Leveraging Machine Learning to Combat Fake News: A Multi-Classifier Approach for Social Media News Verification - Final Report

TABLE OF CONTENTS

1 INTRODUCTION

Template Chosen

The template chosen was CM3060 Natural Language Processing Fake News Detection project template, and the Dataset selected was BuzzFeed dataset from FakeNewsNet (Shu et al., 2018).

Github Repository

The code and the notebook for this project can be found here.

Motivation

The digital era has made information more accessible to everyone, but there is a dangerous problem that comes with it: fake news. Fake news is like a web of lies that can deceive people and affect their ability to think critically, make informed choices, and have meaningful discussions (Muhammed et al., 2022). This project aims to give technology the power to identify fake news and find the truth in the vast online news world.

Having personally witnessed the detrimental effects of fake news during the pandemic, I recognize the urgent need for robust tools to safeguard public health and informed discourse. Witnessing loved ones fall prey to misinformation on matters as crucial as healthcare ignited a passionate desire to contribute to this fight. Beyond personal motivations, this project addresses the profound societal ramifications of falsehoods. Fake news erodes trust in institutions, hinders informed citizenry, and fuels societal polarization (Joseph et al., 2022). By equipping machines with the ability to identify and flag deceptive narratives, I aim to try to safeguard against these detrimental effects and restore integrity to the online news ecosystem.

The possibilities for this project are vast. We can use improved methods to detect and reduce the spread of false information in real-time. This will help platforms and content creators create a more responsible online space. Additionally, studying the language patterns of fake news can guide future research and development, strengthening our defenses against those who try to manipulate public conversations (Chauhan & Palivela, 2021).

Ultimately, this project's core motivation is to use machine learning to expose lies and enable technology to protect against deception online. By teaching machines to distinguish between truth and deception, I aim to create a better online environment where trust and honesty are valued.

Dataset: BuzzFeed News

My primary goal is to address the urgent issue of false information by utilizing the meticulously fact-checked BuzzFeed dataset from FakeNewsNet (Shu et al., 2018). FakenewsNet is a key resource for a research project at ASU that focuses on studying fake news. The dataset includes information from Buzzfeed news and Politifact, containing both real and fake news articles. This dataset offers various perspectives, helping with fake news detection and understanding how fake news spreads. The main focus of this project is the Buzzfeed news dataset.

The dataset from Buzzfeed news consists of a thorough representation of news articles shared on Facebook by nine news agencies during a week close to the 2016 U.S. election. This timeframe includes September 19 to 23 and September 26 to 27. To ensure accuracy, every post and its

corresponding article underwent a rigorous fact-checking process conducted by five BuzzFeed journalists. The dataset is divided into two distinct collections: one containing fake news articles and the other containing real news articles. Both collections are presented in CSV format and consist of 91 observations with 12 features or variables.

Project Aims & Objectives

The main goal of this project is to create strong machine learning models that can accurately differentiate between real news articles and fake stories, in order to reduce the spread of misinformation online.

To achieve this goal, we have set several objectives:

Analyzing and Preprocessing the Dataset:

- Thoroughly analyze the FakeNewsNet dataset to understand its structure, composition, and biases.
- Preprocess the dataset to extract relevant linguistic features, contextual information, and structural cues required for training the machine learning models.
- Perform Exploratory Data Analysis (EDA) on news titles, bodies, and title lengths to extract meaningful insights and patterns.

Model Development and Training:

To ensure accurate identification of fake news, the process of developing and training machine learning models will involve multiple steps. Initially, our focus will be on differentiating between fake and real news using only the news body. Then, we will expand our analysis to include news titles as additional input features. Lastly, we will investigate the combined utilization of both the news title and body to improve the model's ability to distinguish between fake and real news.

- Detecting Fake or Real News Using News Content: Utilize machine learning algorithms like SVM, Naive Bayes, and Random Forest to create models that analyze the language and structure of news articles. Train these models with natural language processing techniques to spot patterns that indicate fake news. Assess the models' performance using metrics like accuracy, precision, recall, and F1-score.
- Detecting Fake or Real News Based on News Headlines: Expand the analysis to include news headlines as additional input features. Examine the efficiency of various machine learning algorithms in categorizing news solely based on the title. Train and evaluate the models using the same metrics as in the previous phase.
- Detecting Fake or Real News Using Both Headlines and Content: Merge information from both news headlines and content to create comprehensive models for detecting fake news. Train and validate these combined models to evaluate their ability to differentiate between fake and real news articles.

853 words

2 LITERATURE REVIEW

The spread of fake news, especially on social media, has become a major problem, making it hard to tell what's real and what's not. This affects society in many ways. Figueira et al. (2017) breaks down different ways to detect fake news:

- Content-based: Analyzing the text itself for features like suspicious grammar, vocabulary, or factual inconsistencies.

- Source-based: Checking the website, author, or domain for known biases or unreliability.

- Diffusion-based: Examining how information spreads - quickly and widely shared content may raise red flags.

Computer algorithms for detection can analyze content, how information spreads, or a combination of both. Early ones focused on identifying rumors, but newer methods target specific fake news. The paper proposes a high-level algorithm that finds topics and key figures in news articles, then checks them against other sources. This highlights challenges like:

- Entity matching: Making sure different mentions of the same person or concept (e.g., "John Smith" vs. "J. Smith") are recognized as the same.

- Limits of technology: Natural language processing (NLP) and entity extraction are not perfect, leading to potential errors.

- Source reputation: Tracking websites' and authors' reliability over time is crucial.

In summary, while early solutions showed promise, robust and reliable automatic fake news detection remains a difficult open challenge that researchers continue working to improve upon. Both human and algorithmic approaches each have advantages and limitations.

Nagaraja et al. (2021) aims to detect fake news using machine learning techniques like Naive Bayes and Support Vector Machine (SVM). It argues that there is a lot of unreliable and misleading news spread online which can influence people's opinions and decisions. By wielding the power of machine learning, it investigates the effectiveness of two algorithms, Naive Bayes and SVM, in discerning truth from falsehood.

