TRUTHLENS

CM3070 Computer Science Final Project - Preliminary Report

Abstract

TruthLens, based on the Fake News Detection template, is a Python based two-stage misinformation classifier. The application provides users with a lens through which they can identify misinformation along with an explanation of why it has been tagged that way.

Hazel Reitz – Student number 190125246

30/12/2024

Introduction

Misinformation and disinformation have been persistent features of human communication, from propaganda coins in Roman times[1] to modern efforts to influence elections and referendums[2]. Recent advancements such as large language models (LLMs), generative AI, and the widespread adoption of social media have exponentially increased the production and dissemination of misleading content. This poses both technical and cognitive challenges. For example, 75% of Americans overestimate their ability to distinguish between legitimate and fake news headlines[3]. This highlights the urgent need for tools that not only detect misinformation, but also help users critically evaluate its content through clear explanations.

The spread of misinformation has significant consequences, from undermining public trust in mainstream media[5] to triggering conflicts between individuals[4]. Addressing this issue is essential to reducing harm from false narratives and maintaining informed democracies. Diverse user groups, such as journalists, educators, social media platforms, and individual consumers, face unique challenges in navigating misinformation. Journalists must verify breaking news quickly, educators need tools to teach media literacy, social media platforms must address misinformation at scale, and individuals often lack the means to assess credibility. This project addresses these challenges by introducing categorisation and explainability into automated misinformation detection, empowering users to better evaluate the content they encounter.

To effectively combat misinformation, it is crucial to understand its diverse forms. The term “fake news” has evolved to encompass a wide range of misleading content and has become weaponised in political discourse, as noted by Vosoughi et al.[6]. This report instead uses the term “misinformation”, which avoids political connotations and better represents the spectrum of misleading content. Building on Molina et al.’s taxonomy of seven misinformation types - false news, polarised content, satire, misreporting, commentary, persuasive information, and citizen journalism[7] - this project employs a nuanced framework to detect and categorise misinformation accurately. These categories are outlined in the table below.

TABLE 1 - MOLINA ET AL. MISINFORMATION TAXONOMY

Misinformation Type	Characteristics	Examples
Fabricated content	Completely false content created with the intent to deceive.	Fake reports of events that never occurred; entirely false claims about public figures
Polarised content	True events or facts presented selectively to promote a biased narrative, often omitting critical context.	Partisan news articles highlighting one side of a political argument while ignoring counterpoints.
Satire	Content intended to entertain or provoke thought through humour, exaggeration, or irony. Often misunderstood.	Satirical articles from outlets like “The Onion” being shared as if they are factual news.
Misreporting	Incorrect information shared unintentionally, often due to errors or lack of verification.	A news outlet incorrectly reporting election results due to early or inaccurate data.
Commentary	Opinion-based content reflecting the writer’s interpretation or viewpoint, often lacking factual grounding.	Editorials or blogs expressing subjective opinions without substantial evidence.
Persuasive information	Content designed to persuade or influence the audience, often including marketing and propaganda.	Politically motivated propaganda campaigns, advertisements disguised as objective news articles.
Citizen journalism	User-generated content that may lack professional journalistic standards, leading to error or bias.	Social media posts about breaking news that spread unverified or incorrect details.

Traditional misinformation detection methods rely on binary classification - labeling content as true or false - but this approach oversimplifies the problem. Misleading content often exists on a spectrum, requiring more granular categorisation to inform users. To foster media literacy and critical engagement, users need to be able to distinguish whether a piece of content is entirely fabricated (false news) or presented selectively to promote bias. By moving beyond binary classification, this project aims to improve detection systems and contribute to a better understanding of misinformation.

While categorisation and explainability are key components of misinformation detection, speed and scalability are equally vital in addressing its rapid spread. Real-time identification of misinformation is necessary to mitigate its rapid spread. Vosoughi et al.[6] found that fake news stories spread farther and faster than true ones, driven by their novelty and emotionally charged framing. Manual fact-checking, while often highly accurate, is time-consuming, requires domain expertise, and is unable to keep pace with the volume of content especially during high-volume events such as elections. Balancing these demands, this project employs natural language processing (NLP) and machine learning (ML) techniques to enable scalable and efficient misinformation detection.

By integrating explainability and multi-class categorisation, this project bridges a critical gap in misinformation detection. It combines technical innovation with a user-centric design to move beyond traditional methods, aligning detection strategies with the nuanced realities of misinformation while fostering critical media literacy.

Literature Review

The proliferation of misinformation presents significant challenges to society, making its detection a critical research area. This review evaluates current methodologies, datasets, industry tools, and the challenges associated with misinformation detection. It also highlights the gaps and insights that inform the design of this project.

Methodologies

Approaches to misinformation detection are broadly divided into manual and automated methods. Manual methods, such as expert analysis and crowdsourced verification, yield high accuracy but are resource-intensive and infeasible for large-scale applications (Tajrian et al.[8]). Automated methods, such as those this project employs, provide scalability and adaptability.

Traditional Machine Learning Approaches

Traditional machine learning models like Logistic Regression (LR), Random Forest (RF), and Support Vector Machines (SVM), have demonstrated strong performance in binary classification tasks. These models rely on feature engineering to extract characteristics like word frequencies and sentiment, which inform predictions.

- Adeyiga et at.[9] demonstrated that logistic regression achieved an impressive accuracy rate of over 97% in binary classification tasks. However, their study revealed that these models struggle with nuanced content requiring contextual understanding such as satire and polarised reporting.

- Sudhakar and Kaliyamurthie[10] highlighted the interpretability of these models but emphasised their limitations in evolving misinformation scenarios due to reliance on static feature sets.

Critical insight: While traditional ML methods are effective for binary classification, their reliance on feature engineering makes them unsuitable for nuanced multi-class classification. Deep learning approaches are better suited for this.

Deep Learning Approaches

Deep learning methods, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformer-based architectures (e.g., BERT and RoBERTa), have revolutionised misinformation detection. These methods eliminate the need for manual feature engineering by learning patterns and contextual relationships directly from data.

- Li et al.[11] found that BERT significantly outperformed traditional models, achieving an F1 score of 0.92 in fake news classification. Its ability to capture long-term dependencies makes it particularly suited for nuanced tasks.
- Ali et al.[12] demonstrated the robustness of RNNs for handling sequential data, although they noted challenges in processing long-form text efficiently.
- Transformer-based models, like BERT and RoBERTa, are considered the gold standard due to their attention mechanisms, which weigh input text elements based on relevance. However, their computational costs are substantial.

Critical insight: Transformer-based architectures are well-suited for fine-grained multi-class classification. While deep learning methods offer unparalleled accuracy, their opaque decision-making processes pose challenges for explainability. This project aims to incorporate Explainable AI (XAI) techniques to address this limitation, ensuring transparency in predications.

Binary vs Multi-Class Classification

The majority of existing work focuses on binary classification (e.g. fake vs. real). While this approach simplifies implementation and evaluation, it oversimplifies the problem, as misinformation often exists on a spectrum. Multi-class classification, which distinguishes between types of misinformation such as satire, polarised content, and hoaxes, provides richer insights. This project will utilise both binary and multi-class classification.

- Li et al.[11] showed that BERT-based models trained on multi-class datasets like LIAR struggle with imbalanced classes, particularly underperforming on minority categories.
- Advanced strategies such as data augmentation and ensemble methods can mitigate these challenges, as highlighted by Adeyiga et al.[9].

Datasets

Datasets play a crucial role in automated fake news detection by providing the foundational data needed to train, validate, and benchmark detection algorithms. There are a number of prominent datasets available, most of which have a significant body of research and benchmarks. This section reviews three significant datasets—LIAR, Fake News Corpus, and ISOT—and provides a comparative analysis to highlight their applicability to different tasks in misinformation detection.

LIAR

LIAR[13] is a publicly available dataset consisting of over 12,800 manually labeled short statements collected over a decade from the Politifact website. Each statement in the dataset is categorized into one of six classes: True, Mostly True, Half True, Barely True, False, and Pants on Fire (a label meaning very false or completely made up). The dataset also includes rich metadata on context, subject, speaker affiliations and more.

One strength of this dataset is that it has been annotated by experts, so we know the labels are high quality. Additionally, the six classes allow for more nuanced detection than a binary approach. However, as LIAR is a politically focused dataset, its usefulness may be limited for fake news detection in other domains. Another issue to note is that it is a static dataset ending in 2017 and thus does not account for evolving misinformation tactics. The classes are also not balanced, with the "Pants on Fire" class being particularly underrepresented, which can introduce bias during model training.

Fake News Corpus

Fake News Corpus[14] is a large-scale dataset containing 9.4 million articles sourced from a diverse range of websites. Articles are grouped by the domains that they come from, with labels such as reliable, bias, and conspiracy. The sheer scale of the dataset is one of its major strengths. The domain-level labelling allows models to get a good understanding of the overall reliability of websites, aiding in pattern recognition across certain websites.

However, the Fake News Corpus has some noticeable limitations. Articles are not individually labeled; instead, labels are applied at the domain level and apply to any article that originated there. This can lead to inaccuracies where a website publishes a mix of true and false content. Similar to the LIAR dataset, the classes in this dataset are imbalanced, with reliable sources far outnumbering biased or conspiratorial ones.

