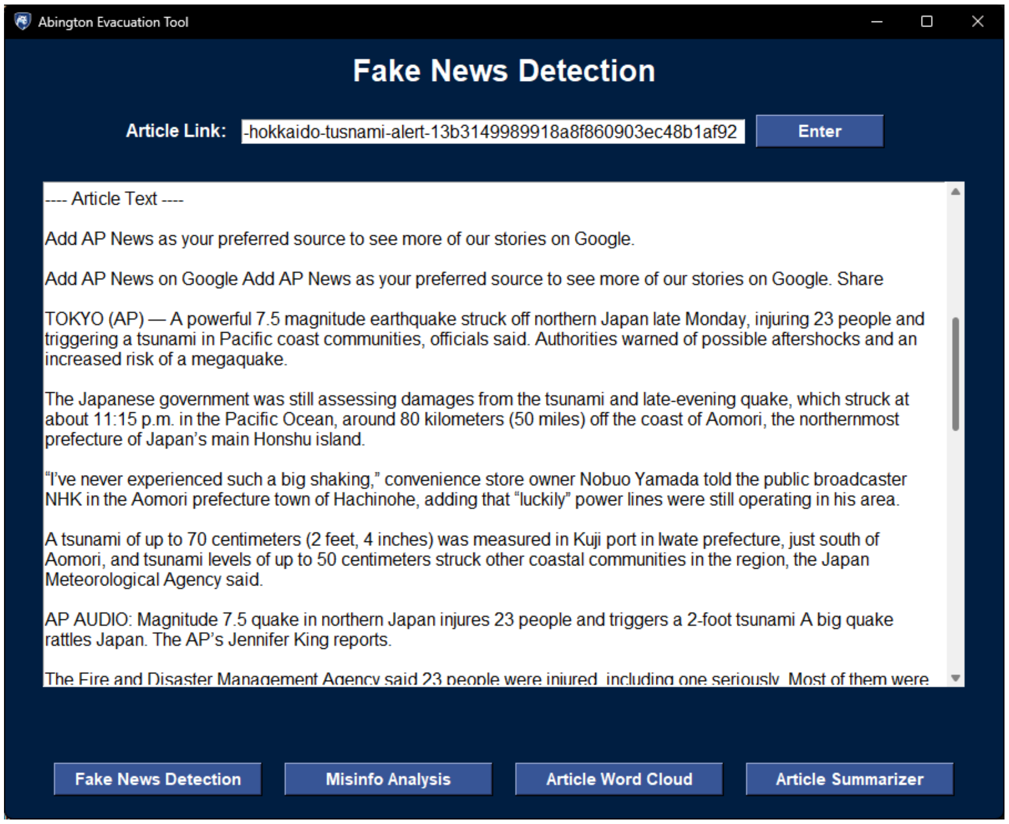
2. The results from the model are printed in the textbox underneath



```
Prediction: REAL

---- Top Fake-Influencing Words ----
japan: 0.5764
agency: 0.4029
region: 0.2922
tuesday: 0.2459
urged: 0.2403
coast: 0.2149
minister: 0.1966
northern: 0.1811
struck: 0.1744
people: 0.1307

---- Top Real-Influencing Words ----
night: -0.0488
big: -0.0490
coming: -0.0557
water: -0.0701
like: -0.0721
reports: -0.0751
morning: -0.0816
reported: -0.0994
news: -0.1595
just: -0.3698
```



---- Article Text ----

Add AP News as your preferred source to see more of our stories on Google.

Add AP News on Google Add AP News as your preferred source to see more of our stories on Google. Share

TOKYO (AP) — A powerful 7.5 magnitude earthquake struck off northern Japan late Monday, injuring 23 people and triggering a tsunami in Pacific coast communities, officials said. Authorities warned of possible aftershocks and an increased risk of a megaquake.