- **Naive Bayes:** A probabilistic method relying on Bayes' theorem to classify data based on conditional probabilities. It works well when features are independent, making it suitable for analyzing individual words in text.
- **Support Vector Machine (SVM):** A powerful algorithm that creates a hyperplane separating data points into distinct classes. This allows SVMs to effectively handle complex relationships between features, potentially providing greater accuracy in fake news detection.

While the study yields promising results, particularly with SVM achieving higher accuracy (75%), precision (74%), recall (74%), and F1-score (75%) compared to Naive Bayes (Fig. 1.1 & 1.2), the study presents opportunities for further investigation and refinement.

Algorithm	Precision	Recall	F1-Score	Accuracy
Naive Bayes	68%	63%	67%	63%
SVM	74%	74%	75%	75%

Figure 1.1: *Nagaraja et al. (2021) machine learning algorithm results*

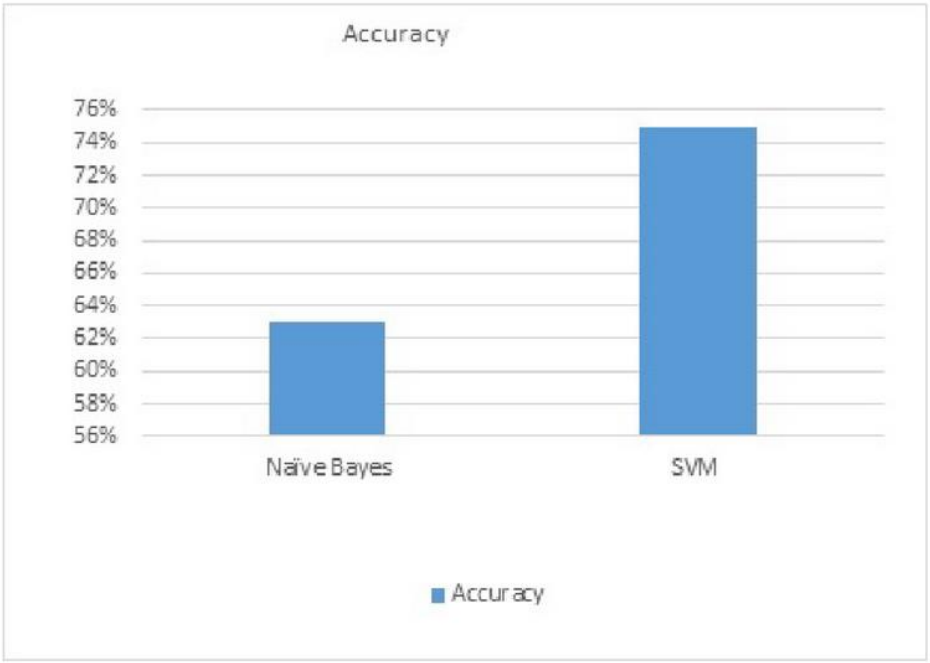

Figure 1.2: *Nagaraja et al. (2021) machine learning algorithm results*

Fake news is a big problem online, and this research is important because it addresses that issue. The paper also compares different algorithms, like Naive Bayes and SVM, which is helpful for understanding their strengths and weaknesses. In addition, the paper goes beyond just looking at word co-occurrence and uses semantic analysis to understand the deeper meaning of the text. This is important for identifying fake news. Finally, the results are presented clearly and concisely, making it easy for readers to understand and compare the findings.

However, the paper also provides chances for more research and improvement, despite its strengths. The findings cannot be applied broadly due to the small dataset used. It is essential to include larger and more diverse datasets to ensure the model's effectiveness in different news environments. Moreover, the study mainly depends on simple text pre-processing techniques. By exploring advanced feature engineering methods like n-grams, sentiment analysis, and network-based features, the data representation can be enhanced, and the model's performance can be improved. Hence, In my project I will explore powerful techniques like n-grams, sentiment analysis, and network-based features to capture richer nuances of language and context and to overcome the generalizability concerns, I will utilize significantly larger and more varied datasets encompassing a spectrum of news sources and writing styles.

Agarwal et al. (2019) aimed to analyze different machine learning tools (classifiers) for spotting fake news. They used a dataset called LIAR, containing real and fake news statements labeled "true" or "false." Before analyzing the statements, they prepared them by:

- Tokenization: Breaking down the text into individual words or meaningful units (like "apple" and "pie" instead of "applepie").
- Stemming: Reducing words to their base form (e.g., "running" becomes "run").

Then, they extracted features (informative aspects) from the statements using several methods:

- Bag-of-words: Treating each word as a separate feature, regardless of its order or context.

- N-grams: Considering groups of consecutive words (like "big house" instead of just "big" and "house").
- TF-IDF weighting: Assigning higher importance to words that appear frequently in the specific statement but less frequently overall in the dataset.

They trained five different classifiers on the processed data:

- Naive Bayes: A probabilistic method based on Bayes' theorem.
- Logistic Regression: A statistical model predicting a binary outcome based on features.
- Linear SVM: A support vector machine that separates data points into classes using a hyperplane.
- Stochastic Gradient Descent: An optimization algorithm that iteratively improves the model by adjusting its parameters.
- Random Forest: An ensemble of decision trees that vote on the final prediction.

They evaluated the classifiers' performance using three metrics:

- Precision: The ratio of correctly classified "fake" statements to all predicted "fake" statements.
- Recall: The ratio of correctly classified "fake" statements to all actual "fake" statements.
- F1 score: A harmonic mean of precision and recall, balancing both.

Results showed Linear SVM and Logistic Regression performed best, with SVM slightly edging out in F1 score. However, the model struggled with news outside its training data (e.g., technology instead of politics/economics). This suggests limitations due to the domain and data quality.

My project extends Agarwal et al.'s research by developing an ensemble model that not only leverages various classifiers but also integrates word embeddings. This will enable us to understand complex word associations and context beyond simple bag-of-words features, potentially leading to a more accurate and reliable fake news detection system.

Imagine a map where words such as "happy," "joyful," and "elated" gather together, while "gloomy," "sad," and "angry" are located on the opposite side. This unique "word map" is not hand-drawn, but rather created using two powerful techniques known as CBOW and Skip-gram, which were developed by Mikolov et al. (2013) as part of the word2vec framework.