ISOT

The ISOT Fake News Dataset[15] contains 44,898 articles, split into two balanced categories: 21,417 fake news articles and 23,481 real news articles. The real news articles are sourced from Reuters, while the fake news articles originate from websites flagged as unreliable by Politifact and Wikipedia. Unlike LIAR and Fake News Corpus, the ISOT dataset offers a balanced class distribution, reducing the risk of bias in binary classification tasks.

A major strength of the ISOT dataset is its longer text samples, which provide richer context for natural language processing models and make it particularly suitable for tasks that require deep semantic understanding. However, the narrow topical focus of the dataset is a limitation, with much of the content being political or world news. Furthermore, the dataset has binary labels, making it less suitable for advanced tasks, like explainability or multi-class classification.

Comparative Analysis

While each dataset has its strengths, their limitations highlight key challenges in fake news detection. The LIAR dataset excels in fine-grained classification but is domain-specific and suffers from class imbalance. The Fake News Corpus provides unparalleled scale and domain-level insights but lacks individual article labeling. The ISOT dataset, with its balanced classes and rich text samples, is ideal for binary classification but limited in its topical diversity and metadata richness.

TABLE 2 - DATASET COMPARISON

Dataset	Size	Labels	Domain	Strengths	Weaknesses
LIAR	12,836	6 classes (True, Half True, etc.)	Politics	Expert annotations, rich metadata	Static, imbalanced classes, narrow topical focus
Fake News Corpus	9.4 million	Domain based (Reliable, Conspiracy etc.)	Diverse websites	Large scale, domain level patterns	No article-level labelling
ISOT	44,898	2 classes (Fake, Real)	Politics / World News	Balanced classes, longer text	Static, Narrow topical focus, lacks metadata

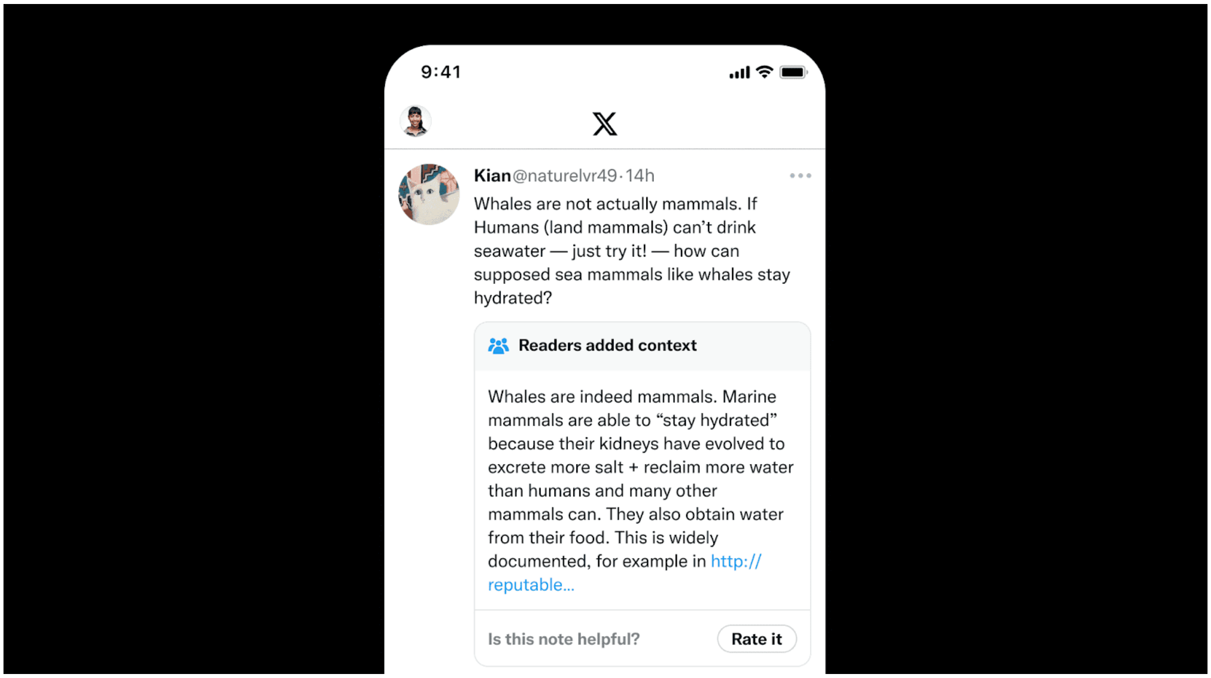
Industry tools

Industry tools demonstrate practical applications of misinformation detection. These tools leverage diverse methodologies, including crowd-sourced verification, algorithmic analysis, and AI-driven evaluations. This section explores three prominent tools -

X/Twitter's misinformation measures, Google Fact Check Explorer, and Facticity.ai - and evaluates their effectiveness, scalability, and accessibility.

X/Twitter

X, formerly known as Twitter, has faced scrutiny for the volume of misinformation on its platform[16]. A variety of tools to address misinformation, utilising both algorithmic and crowd-sources approaches, have been implemented[17]. These include user reporting mechanisms, "community notes" that allow users to add contextual explanations to posts, and curated collections of trusted information under "X Moments". Additionally, the platform can limit the visibility of, or remove, flagged content.



Kian@naturelvr49·14h

Whales are not actually mammals. If Humans (land mammals) can't drink seawater — just try it! — how can supposed sea mammals like whales stay hydrated?

Readers added context

Whales are indeed mammals. Marine mammals are able to "stay hydrated" because their kidneys have evolved to excrete more salt + reclaim more water than humans and many other mammals can. They also obtain water from their food. This is widely documented, for example in http://reputable...

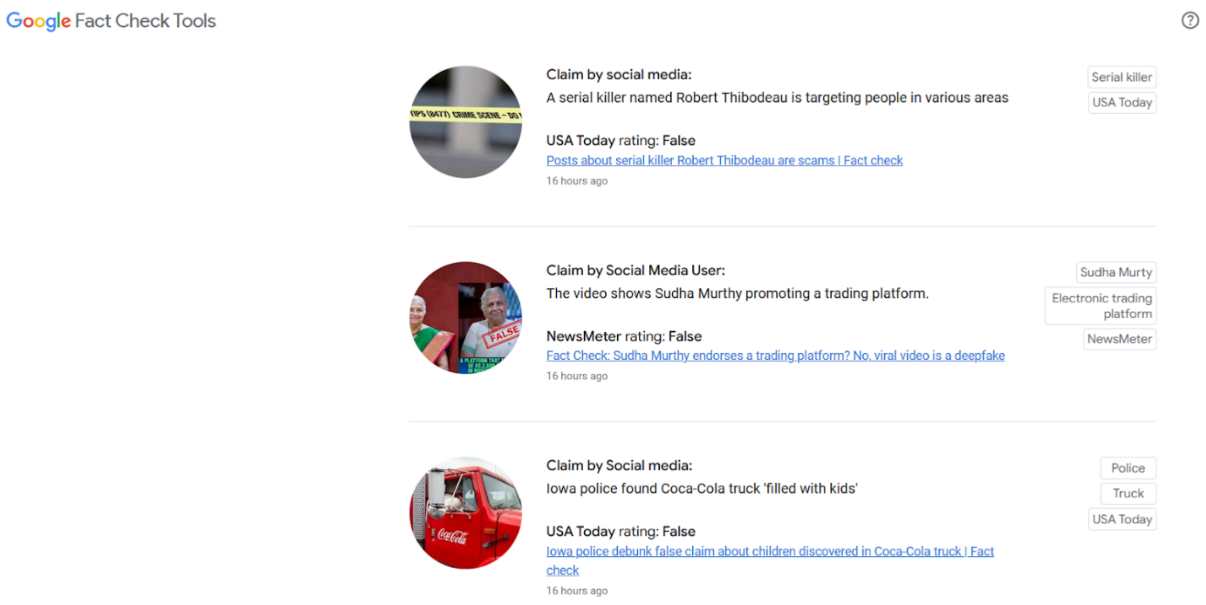
FIGURE 1 - COMMUNITY NOTE ON X

While these measures demonstrate proactive engagement, they are not without limitations. Scalability remains a challenge, as community notes or crowdsourced knowledge are not universally applied across posts. Furthermore, smaller platforms cannot replicate this approach due to their limited user bases. The reliance on voting mechanisms for community notes also raises concerns about potential gamification by malicious actors seeking to promote specific narratives.

Google Fact Check Explorer

Google Fact Check Explorer aggregates fact-checks from reputable organisations worldwide, enabling users to quickly verify the credibility of news stories or claims[18].

By searching for a headline or claim, users can access a range of assessments and contextual information from sources such as Politifact, Snopes, and FactCheck.org.



Google Fact Check Tools

Claim by social media:
A serial killer named Robert Thibodeau is targeting people in various areas

USA Today rating: False
Posts about serial killer Robert Thibodeau are scams | Fact check
16 hours ago

Claim by Social Media User:
The video shows Sudha Murthy promoting a trading platform.