The Japanese government was still assessing damages from the tsunami and late-evening quake, which struck at about 11:15 p.m. in the Pacific Ocean, around 80 kilometers (50 miles) off the coast of Aomori, the northernmost prefecture of Japan's main Honshu island.

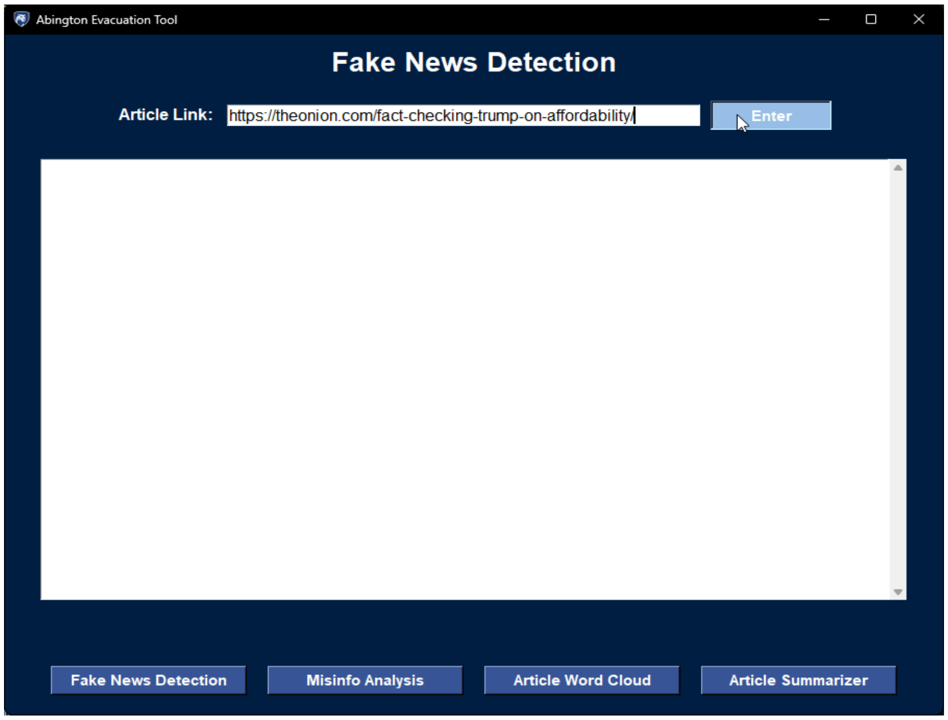
"I've never experienced such a big shaking," convenience store owner Nobuo Yamada told the public broadcaster NHK in the Aomori prefecture town of Hachinohe, adding that "luckily" power lines were still operating in his area.

A tsunami of up to 70 centimeters (2 feet, 4 inches) was measured in Kuji port in Iwate prefecture, just south of Aomori, and tsunami levels of up to 50 centimeters struck other coastal communities in the region, the Japan Meteorological Agency said.