Think of word2vec as a way to teach intelligent machines, called neural networks, to analyze vast amounts of text and understand the connections between words. These networks learn that words like "happy" and "joyful" often go hand-in-hand, while "happy" and "tsunami" are unlikely to be associated. This "word map" becomes a valuable tool for comprehending various forms of language, including the complex realm of fake news.

By inputting fake news stories into this map, one can identify if they contain unusual connections, such as "happy" and "tsunami" appearing close together. These peculiar associations could serve as warning signs for fabricated stories. It's like having a secret codebreaker for fake news!

It is important to bear in mind that the presence of an uncommon word does not necessarily indicate falsehood. Therefore, it is imperative to exercise caution when utilizing this tool. However, the word2vec techniques developed by Mikolov et al. have the potential to revolutionize the battle against misinformation by revealing concealed connections within language. Hence, In my project I plan to use word2vec as a powerful pre-processing tool for my fake news detection model. By analyzing the relationships between words in news articles, I can identify features that capture subtle

signs of misinformation, like unusual word combinations, inconsistent language patterns, or factual discrepancies.

“Support Vector Machine (SVM) and Naive Bayes Classifier (NBC) are frequently used classification models (Conroy et al., 2015; Khurana and Intelligentie, 2017; Shu et al., 2018).These two models differ a lot in structure and both of them are usually used as baseline models. Logistic regression (LR) (Khurana and Intelligentie, 2017; Bhattacharjee et al., 2017) and decision tree such as Random Forest Classifier (RFC) (Hassan et al., 2017) are also used occasionally.” (Oshikawa et al., 2020, p.4).

My project tackles these challenges by employing an ensemble approach that combines SVM, Naive Bayes, Random Forest, and Passive-Aggressive Classifiers. SVM can effectively identify complex patterns in political discourse, while Naive Bayes excels at analyzing word relationships and identifying subjective language. Random Forest's ensemble nature is robust against bias, and Passive-Aggressive Classifiers' efficient updates allow for adapting to the dynamic nature of political news. This combination addresses the specific hurdles of political fake news detection, aiming to achieve superior accuracy and real-world effectiveness.

1395 words

3 DESIGN

Overview of the Project

- **Title:** Leveraging Machine Learning to Combat Fake News: A Multi-Classifier Approach for Social Media News Verification
- **Domain:** Social media news analysis, misinformation detection
- **Problem Statement:** The spread of false information on social media platforms such as Facebook is a big problem for democracy, trust, and making informed choices. This widespread misinformation spreads quickly, manipulates emotions, and uses complicated language, so we need strong solutions to fight against its effects.
- **Project Goal:** The main goal of this project is to create a strong and adaptable machine learning model that can effectively categorize news articles as either real or fake. I am tackling the difficulties presented by news on social media. To train and assess my model, I will be utilizing the BuzzFeed dataset from FakeNewsNet, which is a reliable and labeled resource.

Template: This project builds upon the framework established by the CM3060 Natural Language Processing Fake News Detection project template.

Domain and Users

Domain: Social Media News Analysis

The fast growth of social media, along with the abundance of user-created content, has not only connected a vast number of individuals to accomplish positive things, but it has also offered convenient channels for spreading deceptive information like fake news. (Sahin et al., 2022). Such misinformation has numerous consequences, including the erosion of public trust in institutions, the polarization of opinions, and the hindrance of informed decision-making. (Balshetwar et al., 2023). This project aims to tackle the difficulties of analyzing news on social media by creating a strong machine learning model that can differentiate between genuine and false news articles in this intricate and ever-changing field.

Users:

This project targets a diverse range of stakeholders seeking to mitigate the spread of misinformation and promote the dissemination of factual information:

- **Social Media Platforms:** Improve platform credibility by filtering out false information and promoting informed user interactions.
- **News Agencies:** Verify claims and protect the integrity of journalism, ultimately building public trust in trustworthy news sources.
- **Fact-Checking Organizations:** Automate initial news verification and prioritize investigative efforts, resulting in more efficient and effective fact-checking processes.
- **Individuals:** Empower users to make informed choices based on verified information, fostering a more knowledgeable and discerning society.

Modifying the Model and Dealing with Concerns:

To effectively meet the requirements of different user groups, it is important to customize the model according to their specific difficulties and needs. Moreover, making the tool user-friendly and accessible to a wide range of people will enhance its effectiveness. It is also crucial to prioritize ethical concerns related to privacy, bias, and potential misuse during the entire process of creating and implementing the tool. By following responsible practices and promoting transparency, we can guarantee that this tool serves the greater good.

Design Choices

Dataset:

- **BuzzFeed Dataset from FakeNewsNet** (Shu et al., 2018)**:** This dataset was chosen for its unique strengths and alignment with the project's goals:
    - **Relevance to Social Media News:** The data comes from Facebook and reflects the way people talk, the types of content they share, and the social interactions that happen on social media. This makes the model useful for understanding real-life social media situations.
    - **Professional Fact-Checking:** BuzzFeed journalists have carefully fact-checked every news article, guaranteeing the trustworthiness and precision of the labels. This establishes a solid basis of excellence for training and assessing the model.
    - **Coverage of a Critical Election Period:** The dataset includes news articles from the 2016 U.S. election, a time when there was a lot of false information and division. This helps the model understand the tactics and language often used in fake news during important political events.

Exploratory Data Analysis (EDA):

- EDA is done to understand the features of news articles, looking at titles and content. Visualization methods help in grasping how the data is spread out and any trends.
- Studying news title lengths can reveal patterns linked to fake or real news. Different lengths may show patterns like sensationalism or informativeness, affecting how news is classified.

Analyzing title length distributions can also help spot unusual cases that need more investigation in fake news detection.