NewsMeter rating: False
Fact Check: Sudha Murthy endorses a trading platform? No, viral video is a deepfake
16 hours ago

Claim by Social media:
Iowa police found Coca-Cola truck 'filled with kids'

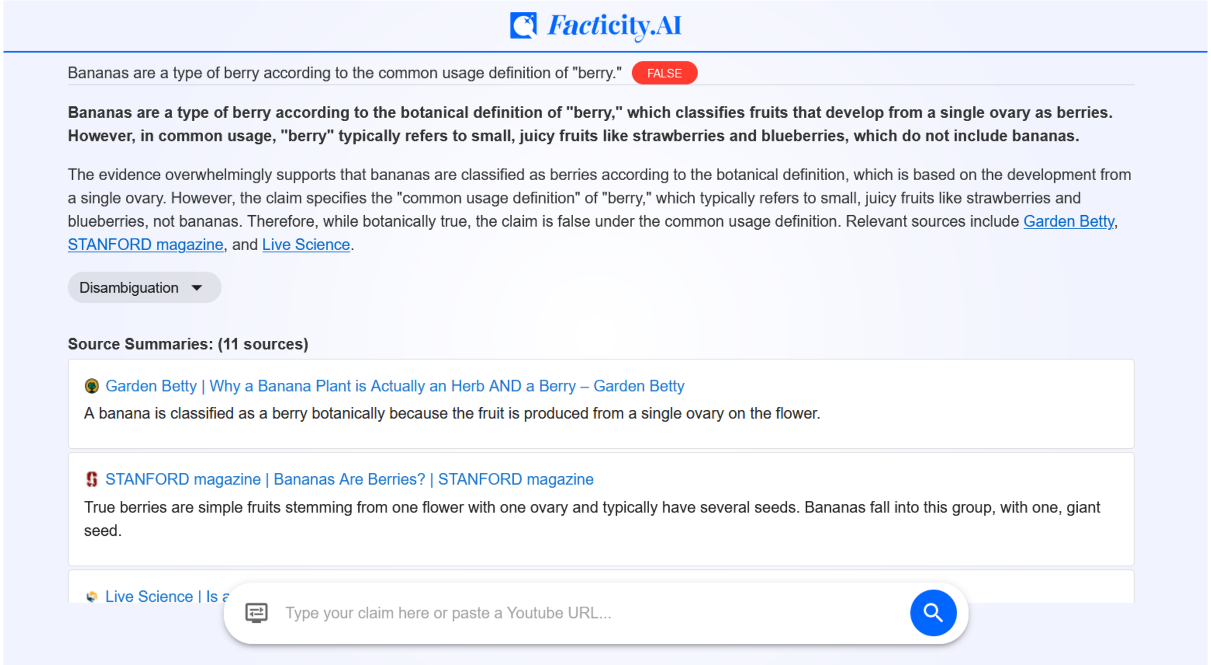
USA Today rating: False
Iowa police debunk false claim about children discovered in Coca-Cola truck | Fact check
16 hours ago

FIGURE 2 - GOOGLE FACT CHECK TOOLS

A significant advantage of Fact Check Explorer is its integration of diverse sources, offering a comprehensive overview of a claim’s credibility. However, it relies heavily on manual fact-checking processes, which limit its speed and scalability, particularly during high-volume events. Additionally, the quality of results depends on the presence of fact-checks for specific claims, meaning that the tool may lack coverage for newer or less widely reported topics.

Facticity.ai

Facticity.ai is a commercial AI-powered tool designed to assess the credibility of online content[19]. It leverages natural language processing and machine learning algorithms to analyze articles, blogs, and other digital media. It provides users with a trustworthiness score based on linguistic patterns, source reliability, and content metadata. It also offers detailed feedback on why a specific piece of content might be misleading, making it particularly useful for educators, journalists, and researchers aiming to combat misinformation.



Bananas are a type of berry according to the common usage definition of "berry." FALSE

Bananas are a type of berry according to the botanical definition of "berry," which classifies fruits that develop from a single ovary as berries. However, in common usage, "berry" typically refers to small, juicy fruits like strawberries and blueberries, which do not include bananas.

The evidence overwhelmingly supports that bananas are classified as berries according to the botanical definition, which is based on the development from a single ovary. However, the claim specifies the "common usage definition" of "berry," which typically refers to small, juicy fruits like strawberries and blueberries, not bananas. Therefore, while botanically true, the claim is false under the common usage definition. Relevant sources include Garden Betty, STANFORD magazine, and Live Science.

Source Summaries: (11 sources)

Garden Betty | Why a Banana Plant is Actually an Herb AND a Berry – Garden Betty

A banana is classified as a berry botanically because the fruit is produced from a single ovary on the flower.

STANFORD magazine | Bananas Are Berries? | STANFORD magazine

True berries are simple fruits stemming from one flower with one ovary and typically have several seeds. Bananas fall into this group, with one, giant seed.

Live Science | Is a

FIGURE 3 - FACTICITY.AI INTERFACE

A notable advantage of Facticity.ai is its explainability, as the tool highlights specific elements of a text that influenced its trustworthiness rating. However, one limitation is that its effectiveness depends on the robustness of its training data, which may not fully capture emerging misinformation trends or domain-specific nuances. Additionally, as a proprietary tool, it may not be accessible to all users without subscription or licensing fees.

Challenges and gaps

Despite advancements in language models, significant challenges remain in misinformation detection. This project addresses key challenges identified in the literature:

- **Class imbalance:** Many datasets are imbalanced, leading to biased predictions. This skews model training and evaluation. This project’s custom dataset will ensure balanced training data.
- **Explainability:** Many models function as “black boxes”, making it difficult to interpret why a particular piece of content was classified in a specific way. This project integrates Explainable AI (XAI) techniques to enhance user trust and understanding.
- **Evolving misinformation patterns:** Misinformation evolves rapidly in terms of form, targets, and platforms. This necessitates adaptive models capable of generalising across datasets and contexts. This project employs transfer learning to generalise across contexts.

Addressing these challenges requires strategies such as transfer learning for adaptability, data augmentation for balancing datasets, and the incorporation of explainable AI techniques to improve model transparency.

Design

Project Objectives

The primary objective of this project is to build a two-stage pipeline for misinformation classification:

1. Binary classification (Stage 1): Distinguish between real news and misinformation using the ISOT dataset. This ensures robust detection at the first stage, leveraging an established dataset.
2. Multi-class classification (Stage 2): Further classify content identified as misinformation into one of seven categories, based on Molina et al.'s taxonomy. A custom dataset will support this nuanced classification.

The scope of the project is limited to text-based, English language content, explicitly excluding images and videos. A user interface will also be developed, enabling users to input articles or URLs and receive classification results.

A secondary objective is to enhance the explainability of classification results, aiming to provide users with interpretable insights into why content was classified in a particular way.

The project aims for high accuracy and reliability, with measurable performance goals. Ethical considerations, including bias mitigation and responsible dataset usage, will guide the design and implementation of the pipeline.

Justification of design choices

The two stage design is justified by the domain's requirements:

- Binary classification efficiently filters out real news.
- Multi-class classification addresses the nuanced nature of misinformation, aligning with the goal of detailed categorisation.

- Explainability addresses the user need for trust in automated systems and supports ethical AI practices.

Data Preparation

Data preparation for the project will occur in two phases to align with the two-stage pipeline:

Phase 1: Real vs. Fake News Classification

The first stage leverages the ISOT [15] dataset, which contains almost 45,000 long-form articles evenly distributed between two classes: real and fake news. The dataset is well-suited for binary classification due to its balance and scale. Using an existing dataset eliminates the need for custom data collection for this phase and accelerates development.

Phase 1 preprocessing includes tokenisation, stop-word removal, and text normalisation. These steps ensure consistency and reduce noise in the input data, enabling efficient model training.

Phase 2: Misinformation Typology Classification

For the second stage, a new dataset will be created to classify misinformation into one of seven categories from the Molina taxonomy. Approximately 200 articles per label will be collected, resulting in a balanced dataset of 1,400 samples. Ethical considerations, such as compliance with scraping regulations and proper attribution of sources, will guide data collection efforts. Additionally, to ensure diversity and representativeness, multiple sources will be used for each category.

The dataset will be created using the following steps:

- **Data collection:** Articles will be scraped from publicly available sources, including news outlets, satire websites, and independent journalism platforms, chosen for their relevance to specific misinformation categories. Tools such as Python’s BeautifulSoup and Scrapy libraries will facilitate efficient data scraping.
- **Preprocessing:** The collected articles will be cleaned to remove advertisements, HTML tags, and non-text content. Additionally, the standard preprocessing tasks outlined in phase 1 will be completed.

- **Annotation:** Articles will be manually reviewed with clear guidelines to ensure accurate labeling.

Dataset split

For both phases, the final datasets will be split into training, validation, and test sets using an 80-10-10 ratio to support robust model evaluation.

Feature Extraction

Feature extraction is a critical step in transforming raw text into numerical representations suitable for machine learning models.