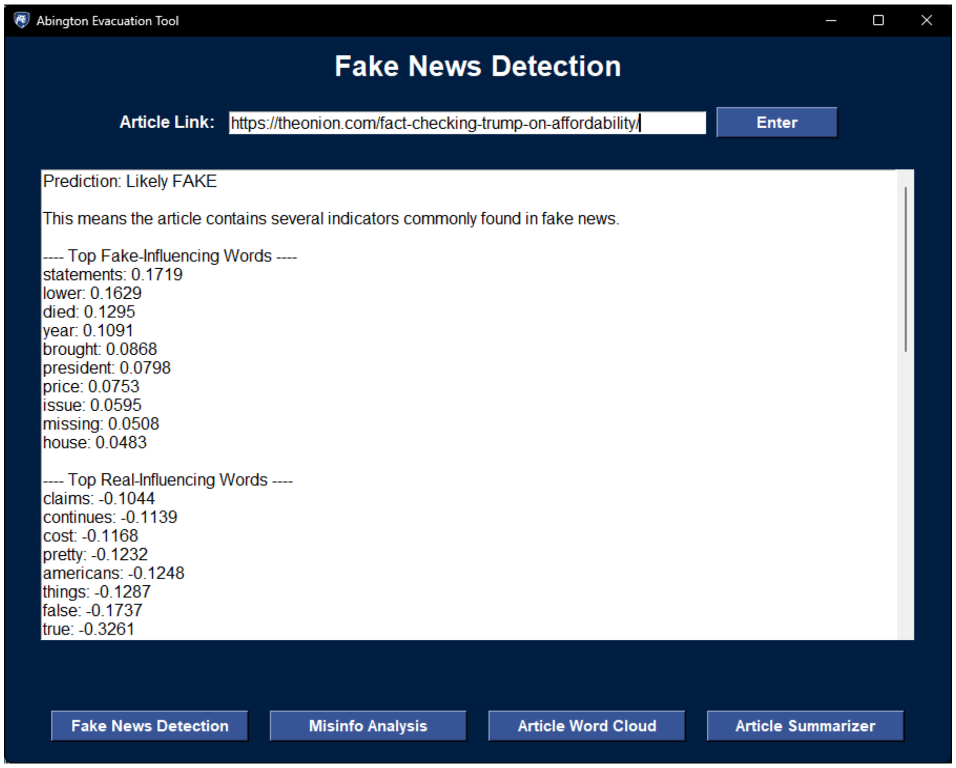
AP AUDIO: Magnitude 7.5 quake in northern Japan injures 23 people and triggers a 2-foot tsunami A big quake rattles Japan. The AP's Jennifer King reports.

The Fire and Disaster Management Agency said 23 people were injured, including one seriously. Most of them were

2. Fake
    1. Input a link to an article into the entry box and click the enter button



    2. The results from the model are printed in the textbox underneath

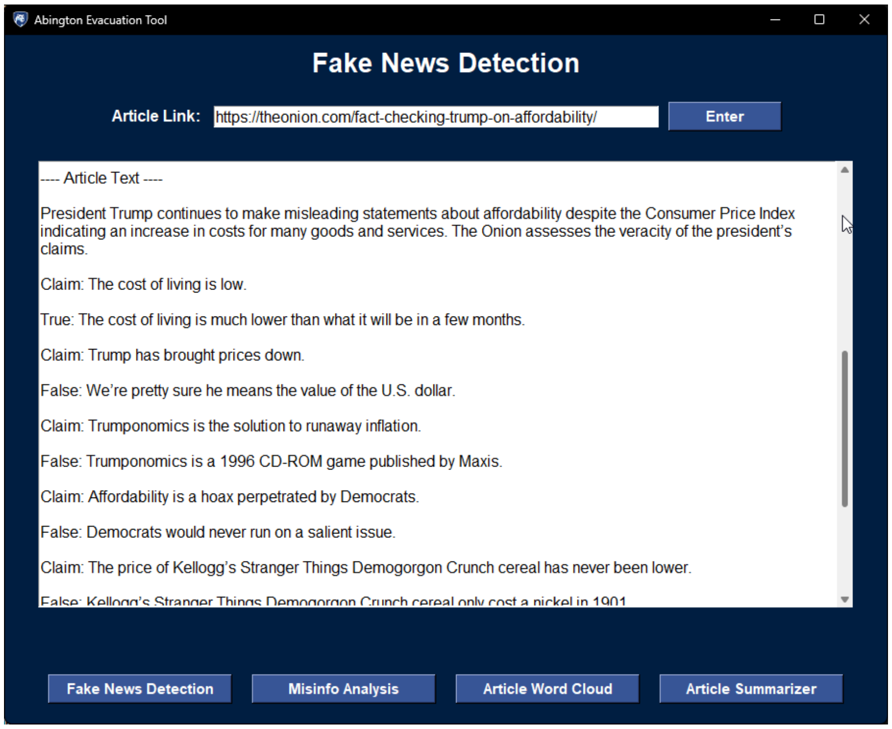


```
Prediction: Likely FAKE

This means the article contains several indicators commonly found in fake news.

---- Top Fake-Influencing Words ----
statements: 0.1719
lower: 0.1629
died: 0.1295
year: 0.1091
brought: 0.0868
president: 0.0798
price: 0.0753
issue: 0.0595
missing: 0.0508
house: 0.0483

---- Top Real-Influencing Words ----
claims: -0.1044
continues: -0.1139
cost: -0.1168
pretty: -0.1232
americans: -0.1248
things: -0.1287
false: -0.1737
true: -0.3261
```