Classifier Selection:

This project employs a multi-classifier approach, leveraging the strengths of diverse algorithms to comprehensively analyze the complex nuances of social media news:

- **Support Vector Machines (SVM):** They are well-suited for high-dimensional text data because they can handle non-linear relationships between features effectively. Moreover, their ability to handle noise and outliers, which are often found in social media content, makes them suitable for this domain.
- **Naive Bayes Classifiers (NBC):** Their efficient modeling of word relationships and conditional probabilities allows them to effectively capture the semantic associations and patterns within news articles, proving advantageous for text classification tasks. Furthermore, their ability to handle high-dimensional data with computational efficiency makes them well-aligned with the project's dataset size.
- **Random Forest Classifiers (RFC):** Social media requires adaptable models to keep up with changing language and content strategies. RFC is a great choice because they can learn and update their decision trees with new data. This ensures that the model remains relevant and effective in a rapidly changing information ecosystem.
- **Passive-Aggressive Classifiers (PAC):** Their effective incremental model updates make them a valuable tool for constantly adjusting to the quickly changing language patterns and content strategies that are common in social media. This guarantees that the model remains relevant and effective in an online learning environment that is characterized by dynamic data streams.

Fake/Real News Classification:

Classifying false information by considering various elements of news articles, like the title, body, or both, enables a thorough examination and could enhance the precision of the categorization. Let's explore the reasons behind selecting each approach.

- Title-Based Classification:
    - The titles of news articles are usually the first thing readers notice and can greatly impact how they perceive the article.
    - Titles are generally shorter and more concise than the rest of the article, which makes them easier to process computationally.
    - Certain fake news articles may have unique patterns or language in their titles, making it simpler to classify them just based on this information.
- Body-Based Classification:
    - The body of the news article contains more detailed information and context, which can provide richer features for classification.
    - Fake news articles often contain misleading or false information within the body, making it easier to spot when analyzing the entire content.

- Analyzing the body allows for a deeper understanding of the article's content and context, which may lead to more accurate classification.

- Combined Title and Body Classification:
  - Utilizing information from both the title and body enhances the effectiveness of both methods and can lead to more robust classification models.
  - Some fake news articles may attempt to deceive readers by presenting a misleading title but a more balanced or nuanced body content. Combining features from both sources can help capture these subtleties.
  - By incorporating information from both sources, the classification model may better capture the overall context and intent of the news article, leading to improved accuracy.

- Different model evaluation metrics are selected with care to ensure a thorough assessment of classifier performance. Metrics like accuracy, precision, recall, and F1-score are chosen because they provide insights into various aspects of classifier performance. Accuracy measures the overall correctness of predictions, precision indicates the proportion of true positive predictions out of all positive predictions, recall measures the proportion of true positives correctly identified, and the F1-score balances precision and recall, making it a suitable metric for imbalanced datasets. By considering these metrics collectively, a more comprehensive evaluation of classifier performance can be attained.

Project Structure

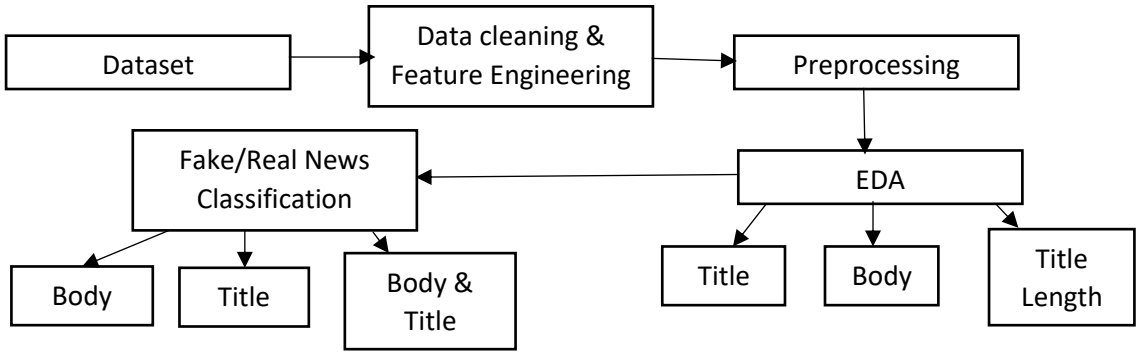

Figure 2: *Project Structure*

- **Loading Dataset:**
  - This part focuses on obtaining and loading the dataset used for analysis. Information about the dataset's origins and features can be found here.

- **Data Cleaning and Feature Engineering:**
  - During this stage, the data is cleaned to deal with missing values and unnecessary columns. Feature engineering methods are also used to uncover useful insights and generate new features, like the 'news_type' variable that distinguishes between fake and real news.

- **Exploratory Data Analysis (EDA):**

- This section is divided into several sub-sections:
    - **Preprocessing:** Describes the preprocessing steps taken before the analysis.
    - **Analysis of News Title:** Investigates the characteristics of news titles, including word frequency analysis and visualization, separately for fake and real news articles.
    - **Analysis of News Body:** Similar to the title analysis but focusing on the content/body of the news articles.
    - **Analysis of Title Length:** Studies the length distribution of news titles.

- **Fake/Real News Classification:**
    - This section focuses on the implementation of machine learning classifiers for fake/real news classification:
        - **Fake/Real News Detection Based on News Body:** Various machine learning algorithms such as Support Vector Machine, Naive Bayes, Random Forest Classifier, and Passive-Aggressive Classifiers are implemented using features extracted from the news bodies.
            1. **Splitting Data into Train and Test Datasets:** Describes the process of dividing the dataset into training and testing sets.
            2. **Model Implementation:** Details the implementation of each classifier.
            3. **Evaluation:** Evaluates the performance of the classifiers using appropriate metrics.
        - **Fake/Real News Detection Based on News Title:** Similar to the previous section, but focusing on features extracted from news titles.
        - **Fake/Real News Detection Based on Both Body and Title of News:** Combines features from both news bodies and titles for classification, followed by model implementation and evaluation.

Technologies and Methods

Programming Languages and Libraries:

- **Python:** Selected as the primary programming language due to its extensive libraries for data analysis, machine learning, and scientific computing.
- **scikit-learn:** A comprehensive library providing a wide range of machine learning algorithms, including the SVM, NBC, RFC, and PAC classifiers used in this project.
- **pandas:** A powerful library for data manipulation and analysis, enabling efficient handling of tabular data, such as the BuzzFeed dataset.
- **NumPy:** A fundamental library for numerical computing in Python, providing efficient array operations and mathematical functions essential for machine learning tasks.