Phase 1: Real vs. Fake News Classification

For the first phase, feature extraction will focus on capturing key linguistic and textual characteristics that differentiate real news from misinformation. The following approaches will be employed:

- **TF-IDF** (Term Frequency- Inverse Document Frequency): Quantifies the importance of words in an article relative to the entire dataset, emphasising unique terms.
- **N-Grams:** Bigrams and trigrams capture contextual patterns like repetitive phrases.
- **Readability metrics:** Features such as word count, sentence length, and lexical diversity will be calculated to account for the stylistic differences between real and fake articles.

Phase 2: Misinformation Typology Classification

For the second phase, a more nuanced feature set is required to capture the stylistic and content-based differences across the seven misinformation types:

- **Topic modelling:** Latent Dirichlet Allocation (LDA) identifies thematic patterns within articles, aligning with specific misinformation categories.
- **Sentiment analysis:** Sentiment polarity and subjectivity scores distinguish categories such as polarised content and persuasive information
- **Named Entity Recognition (NER):** Identifies named entities such as people, places, and organisations to detect fabricated content or misreporting
- **Domain-specific keywords:** A custom dictionary of keywords and phrases will be compiled for each misinformation type.

Both stages will use word embeddings (e.g. Word2Vec or GloVe) to capture semantic relationships between words. Deep learning models will use contextual embeddings like BERT.

Model Selection

Different models will be evaluated for each phase, prioritising accuracy.

Phase 1: Real vs. Fake News Classification

For binary classification, the following models will be considered:

- **Logistic regression:** Serves as a baseline model due to its simplicity and interpretability. This model was noted to be very effective in the literature review.
- **Support Vector Machines** (SVM): Effective for high-dimensional text data, offering robust decision boundaries.
- **Random forest:** Ensemble-based method known for its feature importance insights.
- **BERT** (Bidirectional Encoder Representations from Transformers): Pre-trained BERT models leverage contextual embeddings for superior performance.

Phase 2: Misinformation Typology Classification

For multi-class classification, the following models will be considered:

- **Multinomial Naive Bayes:** Efficient for initial testing, particularly with TF-IDF features.
- **XGBoost:** Provides scalability and strong performance in multi-class settings.
- **Neural networks:** A feed-forward architecture explores non-linear relationships in the feature space.
- **Fine-tuned BERT:** A BERT model fine-tuned on the custom dataset will be employed, addressing subtle distinctions between misinformation classes.

Pipeline architecture

The pipeline is divided into two stages: Binary Classification and Multiclass Classification. The stages are connected sequentially, and the architecture incorporates feature extraction, model training, and explainability at each stage.

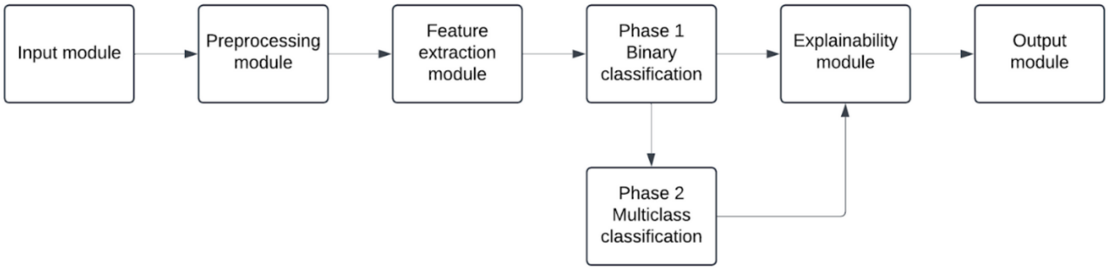

Input module
Preprocessing module
Feature extraction module
Phase 1 Binary classification
Explainability module
Output module
Phase 2 Multiclass classification

FIGURE 4 - PIPELINE ARCHITECTURE

1. Input Module

- **User Input**: Users can input either a raw text article or a URL pointing to an online news piece.
- **Preprocessing for URL Input**:
  - Scrape the article content using libraries such as BeautifulSoup or Newspaper3k.
  - Remove ads, HTML tags, and extraneous content.
  - Extract the main body of the text for analysis.

---

2. Preprocessing Module

- **Tokenization**: Split the text into individual tokens for feature extraction.
- **Text Normalization**: Case conversion, and removal of stop words, punctuation, and special characters.
- **Optional Language Detection**: Ensure the input is in English, as the system is trained on English datasets.

---

3. Feature Extraction Module

Stage 1: Binary Classification

- **TF-IDF Vectorizer**: Quantifies word importance across the dataset.
- **N-Grams**: Extract bigrams and trigrams to capture word context.
- **Readability Metrics**: Compute features like average sentence length and lexical diversity.
- **Vector Output**: A feature vector for each article, ready for the binary classifier.

Stage 2: Multi-Class Classification

- **Topic Modeling (LDA)**: Extract thematic features aligning with misinformation categories.
- **Sentiment Analysis**: Polarity and subjectivity scores for the text.
- **Named Entity Recognition (NER)**: Detect names, places, and organizations to identify fabricated content.
- **Domain-Specific Keywords**: Match article text against a custom dictionary for each category.
- **Vector Output**: An enriched feature vector for the multi-class classifier.

---

4. Classification Module

Stage 1: Binary Classification

- **Model Options**:
    - Logistic Regression (Baseline)
    - Support Vector Machine (SVM)
    - Random Forest
    - Fine-tuned BERT
- **Output**: Probability scores for "Real" or "Fake", with “Fake” proceeding to Stage 2.

Stage 2: Multi-Class Classification

- **Model Options**:
    - Multinomial Naive Bayes
    - XGBoost
    - Feedforward Neural Network
    - Fine-tuned BERT
- **Output**: A predicted label for one of seven misinformation types:
    - Fabricated Content
    - Polarised Content
    - Satire
    - Misreporting
    - Commentary

    - Persuasive Information
    - Citizen Journalism

---

5. Explainability Module

- **SHAP (SHapley Additive exPlanations)**:
    - For binary classification, highlight key words and phrases that influenced the real/fake decision.
    - For multi-class classification, explain the reasoning behind the predicted misinformation type.
- **User-Friendly Output**:
    - Highlighted text sections in the original input with explanations.
    - Visual summaries (e.g., bar charts) showing feature contributions.

---

6. Evaluation and Feedback Module

- **Evaluation Metrics**:
    - For both stages: Accuracy, Precision, Recall, F1-Score, Confusion Matrix.
    - Specific focus on minority classes for multi-class classification.
- **Feedback Collection**:
    - Allow users to provide feedback on predictions and explanations.
    - Feedback data can be used to improve model performance iteratively.

---

7. Output Module

- **Binary Classification Results**:
    - “Real” or “Fake” with a confidence score.
- **Multi-Class Classification Results**:
    - Misinformation type (if applicable) with a confidence score.
- **Explanations**:
    - Key contributing factors to the classification.
- **User Interface**:
    - Interactive UI for displaying results and explanations.

  - Options for exporting results (e.g., CSV, PDF).

Key Technologies

This project relies on Python as the primary programming language due to its extensive ecosystem of libraries for machine learning, data preprocessing, and natural language processing. The development environment will primarily use Jupyter Notebooks for iterative development and experimentation.

Key technologies and tools include:

- Programming language: **Python**, chosen for its versatility and robust support for data science workflows.
- Development environment: **Jupyter Notebooks**, enabling interactive development and seamless integration of code, visualisations, and documentation.
- Data Collection:
  - **BeautifulSoup**: For HTML parsing and web scraping
  - **Scrapy**: for large-scale scraping from diverse sources
- Text Preprocessing:
  - **NLTK** (Natural Language Toolkit): Tokenisation, stop-word removal, and lemmatisation.
  - **SpaCy**: Named Entity recognition (NER) and dependency parsing.
  - **TextBlob**: Sentiment analysis and text normalisation
- Feature extraction:
  - **TF-IDF:** via scikit-learn’s TfidfVectorizer.
  - **N-grams**: extracted using CountVectorizer in scikit-learn.
  - **Topic modeling**: Gensim for Latent Dirichlet Allocation (LDA).
  - **Word embeddings**:
    - GloVe: Pre-trained embeddings available via the Gensim library.
    - Word2Vec: Also accessible through Gensim.
    - BERT embeddings using transformers from Hugging Face.
- Classification models:

  - **Scikit-learn:** Logistic regression, Support Vector Machines (SVM), Random Forest, and Multinomial Naive Bayes.
  - **XGBoost**: A high-performance library for gradient-boosted trees, used for multiclass classification
  - **Transformers:** Fine-tuned BERT models via the transformers library by Hugging Face
  - **Tensorflow and PyTorch**: for building and fine-tuning neural network models.
- Explainability tools:
  - **SHAP** (SHapley Additive exPlanations): for feature importance analysis and user facing explanations
- User interface:
  - **Flask**: to create a lightweight backend for accepting user inputs such as text articles or URLs
  - **Newspaper3k**: to scrape and extract article text from URLs provided by users
  - **Streamlit**: For building an interactive web-based interface to display classification results and explanations.