Article Link: https://theonion.com/fact-checking-trump-on-affordability/

---- Article Text ----

President Trump continues to make misleading statements about affordability despite the Consumer Price Index indicating an increase in costs for many goods and services. The Onion assesses the veracity of the president's claims.

Claim: The cost of living is low.

True: The cost of living is much lower than what it will be in a few months.

Claim: Trump has brought prices down.

False: We're pretty sure he means the value of the U.S. dollar.

Claim: Trumponomics is the solution to runaway inflation.

False: Trumponomics is a 1996 CD-ROM game published by Maxis.

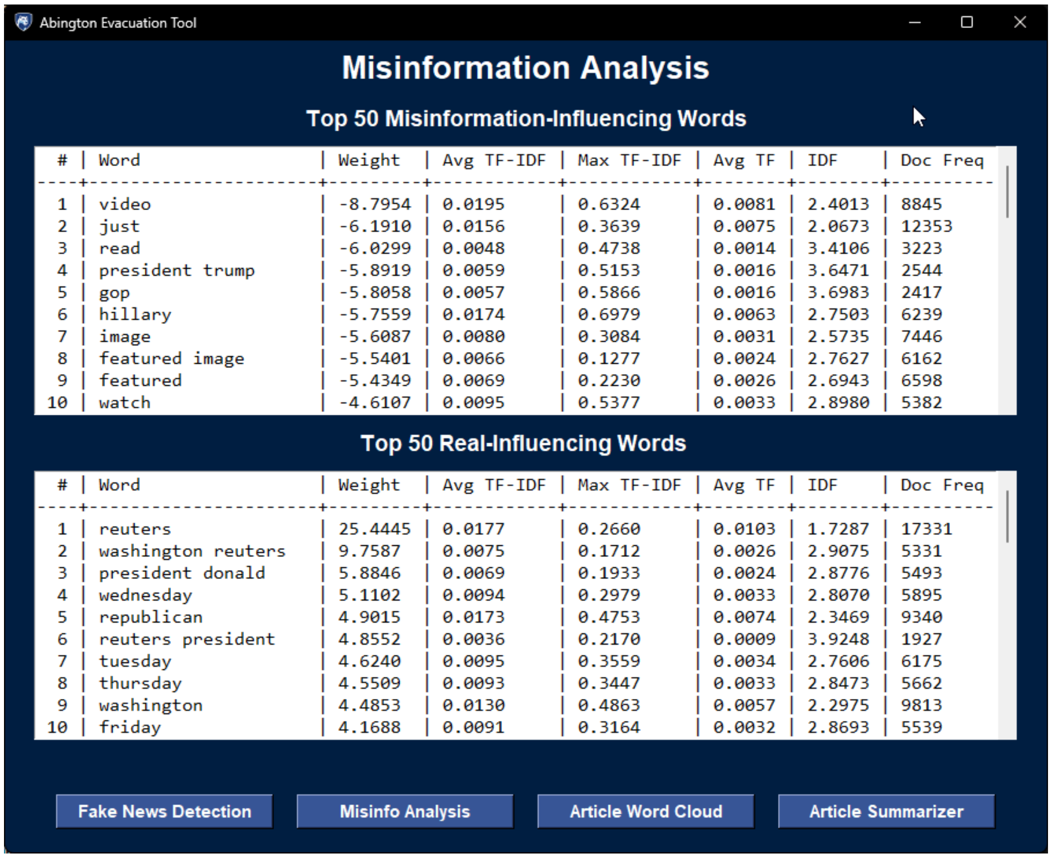
Claim: Affordability is a hoax perpetrated by Democrats.