Machine Learning Algorithms:

- **Support Vector Machines (SVM):** Excel at handling high-dimensional data and non-linear relationships between features, making them well-suited for classifying textual data like news articles.
- **Naive Bayes Classifiers (NBC):** Efficiently model word relationships and conditional probabilities, effectively capturing semantic associations within text, and perform well in text classification tasks.
- **Random Forest Classifiers (RFC):** Robust against bias by combining multiple decision trees, reducing the impact of individual errors, and promoting generalizability to unseen data.
- **Passive-Aggressive Classifiers (PAC):** Efficiently update models incrementally, making them suitable for online learning scenarios and adapting to evolving language patterns in social media.

Data Analysis Techniques:

- **Feature Engineering:** The process of creating informative features from raw data to improve model performance. In this project, it involves:
    - Text processing techniques for cleaning, tokenizing, and representing text data.
    - Extracting relevant features from metadata and structural elements of news articles.
- **Text Processing:** Essential for preparing text data for machine learning algorithms, including:
    - Tokenization: Splitting text into words or phrases.
    - Stop word removal: Filtering out common, uninformative words.
    - Stemming or lemmatization: Reducing words to their root forms.
    - Vectorization: Representing text as numerical vectors using techniques like bag-of-words or TF-IDF.
- **Evaluation Metrics:** Used to assess model performance and guide model selection and refinement:
    - Accuracy, Precision, Recall & F1-score

Workplan

To successfully complete this project, it is important to have a structured and organized approach. Figure 3 provides an overview of the main tasks and their respective timelines.

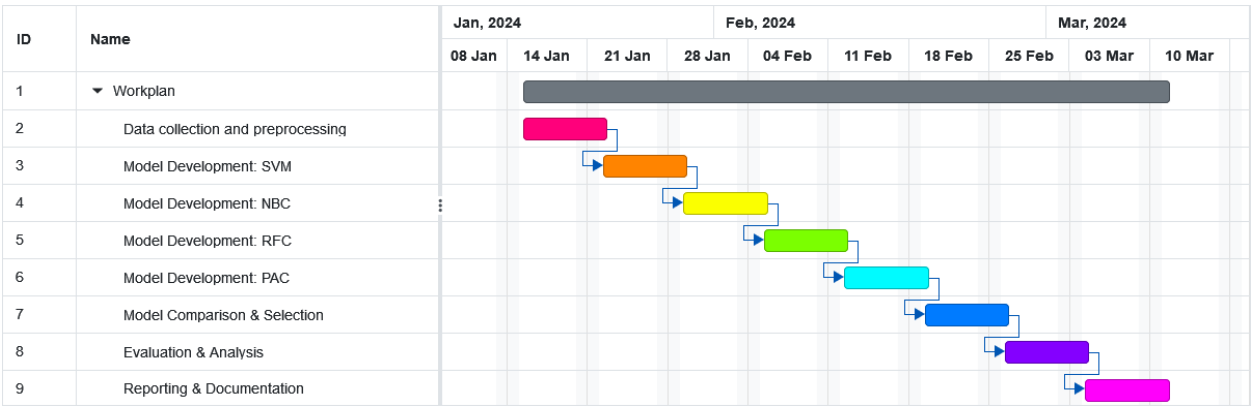

Figure 3: *Gnatt Chart*

1878 words

4 IMPLEMENTATION

The implementation of the project involved several key steps, including data cleaning, data preprocessing, feature engineering, exploratory data analysis (EDA), and the application of machine learning algorithms for fake news classification. This section provides a detailed overview of the major algorithms and techniques used, along with explanations of important parts of the code and visual representations of the results.

Data Cleaning and Feature Engineering

In this section, the dataset underwent crucial preprocessing steps to prepare it for analysis. The following techniques were employed to enhance the dataset's quality and utility:

- **Creating New Feature ('news_type'):** A new feature, 'news_type', was created by extracting relevant information from the existing 'id' column. This was achieved using a lambda function to split the 'id' string and extract the first part, which indicated whether the news was fake or real. The code snippet used for this operation is as follows:

  ```
  df['news_type'] = df['id'].apply(lambda x: x.split('_')[0])
  ```

- **Removing Unnecessary Features:** Certain columns in the dataset were deemed irrelevant for the classification task and were therefore removed to streamline the dataset. The columns removed include 'id', 'url', 'top_img', 'authors', 'publish_date', 'canonical_link', 'meta_data', 'source', 'movies', and 'images'. The following code snippet demonstrates the removal of these columns:

  ```
  df.drop(['id', 'url', 'top_img', 'authors', 'publish_date', 'canonical_link', 'meta_data', 'source', 'movies', 'images'], axis=1, inplace=True)
  ```

By creating the 'news_type' feature and removing unnecessary columns, the dataset was effectively prepared for further analysis. These steps laid the foundation for accurate classification of fake and real news articles in later stages of the project. Figure 4 displays the initial two columns of the dataset, and also that all the features are devoid of any null values.

```
[51]: df.head(2)
```

[51]:

	title	text	news_type
0	Another Terrorist Attack in NYC...Why Are we STI...	On Saturday, September 17 at 8:30 pm EST, an e...	Real
1	Donald Trump: Drugs a 'Very, Very Big Factor' ...	Less than a day after protests over the police...	Real

```
[52]: df.isnull().sum()
```

```
[52]: title        0
      text         0
      news_type    0
      dtype: int64
```

The dataset lacks null values in the text and news_type columns, which are of utmost importance to us.

Figure 4: *First two columns of the dataset, with no missing values in any features*

Exploratory Data Analysis (EDA)

In this section, we delve deeper into the exploration of the dataset, focusing on preprocessing the text data and analyzing the frequency distribution of words in news titles and bodies for both fake and real news articles.

Preprocessing

To prepare the text data for analysis, a custom preprocessing function named 'preprocess_text' (Figure 5) was employed. Here's what each step of the preprocessing function does:

- Converting text to lowercase to ensure consistency.
- Eliminating numerical digits from the text, as they often don't carry meaningful information in natural language processing tasks.
- Removing punctuation marks to focus solely on word tokens.
- Tokenizing the text into individual words using the Whitespace Tokenizer.
- Filtering out common English stopwords to remove noise from the data.
- Stemming each word using the Porter Stemmer to reduce different word forms to a common base form.