Evaluation and Testing

For both phases, model performance will be evaluated using:

- Accuracy: The primary metric for assessing classification success
- Precision, Recall, and F1-Score: To ensure balanced performance, particularly for classes that may be harder to distinguish. This is particularly important for Phase 2.
- Confusion matrix analysis: To identify areas of misclassification.
- Explainability validation: Tools like SHAP (SHapley Additive exPlanations) analyse feature importance and provide user-facing interpretability. Feedback from user studies will assess whether explanations align with human intuition.

Success metrics for this project include:

- Achieving classification accuracy of at least 90% in Stage 1
- An F1 score of at least 0.60 across all classes in Stage 2

- High-quality explainability for classification results, validated through a user-focused evaluation process.

Additionally, stress testing will be performed by testing the system on edge cases such as short form content, and mixed-genre content. Should time allow, some scalability testing will be performed to evaluate pipeline efficiency with large-scale inputs to simulate real-world deployment.

Plan of Work

The proposed timeline for this project runs from January 1st 2025 to submission on March 24th 2025. The project plan is visualised in the Gantt chart below, and includes seven workstreams, some of which can happen concurrently.

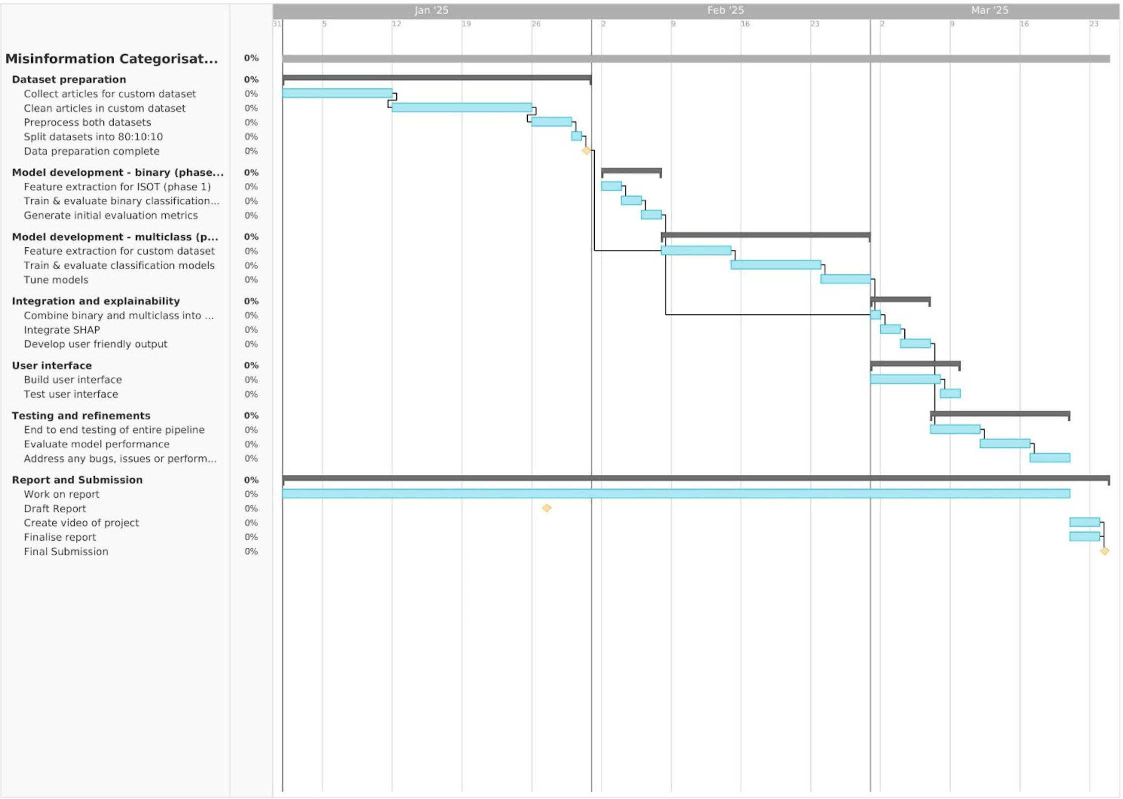


Figure 5 - Gantt Chart for Project

A simplified overview of this plan is as follows:

- Dataset preparation (January 1 - 31): Much of the time in this stage will be spent collecting and cleaning the data for the custom dataset. By the end of this phase, both datasets will be ready.
- Binary model development (February 2 - 7): Implement binary classification, phase 1 of the two stage pipeline. Model selection, tuning and evaluation should happen here.
- Multiclass model development (February 8 - 28): Implement multiclass classification, phase 2 of the two stage pipeline. Model selection, tuning and evaluation should happen here.
- Integration and explainability (March 1 - 7): Implement and test explainability features
- User interface (March 1 - 9): Build and test the user interface.
- Testing (March 7 - 20): Test the entire build, edge cases and scalability. Evaluate the models, and address any bugs or performance issues.
- Report (January 1 - March 24): Iteratively work on the project report as the project advances, in preparation for the submission on March 24th.

Prototype

Prototype Overview

The prototype focuses on implementing some key components of the proposed system for misinformation classification. It includes binary classification of news articles into "True" and "Fake" using machine learning models trained on the ISOT dataset. It also includes a sample pipeline for scraping and classifying custom data. The prototype aims to demonstrate the feasibility of these core technical features, while also exploring additional insights such as sentiment analysis and named entity recognition (NER).

The binary classification leverages logistic regression, support vector machines (SVM), and random forest classifiers, showcasing a comparative evaluation of their performance. Additionally, the content scraping pipeline uses web scraping tools to gather data from The Onion, a well-known satire site. This data is then preprocessed as the ISOT dataset was, and also processed for sentiment analysis and NER.

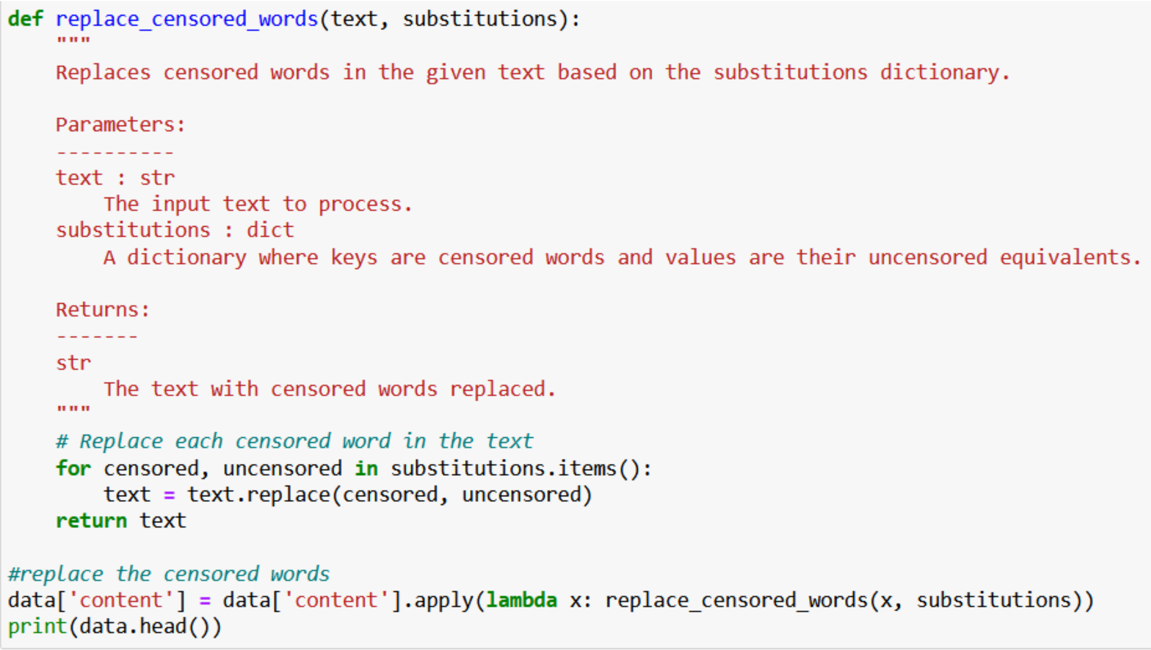
The prototype does not include a user interface or the ability for a user to test an article. No multiclass classification has been attempted as the custom dataset has not yet been collected, and the scraped articles for the prototype all belong to a single misinformation category "Satire". While some explainability features have been tested, more work is required here.