False: Democrats would never run on a salient issue.

Claim: The price of Kellogg's Stranger Things Demogorgon Crunch cereal has never been lower.

False: Kellogg's Stranger Things Demogorgon Crunch cereal only cost a nickel in 1901

- Misinfo Analysis



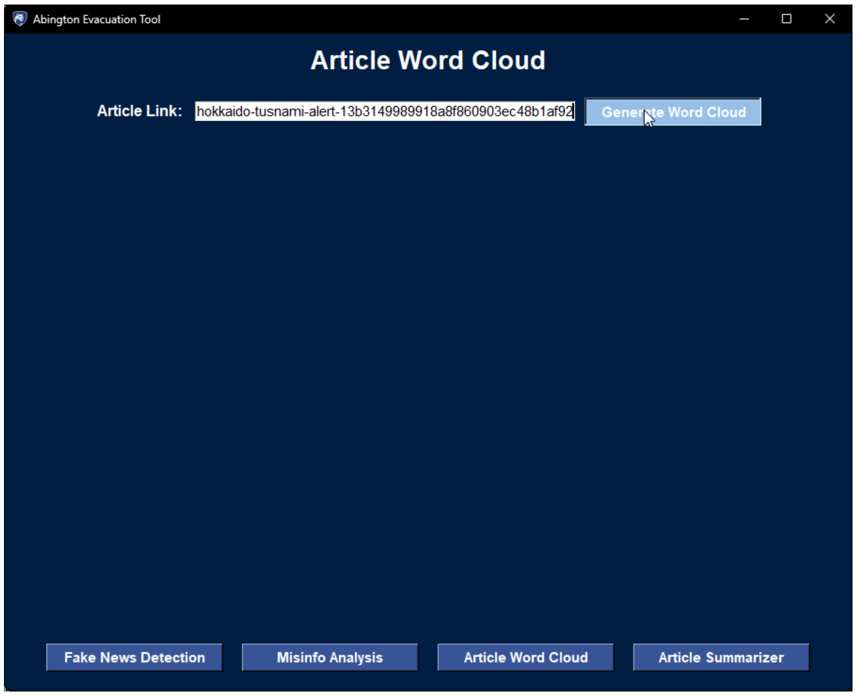
Top 50 Misinformation-Influencing Words

```
 #  | Word                       | Weight  | Avg TF-IDF | Max TF-IDF | Avg TF | IDF    | Doc Freq
----+----------------------------+---------+------------+------------+--------+--------+---------
  1 | video                      | -8.7954 | 0.0195     | 0.6324     | 0.0081 | 2.4013 | 8845
  2 | just                       | -6.1910 | 0.0156     | 0.3639     | 0.0075 | 2.0673 | 12353
  3 | read                       | -6.0299 | 0.0048     | 0.4738     | 0.0014 | 3.4106 | 3223
  4 | president trump            | -5.8919 | 0.0059     | 0.5153     | 0.0016 | 3.6471 | 2544
  5 | gop                        | -5.8058 | 0.0057     | 0.5866     | 0.0016 | 3.6983 | 2417
  6 | hillary                    | -5.7559 | 0.0174     | 0.6979     | 0.0063 | 2.7503 | 6239
  7 | image                      | -5.6087 | 0.0080     | 0.3084     | 0.0031 | 2.5735 | 7446
  8 | featured image             | -5.5401 | 0.0066     | 0.1277     | 0.0024 | 2.7627 | 6162
  9 | featured                   | -5.4349 | 0.0069     | 0.2230     | 0.0026 | 2.6943 | 6598
 10 | watch                      | -4.6107 | 0.0095     | 0.5377     | 0.0033 | 2.8980 | 5382
```