This preprocessing pipeline standardized the text data and made it ready for further analysis, guaranteeing that only important information was kept for later tasks.

```
ps = PorterStemmer()
wst = WhitespaceTokenizer()

def preprocess_text(text):
    # Convert text to lowercase
    text = text.lower()

    # Remove numbers
    text = re.sub(r'\d+', '', text)

    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))

    # Tokenize text and remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [ps.stem(word) for word in wst.tokenize(text) if word not in stop_words and word.isalpha()]

    return tokens
```

Figure 5: *Preprocess function*

Analysis of News Title

The analysis of news article titles involved examining the frequency distribution of words in both fake and real news titles. This analysis provided insights into the linguistic characteristics and thematic priorities of news articles. The following steps were undertaken:

- Separate analysis was conducted for fake and real news titles.
- CountVectorizer was initialized with the 'preprocess_text' function to transform the text data into numerical features.
- The 20 most frequent words in the titles of both fake and real news were identified.
- These top words were visualized using a bar plot to compare their frequency distribution between fake and real news titles.

Figure 6 shows the top 20 words used in titles. Term Frequency Analysis shows different patterns in fake and real news titles. Fake news often mentions terms like "Hillary," "Clinton," "freedom," and "Obama," focusing on political figures. On the other hand, real news titles feature terms like "Trump," "Clinton," "Donald," and "debate," highlighting current events and political discussions related to the 2016 U.S. election. This difference in word usage indicates varying thematic priorities and narrative angles taken by news outlets, with fake news concentrating on political figures and real news emphasizing broader political discussions and events.

The dataset provides information about the main topics and content of news articles that were shared on Facebook before the 2016 U.S. election. The prominence of specific terms in fake and real news titles highlights possible biases and divisions in news reporting during the election period. Additional analysis, like sentiment analysis or topic modeling, could offer more detailed understanding of the emotions and themes present in the news articles.

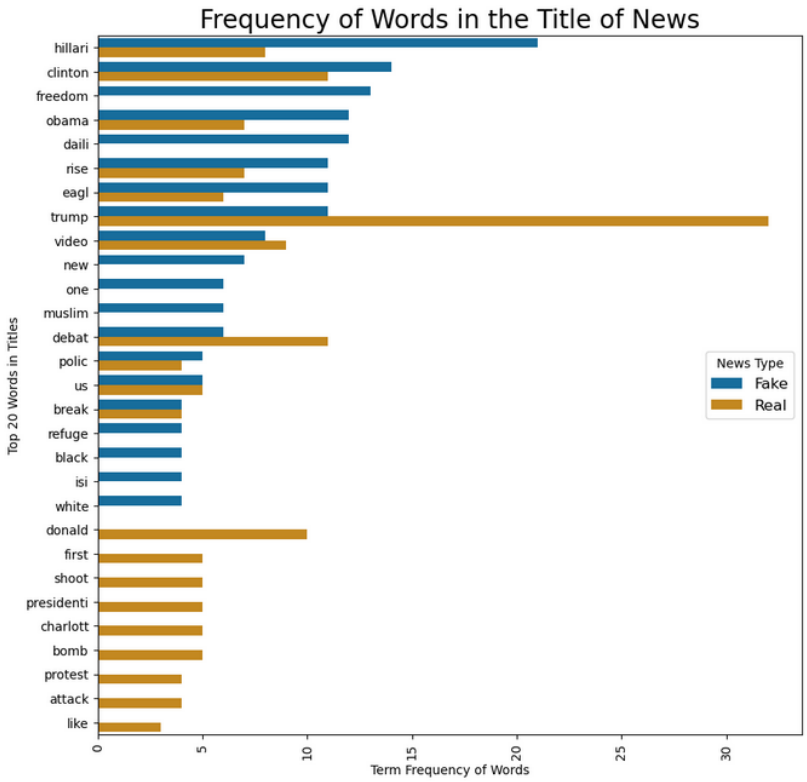

Figure 6: *Top 20 words in Titles*

Analysis of News Body

Similar to the title analysis, the content of news articles bodies was analyzed to identify the top 50 significant words in both fake and real news articles. This analysis provided a deeper understanding of the linguistic patterns and topical themes present in the news articles. The steps involved in this analysis were as follows:

- Separate analysis was conducted for fake and real news bodies.
- CountVectorizer was initialized with the 'preprocess_text' function to transform the text data into numerical features.
- The 50 most common words in the bodies of both fake and real news articles were identified.

- These top words were visualized using a bar plot to compare their frequency distribution between fake and real news bodies.

From the plot (Figure 7), it is evident that words such as "Trump" and "Clinton" dominate the frequency distribution in the news body. This suggests a significant focus on these political figures in the news articles. Furthermore, the term frequency analysis indicates distinct patterns between fake and real news. In fake news, terms like "Clinton," "Hillary," and "Trump" emerge as prominent, implying a strong bias towards political narratives involving these figures. Conversely, in real news, terms like "Trump," "said," and "Clinton" stand out, indicating a focus on factual reporting and discussions surrounding the statements made by political figures like Trump and Clinton.

Additionally, it is interesting to note that terms like "Muslim" and "terrorist" exclusively appear in fabricated news articles, indicating a potential trend of sensationalism or prejudiced reporting aimed at inciting fear or prejudice. This emphasizes the importance of critically evaluating news sources and being aware of potential biases, particularly when dealing with sensitive topics like religion and terrorism.