Prototype Features

Binary classification

Dataset preparation

- A random subset of the ISOT dataset was generated with 500 “True” and 500 “Fake” articles, ensuring balanced class representation.
- The articles were preprocessed to clean the text, including lowercasing, removing punctuation and stop words, and expanding censored words.



```python
def replace_censored_words(text, substitutions):
    """
    Replaces censored words in the given text based on the substitutions dictionary.

    Parameters:
    ----------
    text : str
        The input text to process.
    substitutions : dict
        A dictionary where keys are censored words and values are their uncensored equivalents.

    Returns:
    -------
    str
        The text with censored words replaced.
    """
    # Replace each censored word in the text
    for censored, uncensored in substitutions.items():
        text = text.replace(censored, uncensored)
    return text

#replace the censored words
data['content'] = data['content'].apply(lambda x: replace_censored_words(x, substitutions))
print(data.head())
```

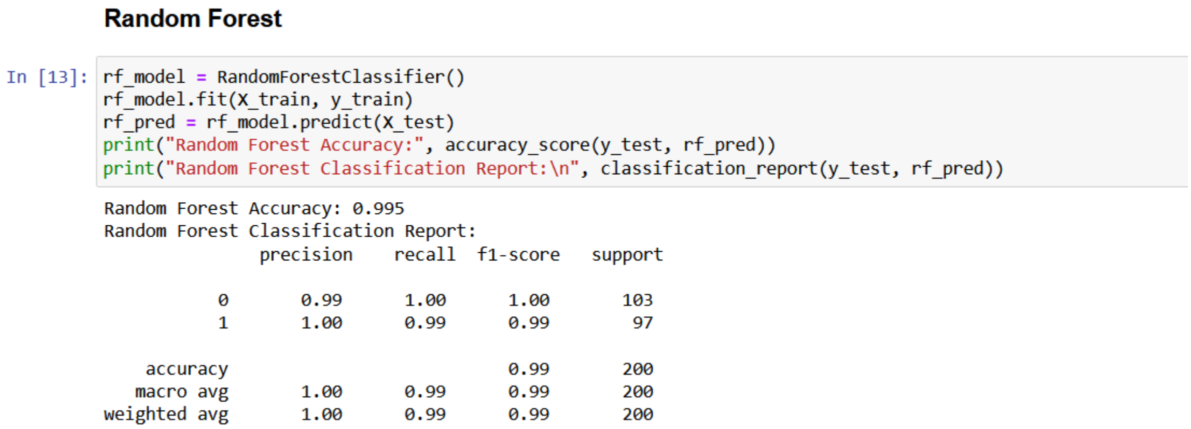
FIGURE 6 - SNIPPET OF CODE FOR EXPANDING CENSORED WORDS

TF-IDF Feature Extraction

- TF-IDF vectors were computed with unigram, bigram and trigram features to capture contextual relationships between words.
- A maximum feature count of 5000 was used to balance computational efficiency and representational power.

Model training and evaluation

- Three classifiers – Logistic Regression, Support Vector Machine and Random Forest – were trained and tested using an 80-20 train-test split.
- Evaluation metrics included accuracy, precision, recall, and F1-score. Random Forest achieved the highest accuracy of 99.5%.



Random Forest

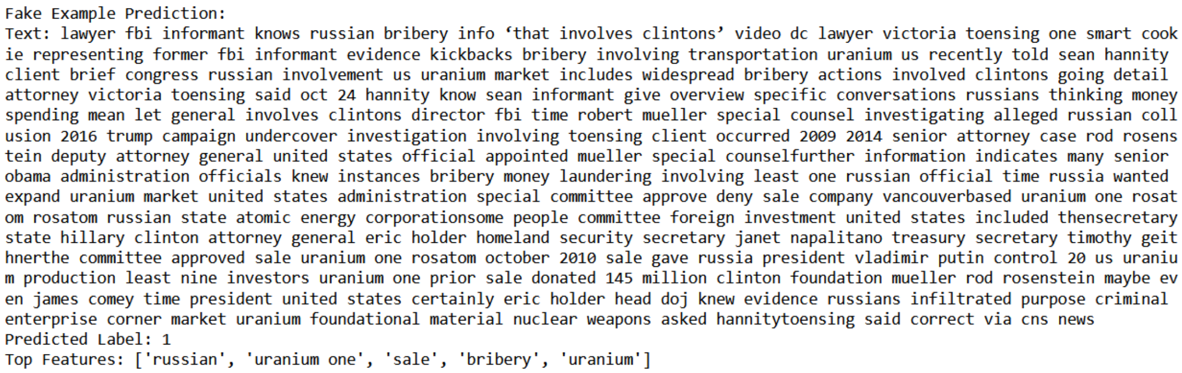
```
In [13]: rf_model = RandomForestClassifier()
         rf_model.fit(X_train, y_train)
         rf_pred = rf_model.predict(X_test)
         print("Random Forest Accuracy:", accuracy_score(y_test, rf_pred))
         print("Random Forest Classification Report:\n", classification_report(y_test, rf_pred))
```

```
Random Forest Accuracy: 0.995
Random Forest Classification Report:
               precision    recall  f1-score   support

           0       0.99      1.00      1.00       103
           1       1.00      0.99      0.99        97

    accuracy                           0.99       200
   macro avg       1.00      0.99      0.99       200
weighted avg       1.00      0.99      0.99       200
```

FIGURE 7 - RANDOM FOREST IMPLEMENTATION AND RESULTS

Explainable AI

- A feature explanation function was implemented to highlight the top influential words in predictions, enabling better interpretability.



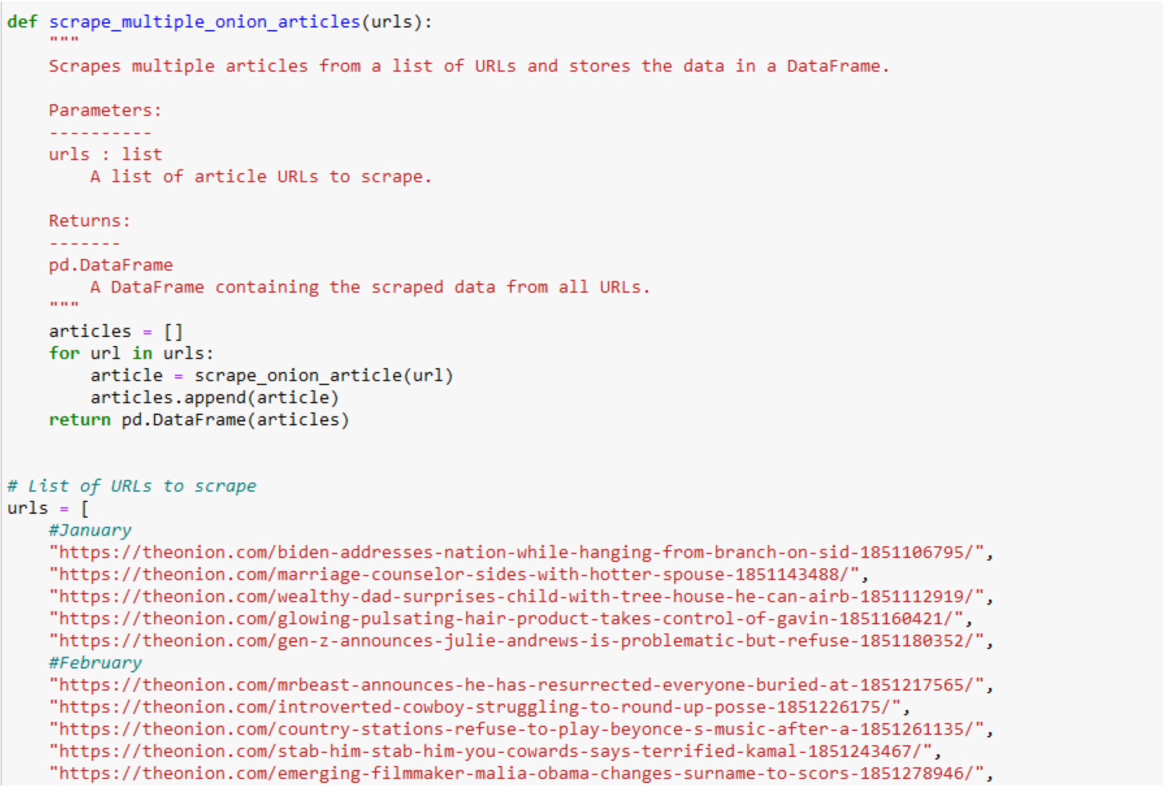
```
Fake Example Prediction:
Text: lawyer fbi informant knows russian bribery info ‘that involves clintons’ video dc lawyer victoria toensing one smart cook
ie representing former fbi informant evidence kickbacks bribery involving transportation uranium us recently told sean hannity
client brief congress russian involvement us uranium market includes widespread bribery actions involved clintons going detail
attorney victoria toensing said oct 24 hannity know sean informant give overview specific conversations russians thinking money
spending mean let general involves clintons director fbi time robert mueller special counsel investigating alleged russian coll
usion 2016 trump campaign undercover investigation involving toensing client occurred 2009 2014 senior attorney case rod rosens
tein deputy attorney general united states official appointed mueller special counselfurther information indicates many senior
obama administration officials knew instances bribery money laundering involving least one russian official time russia wanted
expand uranium market united states administration special committee approve deny sale company vancouverbased uranium one rosat
om rosatom russian state atomic energy corporationsome people committee foreign investment united states included thensecretary
state hillary clinton attorney general eric holder homeland security secretary janet napalitano treasury secretary timothy geit
hnerthe committee approved sale uranium one rosatom october 2010 sale gave russia president vladimir putin control 20 us uraniu
m production least nine investors uranium one prior sale donated 145 million clinton foundation mueller rod rosenstein maybe ev
en james comey time president united states certainly eric holder head doj knew evidence russians infiltrated purpose criminal
enterprise corner market uranium foundational material nuclear weapons asked hannitytoensing said correct via cns news
Predicted Label: 1
Top Features: ['russian', 'uranium one', 'sale', 'bribery', 'uranium']
```

FIGURE 8 - CODE SNIPPET SHOWING FEATURE HIGHLIGHTING

Web Scraping and Multi-class Preparation

Web Scraping

- A scraping pipeline was developed using BeautifulSoup to extract articles from The Onion. The scraper captured metadata including title, article text, publication date, category and URL. The URLs to scrape from were manually provided. 55 articles were scraped.