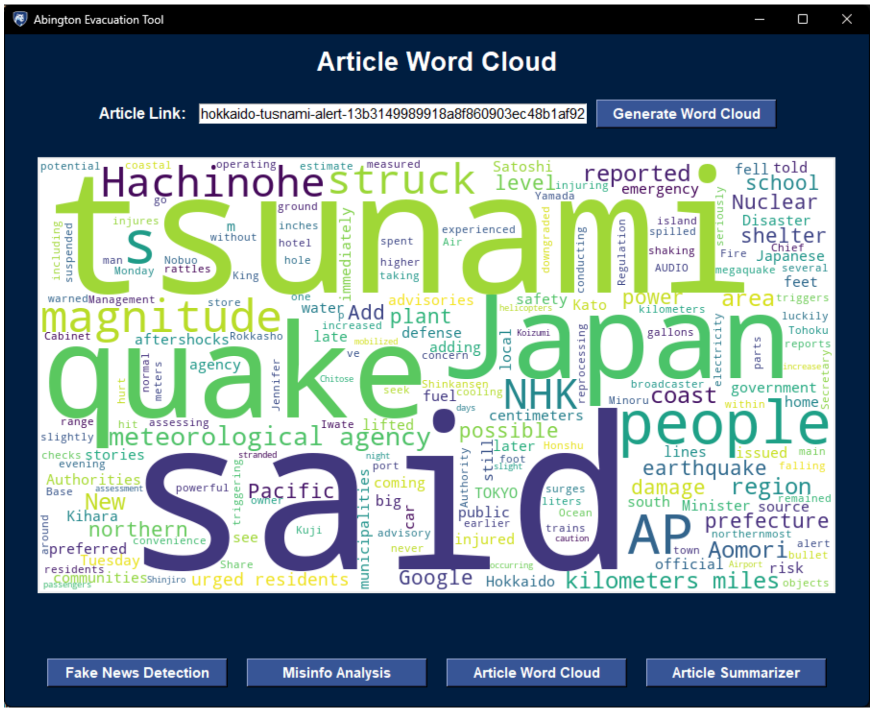
Top 50 Real-Influencing Words

```
 #  | Word                       | Weight  | Avg TF-IDF | Max TF-IDF | Avg TF | IDF    | Doc Freq
----+----------------------------+---------+------------+------------+--------+--------+---------
  1 | reuters                    | 25.4445 | 0.0177     | 0.2660     | 0.0103 | 1.7287 | 17331
  2 | washington reuters         | 9.7587  | 0.0075     | 0.1712     | 0.0026 | 2.9075 | 5331
  3 | president donald           | 5.8846  | 0.0069     | 0.1933     | 0.0024 | 2.8776 | 5493
  4 | wednesday                  | 5.1102  | 0.0094     | 0.2979     | 0.0033 | 2.8070 | 5895
  5 | republican                 | 4.9015  | 0.0173     | 0.4753     | 0.0074 | 2.3469 | 9340
  6 | reuters president          | 4.8552  | 0.0036     | 0.2170     | 0.0009 | 3.9248 | 1927
  7 | tuesday                    | 4.6240  | 0.0095     | 0.3559     | 0.0034 | 2.7606 | 6175
  8 | thursday                   | 4.5509  | 0.0093     | 0.3447     | 0.0033 | 2.8473 | 5662
  9 | washington                 | 4.4853  | 0.0130     | 0.4863     | 0.0057 | 2.2975 | 9813
 10 | friday                     | 4.1688  | 0.0091     | 0.3164     | 0.0032 | 2.8693 | 5539
```