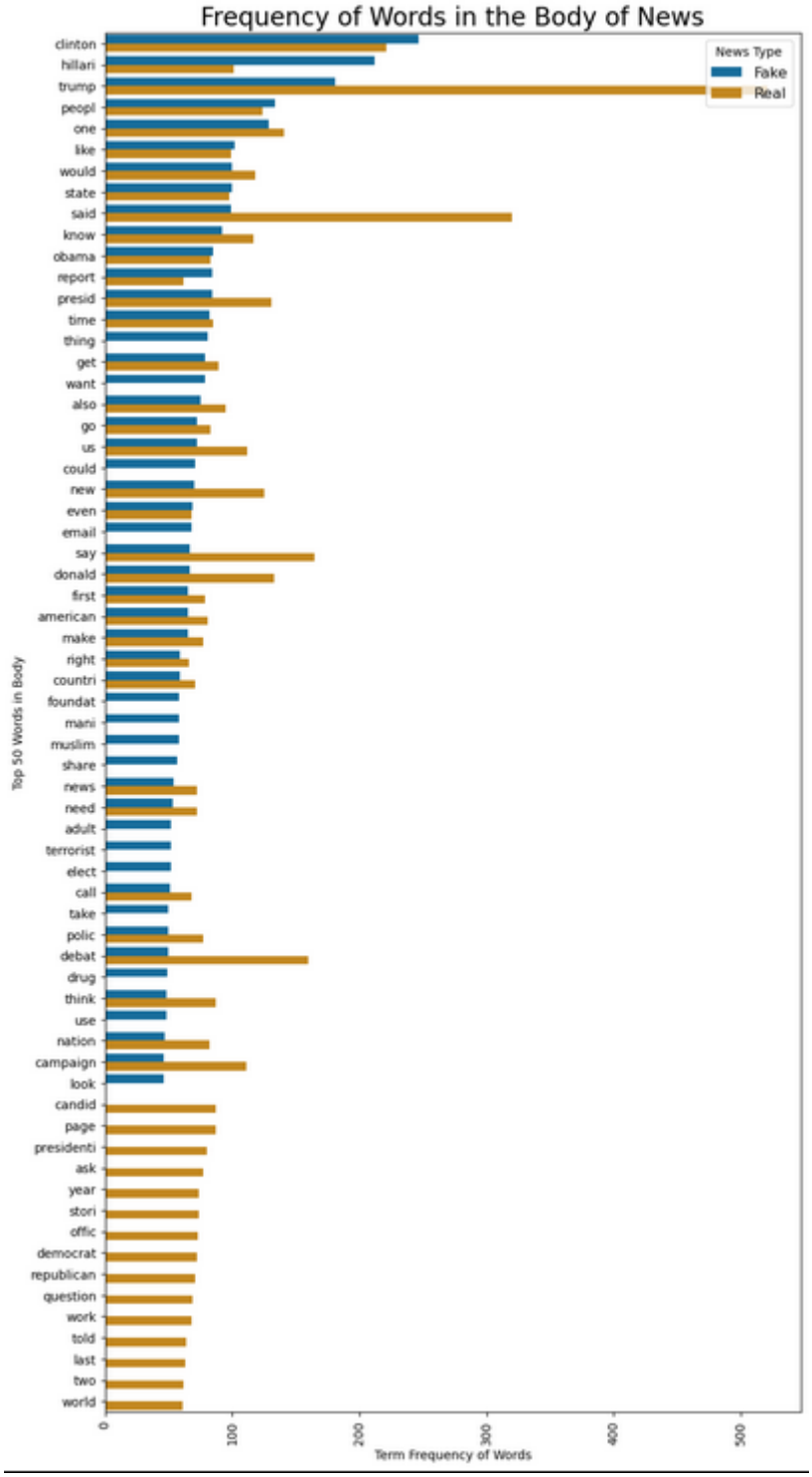

Figure 7: *Top 50 Words in Body*

Analysis of Title Length

Finally, the length of news article titles was examined to determine if it served as a distinguishing factor between fake and real news. This analysis provided insights into how news articles are written and what they focus on. The following steps were undertaken:

- The length of each news article title was calculated.

- Kernel Density Estimation (KDE) plots were generated to visualize the distribution of title lengths for both fake and real news articles.

Figure 8 shows that on average, fake news titles are a bit longer than real news titles. Real news titles usually have around 60 characters, showing a consistent pattern. However, fake news titles have a slight skewness, with the highest density around 80 characters. This difference in title length suggests that there may be variations in editorial styles or content emphasis between fake and real news articles.

Overall, the EDA section thoroughly examined the dataset, revealing insights into the linguistic characteristics, thematic priorities, and editorial practices of news articles.

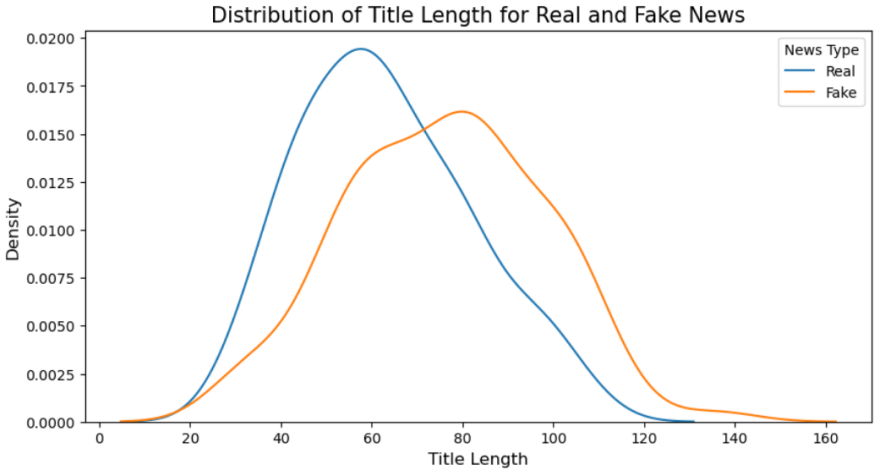

Figure 8: *Distribution of Title Length for Real and Fake News*

Fake/Real News Classification

In this section, we delve into the implementation of machine learning classifiers to categorize news articles as either fake or real based on their content. The objective is to assess the performance of various classification algorithms and determine their effectiveness in distinguishing between fake and real news.