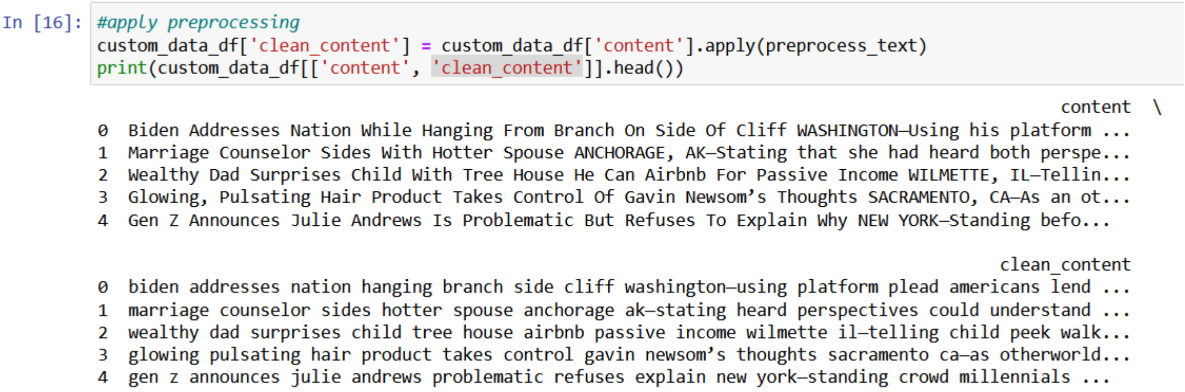


```
def scrape_multiple_onion_articles(urls):
    """
    Scrapes multiple articles from a list of URLs and stores the data in a DataFrame.

    Parameters:
    ----------
    urls : list
        A list of article URLs to scrape.

    Returns:
    -------
    pd.DataFrame
        A DataFrame containing the scraped data from all URLs.
    """
    articles = []
    for url in urls:
        article = scrape_onion_article(url)
        articles.append(article)
    return pd.DataFrame(articles)


# List of URLs to scrape
urls = [
    #January
    "https://theonion.com/biden-addresses-nation-while-hanging-from-branch-on-sid-1851106795/",
    "https://theonion.com/marriage-counselor-sides-with-hotter-spouse-1851143488/",
    "https://theonion.com/wealthy-dad-surprises-child-with-tree-house-he-can-airb-1851112919/",
    "https://theonion.com/glowing-pulsating-hair-product-takes-control-of-gavin-1851160421/",
    "https://theonion.com/gen-z-announces-julie-andrews-is-problematic-but-refuse-1851180352/",
    #February
    "https://theonion.com/mrbeast-announces-he-has-resurrected-everyone-buried-at-1851217565/",
    "https://theonion.com/introverted-cowboy-struggling-to-round-up-posse-1851226175/",
    "https://theonion.com/country-stations-refuse-to-play-beyonce-s-music-after-a-1851261135/",
    "https://theonion.com/stab-him-stab-him-you-cowards-says-terrified-kamal-1851243467/",
    "https://theonion.com/emerging-filmmaker-malia-obama-changes-surname-to-scors-1851278946/",
```

FIGURE 9 - SNIPPET OF THE DATA SCRAPING CODE

Data Preprocessing

- Similar preprocessing steps as the binary classifier were applied, including text cleaning and tokenization.

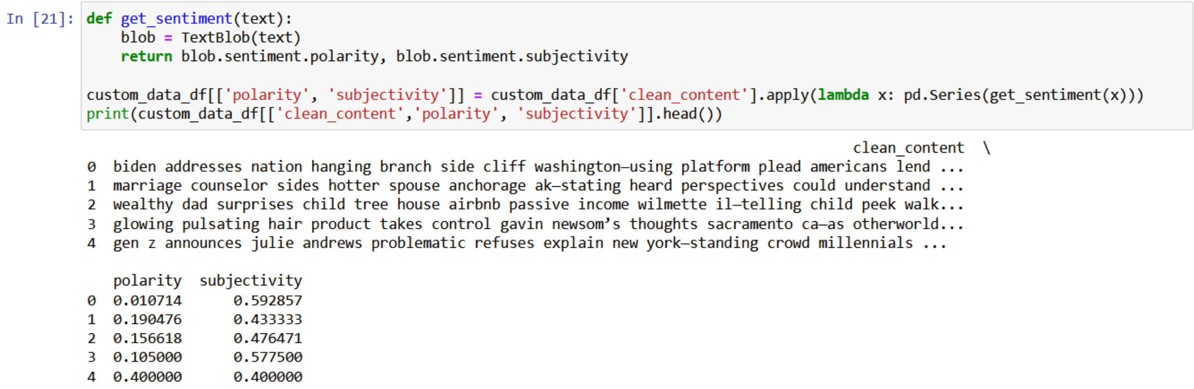


```
In [16]: #apply preprocessing
         custom_data_df['clean_content'] = custom_data_df['content'].apply(preprocess_text)
         print(custom_data_df[['content', 'clean_content']].head())

                                                   content  \
0  Biden Addresses Nation While Hanging From Branch On Side Of Cliff WASHINGTON—Using his platform ...
1  Marriage Counselor Sides With Hotter Spouse ANCHORAGE, AK—Stating that she had heard both perspe...
2  Wealthy Dad Surprises Child With Tree House He Can Airbnb For Passive Income WILMETTE, IL—Tellin...
3  Glowing, Pulsating Hair Product Takes Control Of Gavin Newsom’s Thoughts SACRAMENTO, CA—As an ot...
4  Gen Z Announces Julie Andrews Is Problematic But Refuses To Explain Why NEW YORK—Standing befo...

                                             clean_content
0  biden addresses nation hanging branch side cliff washington—using platform plead americans lend ...
1  marriage counselor sides hotter spouse anchorage ak—stating heard perspectives could understand ...
2  wealthy dad surprises child tree house airbnb passive income wilmette il—telling child peek walk...
3  glowing pulsating hair product takes control gavin newsom’s thoughts sacramento ca—as otherworld...
4  gen z announces julie andrews problematic refuses explain new york—standing crowd millennials ...
```

FIGURE 10 - SNIPPET OF PREPROCESSING CODE SHOWING BEFORE AND AFTER

Sentiment analysis

- Sentiment polarity and subjectivity scores were computed for each article using TextBlob.



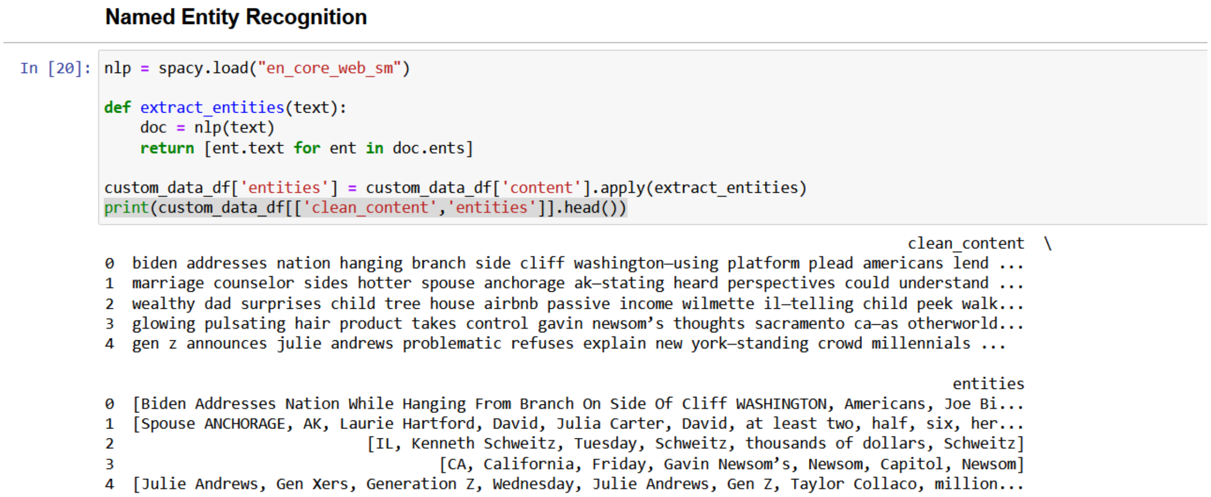
```
In [21]: def get_sentiment(text):
             blob = TextBlob(text)
             return blob.sentiment.polarity, blob.sentiment.subjectivity

         custom_data_df[['polarity', 'subjectivity']] = custom_data_df['clean_content'].apply(lambda x: pd.Series(get_sentiment(x)))
         print(custom_data_df[['clean_content','polarity', 'subjectivity']].head())
```

```
                                                       clean_content  \
0  biden addresses nation hanging branch side cliff washington—using platform plead americans lend ...
1  marriage counselor sides hotter spouse anchorage ak—stating heard perspectives could understand ...
2  wealthy dad surprises child tree house airbnb passive income wilmette il—telling child peek walk...
3  glowing pulsating hair product takes control gavin newsom's thoughts sacramento ca—as otherworld...
4  gen z announces julie andrews problematic refuses explain new york—standing crowd millennials ...

   polarity  subjectivity
0  0.010714      0.592857
1  0.190476      0.433333
2  0.156618      0.476471
3  0.105000      0.577500
4  0.400000      0.400000
```

FIGURE 11 - SNIPPET OF CODE SHOWING SENTIMENT ANALYSIS IMPLEMENTATION

Named Entity Recognition

- Named entities, such as places, people and organisations, were extracted using SpaCy to provide additional insights into the article content.