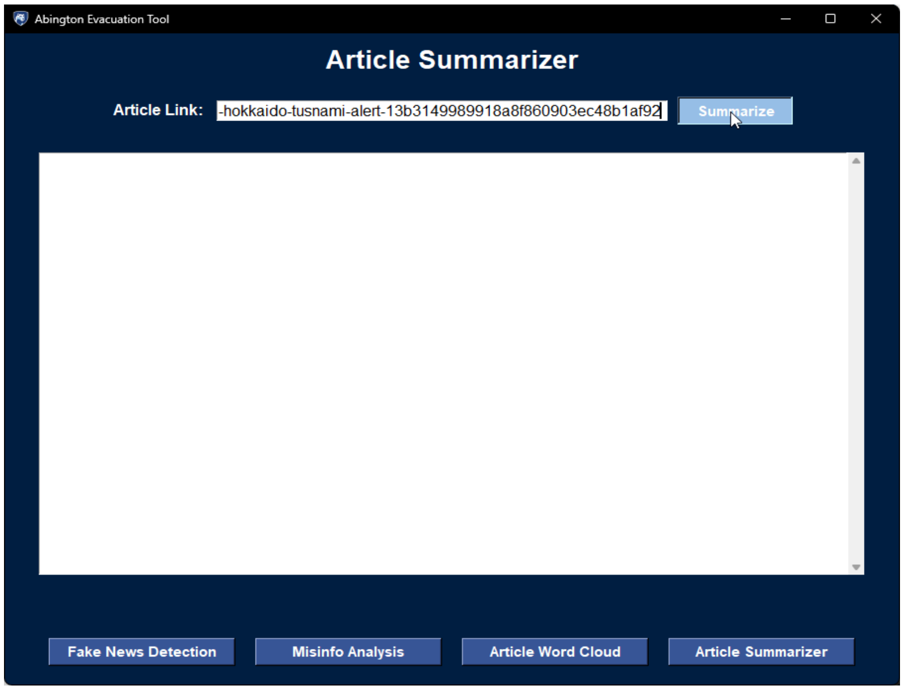
- Article Word Cloud
  1. Input a link to an article into the entry box and click the "Generate Word Cloud" button



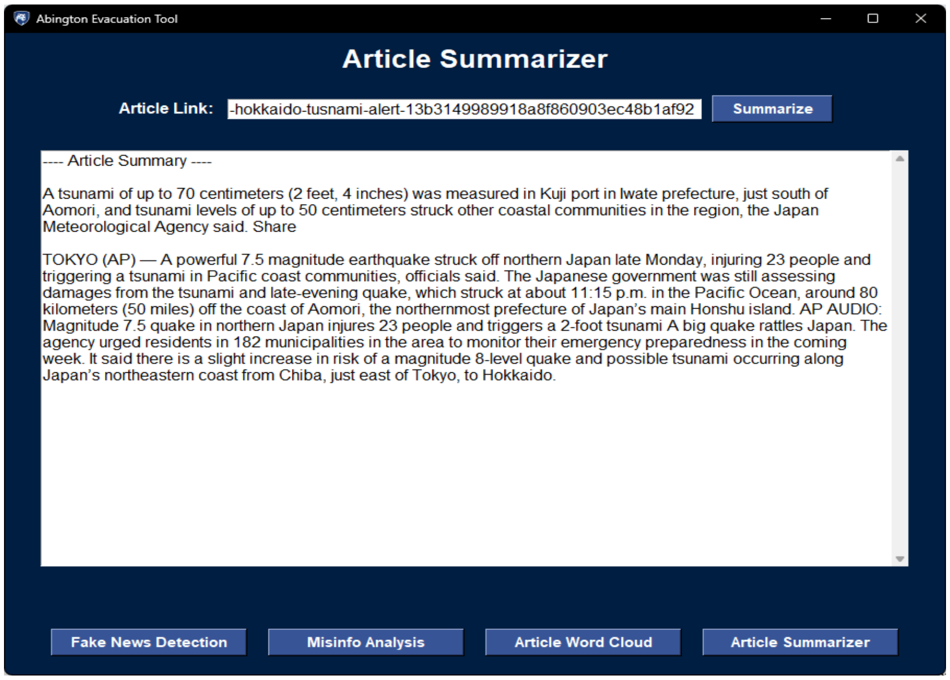
  2. Resulting word cloud is displayed



- Article Summarizer
    1. Input a link to an article into the entry box and click the “Summarize” button