Named Entity Recognition

```
In [20]: nlp = spacy.load("en_core_web_sm")

         def extract_entities(text):
             doc = nlp(text)
             return [ent.text for ent in doc.ents]

         custom_data_df['entities'] = custom_data_df['content'].apply(extract_entities)
         print(custom_data_df[['clean_content','entities']].head())
```

```
                                                       clean_content  \
0  biden addresses nation hanging branch side cliff washington—using platform plead americans lend ...
1  marriage counselor sides hotter spouse anchorage ak—stating heard perspectives could understand ...
2  wealthy dad surprises child tree house airbnb passive income wilmette il—telling child peek walk...
3  glowing pulsating hair product takes control gavin newsom's thoughts sacramento ca—as otherworld...
4  gen z announces julie andrews problematic refuses explain new york—standing crowd millennials ...

                                                            entities
0  [Biden Addresses Nation While Hanging From Branch On Side Of Cliff WASHINGTON, Americans, Joe Bi...
1  [Spouse ANCHORAGE, AK, Laurie Hartford, David, Julia Carter, David, at least two, half, six, her...
2                          [IL, Kenneth Schweitz, Tuesday, Schweitz, thousands of dollars, Schweitz]
3                                  [CA, California, Friday, Gavin Newsom's, Newsom, Capitol, Newsom]
4  [Julie Andrews, Gen Xers, Generation Z, Wednesday, Julie Andrews, Gen Z, Taylor Collaco, million...
```

FIGURE 12 - SNIPPET OF CODE SHOWING NAMED ENTITY RECOGNITION IMPLEMENTATION

Prototype Evaluation

Binary classification

- **Accuracy**: the models demonstrated a higher accuracy than expected with Logistic Regression achieving 96.5%, SVM 97.5% and Random Forest 99.5%. All of these are above our stated success metric, which was to achieve an accuracy of over 90% for this phase.
- **Insights:** Feature importance analysis revealed that context-specific terms significantly influence predictions, showcasing the relevance of the TF-IDF approach.
- **Limitations:** Despite high accuracy, the models are sensitive to the quality of preprocessing and may struggle with more complex or nuanced articles.

Web Scraping and Multi-class Preparation

- **Scalability:** the scraper effectively captured and processed data for 55 satire articles, demonstrating scalability for larger datasets. However, the code behind the scraper is specifically built for The Onion, meaning that it will need to be rewritten for other sources, limiting scalability.
- **Sentiment analysis:** Satire articles showed varied polarity and subjectivity scores, indicating diverse tonal characteristics that could aid classifications.
- **NER insights:** Extracted entities revealed patterns, such as frequent mentions of public figures, locations, and organisations. This could support category-specific classification. However, it should be noted that the implementation of NER detection in the prototype is very noisy.

Feasibility and challenges

- The prototype showed that implementing binary classification on an existing binary dataset, and collecting a custom dataset are feasible.
- Collecting the dataset presents two challenges:
    - Some categories are easier than others to find. Satire was a simple category with a number of well known websites. Other categories, such as misreporting, will be hard to find.
    - Custom data scrapers need to be written for each website, or a new, more general data scraper needs to be written. This is because the scraper in the prototype is exclusively for articles on The Onion.
    - Custom data may require additional cleaning.
- Implementing multiclass classification may be a challenge. Implementation cannot begin until the dataset is collected, which is not until the end of January.

References

[1]

Jakub Jasiński. 2022. Propaganda on Roman Coins «IMPERIUM ROMANUM. *Imperium Romanum*. Retrieved from https://imperiumromanum.pl/en/article/propaganda-on-roman-coins/

[2]

Darrell M West. 2024. How Disinformation Defined the 2024 Election Narrative. *Brookings*. Retrieved from https://www.brookings.edu/articles/how-disinformation-defined-the-2024-election-narrative/

[3]

Benjamin A. Lyons, Jacob M. Montgomery, Andrew M. Guess, Brendan Nyhan, and Jason Reifler. 2021. Overconfidence in news judgments is associated with false news susceptibility. *Proceedings of the National Academy of Sciences* 118, 23 (June 2021). DOI:https://doi.org/10.1073/pnas.2019527118

[4]

Tobias Biró. 2024. Zwischen Wahrheit und Lüge. *Nürnberg Institut für Marktentscheidungen e.V.* Retrieved December 13, 2024 from https://www.nim.org/publikationen/detail/zwischen-wahrheit-und-luege

[5]

Katherine Ognyanova, David Lazer, Ronald E. Robertson, and Christo Wilson. 2020. Misinformation in action: Fake news exposure is linked to lower trust in media, higher trust in government when your side is in power. *Harvard Kennedy School Misinformation Review* 1, 4 (June 2020), 1–19. DOI:https://doi.org/10.37016/mr-2020-024

[6]

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (March 2018), 1146–1151. DOI:https://doi.org/10.1126/science.aap9559

[7]

Maria D. Molina, S. Shyam Sundar, Thai Le, and Dongwon Lee. 2019. "Fake News" Is Not Simply False Information: A Concept Explication and Taxonomy of Online Content. *American Behavioral Scientist* 65, 2 (2019), 000276421987822. DOI:https://doi.org/10.1177/0002764219878224

[8]

Mehedi Tajrian, Azizur Rahman, Muhammad Ashad Kabir, and Rafiqul Islam. 2023. A Review of Methodologies for Fake News Analysis | IEEE Journals & Magazine | IEEE Xplore. *ieeexplore.ieee.org*. Retrieved from https://ieeexplore.ieee.org/document/10182240

[9]

Johnson A Adeyiga, Philip Gbounmi Toriola, Temitope Elizabeth Abioye, Adebisi Esther Oluwatosin, and Oluwasefunmi 'Tale Arogundade. 2023. Fake News Detection Using a Logistic Regression Model and Natural Language Processing Techniques. *Research Square (Research Square)* (July 2023). DOI:https://doi.org/10.21203/rs.3.rs-3156168/v1

[10]

M. Sudhakar and K.P. Kaliyamurthie. 2022. Effective prediction of fake news using two machine learning algorithms. *Measurement: Sensors* 24, (December 2022), 100495. DOI:https://doi.org/10.1016/j.measen.2022.100495

[11]

Chih-yuan Li, Soon Ae Chun, and James Geller. 2024. Enhanced Multi-Class Detection of Fake News. *The International FLAIRS Conference Proceedings* 37, (May 2024). DOI:https://doi.org/10.32473/flairs.37.1.135581

[12]

Hassan Ali, Muhammad Suleman Khan, Amer AlGhadhban, Meshari Alazmi, and Junaid Qadir. 2021. All Your Fake Detector Are Belong to Us: Evaluating Adversarial Robustness of Fake-news Detectors Under Black-Box Settings. *ResearchGate*. DOI:https://doi.org/10.36227/techrxiv.1432997

[13]

William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *ACLWeb*, 422–426. DOI:https://doi.org/10.18653/v1/P17-20676

[14]

Maciej Szpakowski. 2024. several27/FakeNewsCorpus. *GitHub*. Retrieved from https://github.com/several27/FakeNewsCorpus

[15]

University of Victoria. 2022. Fake News Detection Datasets | ISOT research lab. *University of Victoria | The Online Academy Community*. Retrieved from https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets/

[16]

Clothilde Goujard. 2022. Twitter faces renewed scrutiny over disinformation in Europe. *POLITICO*. Retrieved December 22, 2024 from https://www.politico.eu/article/twitter-faces-renewed-scrutiny-over-disinformation-in-europe/

[17]

Yoel Roth. 2022. Introducing our crisis misinformation policy. *blog.x.com*. Retrieved from https://blog.x.com/en_us/topics/company/2022/introducing-our-crisis-misinformation-policy

[18]

Fact Check Tools. *toolbox.google.com*. Retrieved from https://toolbox.google.com/factcheck/about

[19]

Dennis Yap. 2024. Facticity.AI by AI Seer Named to TIME’s List of the Best Inventions of 2024 - PR.com. *PR.com*. Retrieved December 22, 2024 from https://www.pr.com/press-release/924279