    2. A summary of the article proportional to its size is then output in the textbox



---- Article Summary ----

A tsunami of up to 70 centimeters (2 feet, 4 inches) was measured in Kuji port in Iwate prefecture, just south of Aomori, and tsunami levels of up to 50 centimeters struck other coastal communities in the region, the Japan Meteorological Agency said. Share

TOKYO (AP) — A powerful 7.5 magnitude earthquake struck off northern Japan late Monday, injuring 23 people and triggering a tsunami in Pacific coast communities, officials said. The Japanese government was still assessing damages from the tsunami and late-evening quake, which struck at about 11:15 p.m. in the Pacific Ocean, around 80 kilometers (50 miles) off the coast of Aomori, the northernmost prefecture of Japan’s main Honshu island. AP AUDIO: Magnitude 7.5 quake in northern Japan injures 23 people and triggers a 2-foot tsunami A big quake rattles Japan. The agency urged residents in 182 municipalities in the area to monitor their emergency preparedness in the coming week. It said there is a slight increase in risk of a magnitude 8-level quake and possible tsunami occurring along Japan’s northeastern coast from Chiba, just east of Tokyo, to Hokkaido.

Challenges

Several challenges appeared throughout the development of this project, both on the machine-learning side and the GUI integration side. The most significant challenges were the following:

1. Dataset Bias and Poor Real-World Performance

Although the model achieved high accuracy (around 99%) during training, it struggled when classifying real news articles from reliable sources such as AP News, BBC, and Reuters.
Many of these articles were incorrectly labeled as fake.

This revealed a major issue:
The dataset did not accurately represent real journalistic writing, leading the model to learn patterns that do not generalize to real-world text.
This challenge highlighted how misleading accuracy scores can be when the training data is not diverse or balanced.

2. Difficulty Extracting Article Text From Websites

Using the newspaper library worked for some sites but failed for many others due to:

- websites blocking automated scraping
- missing HTML structures
- inconsistent article formatting
- 403/404 access errors
- SSL certificate issues

Conclusions

While the project successfully integrates machine learning, NLP, and GUI features into a unified tool, the results show a clear limitation:

The model performs well on the dataset it was trained on but poorly on real-world news articles.

This is likely caused by:

- **Dataset bias:** Many fake news datasets oversimplify writing style, causing the model to associate normal journalistic words (such as “president,” “report,” “statement,” “agency,” etc.) with fake news. The dataset also contains a bias towards reuters which is considered a reliable source. This means that the term Reuters is seen as a very large indicator of a true article compared to other reliable news sources.

- **Overfitting:** The model may have learned patterns that only apply to the training data, not real-world examples.
- **Differences in writing style:** Professional journalism uses structured wording the dataset may not represent.

Key Takeaways

- The classification accuracy reported during training (around 99%) does not transfer to real-world performance.
- The tool’s analysis components (keyword weights, word cloud, summarization) work correctly and provide helpful insights.
- Improving real-world performance would require:
  - A better dataset with more balanced examples
  - Additional preprocessing
  - Possibly using deep learning instead of a simple SVM
  - Incorporating more sophisticated feature extraction methods

Overall Conclusion

The project is successful as a demonstration of machine learning pipeline integration, GUI development, and text analysis tools, but **the fake news prediction model is not reliable for real online articles without further refinement**.