Fake/Real News Detection Based on News Body

The classification task begins with the analysis of news article bodies. The following steps outline the implementation process:

- **Data Splitting**: The dataset is divided into training and testing sets using the train_test_split function from scikit-learn. This ensures that the classifiers are trained on a subset of the data and evaluated on an independent subset.
- **Feature Extraction**: The CountVectorizer is initialized with the 'preprocess_text' function to transform the textual data into numerical features. This step converts the text data into a matrix of token counts, which serves as input for the classification algorithms.
- **Classifier Implementation**: Several classification algorithms are implemented to classify news articles based on their bodies (Figure 9.1):

- **Support Vector Machine (SVM)**: The transformed training data is used to train a SVM model with a linear kernel. SVM is effective in high-dimensional spaces and is suitable for binary classification tasks.
- **Naive Bayes Classifiers**: The Multinomial Naive Bayes classifier is trained using the modified training data. Naive Bayes classifiers are statistical models that rely on Bayes' theorem and are frequently employed for categorizing text.
- **Random Forest Classifier**: The training data is used to train a Random Forest classifier with 100 trees. This method combines predictions from multiple decision trees to enhance accuracy.
- **Passive-Aggressive Classifiers**: The transformed training data is used to train a Passive-Aggressive classifier with a maximum of 1000 iterations. These classifiers are online learning algorithms that adjust the model parameters when instances are misclassified.

- **Model Evaluation**: The performance of each classifier is evaluated using various evaluation metrics, including accuracy, precision, recall, and F1-score by an evaluation function (Figure 9.2). These metrics provide insights into the classifiers ability to correctly classify fake and real news articles.
- **Results:** SVM is the top performer, offering the highest accuracy and a well-balanced mix of precision and recall (Figure 9.3). Nevertheless, the other classifiers also demonstrate competitive performance, with different trade-offs between precision and recall.

Support Vector Machine

```
# Initialize Support Vector Machine classifier with linear kernel
svm_model = SVC(kernel='linear')
# Train the SVM model using the transformed training data
svm_model.fit(X_train_transformed, y_train)
# Use the trained SVM model to predict labels for the test data
svm_pred = svm_model.predict(X_test_transformed)
```

Naive Bayes Classifiers

```
# Initialize Naive Bayes classifier
nb_model = MultinomialNB()
# Train the Naive Bayes model using the transformed training data
nb_model.fit(X_train_transformed, y_train)
# Use the trained Naive Bayes model to predict labels for the test data
nb_pred = nb_model.predict(X_test_transformed)
```

Random Forest Classifier

```
# Initialize Random Forest classifier with 100 trees
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
# Train the Random Forest model using the transformed training data
rf_model.fit(X_train_transformed, y_train)
# Use the trained Random Forest model to predict labels for the test data
rf_pred = rf_model.predict(X_test_transformed)
```

Passive-Aggressive Classifiers

```
# Initialize Passive-Aggressive classifier with maximum iterations set to 1000
pa_model = PassiveAggressiveClassifier(max_iter=1000, random_state=42)
# Train the Passive-Aggressive model using the transformed training data
pa_model.fit(X_train_transformed, y_train)
# Use the trained Passive-Aggressive model to predict labels for the test data
pa_pred = pa_model.predict(X_test_transformed)
```

Figure 9.1: *Classifiers Implementation of Support Vector Machines, Naive Bayes classifiers, Random Forest Classifier, and Passive-Aggressive Classifiers*

```
# Evaluation
def evaluate_model(y_true, y_pred):
    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, pos_label='Fake')
    recall = recall_score(y_true, y_pred, pos_label='Fake')
    f1 = f1_score(y_true, y_pred, pos_label='Fake')
    print("Accuracy: {:.2f}".format(accuracy))
    print("Precision: {:.2f}".format(precision))
    print("Recall: {:.2f}".format(recall))
    print("F1-score: {:.2f}".format(f1))
```

Figure 9.2: *Evaluation function*

```
Support Vector Machine:
Accuracy: 0.81
Precision: 0.82
Recall: 0.78
F1-score: 0.80

Naive Bayes Classifier:
Accuracy: 0.73
Precision: 0.75
Recall: 0.67
F1-score: 0.71

Random Forest Classifier:
Accuracy: 0.78
Precision: 0.81
Recall: 0.72
F1-score: 0.76

Passive-Aggressive Classifier:
Accuracy: 0.76
Precision: 0.74
Recall: 0.78
F1-score: 0.76
```

Figure 9.3: Evaluation Result for Fake/Real News Detection Based on News Body

Fake/Real News Detection Based on News Title

Similarly, the classification task is repeated based on news article titles. The implementation process follows a similar outline as described above, with the main difference being the input features (i.e., news titles instead of news bodies) used for classification.

Results: The classifiers' performance varied when analyzing news titles (Figure 10). The Naive Bayes Classifier had the highest accuracy at 65%, with a precision of 60% and recall of 83%, resulting in an F1-score of 70%. However, the Support Vector Machine and Passive-Aggressive Classifier achieved lower accuracies of 54% and 59% respectively. The Support Vector Machine and Passive-Aggressive Classifier had F1-scores of 51% and 62% respectively. The Random Forest Classifier had an accuracy of 62%, with a precision of 75% and a recall of 33%, resulting in an F1-score of 46%. Overall, although the Naive Bayes Classifier performed relatively well, the Support Vector Machine and Passive-Aggressive Classifier showed lower performance in classifying news titles as fake or real.

```
Support Vector Machine:
Accuracy: 0.54
Precision: 0.53
Recall: 0.50
F1-score: 0.51

Naive Bayes Classifier:
Accuracy: 0.65
Precision: 0.60
Recall: 0.83
F1-score: 0.70

Random Forest Classifier:
Accuracy: 0.62
Precision: 0.75
Recall: 0.33
F1-score: 0.46

Passive-Aggressive Classifier:
Accuracy: 0.59
Precision: 0.57
Recall: 0.67
F1-score: 0.62
```

Figure 10: Evaluation Result for Fake/Real News Detection Based on News Title

Fake/Real News Detection Based on Both Body and Title of News

In this approach, information from the news title and body to enhance classification accuracy. The process involves merging the title and body into one feature and using the same classification algorithms as mentioned earlier. By utilizing information from both the title and body, this approach aims to improve the classifiers' ability to differentiate between fake and real news.

Results: The Support Vector Machine and Passive-Aggressive Classifier had the best accuracy and F1-score among the models (Figure 11). SVM and Passive-Aggressive Classifier both had an accuracy of 81%, with precision scores of 87% and 82%. The Naive Bayes Classifier had a slightly lower accuracy of 73%, while the Random Forest Classifier had an accuracy of 78%. In conclusion, SVM and Passive-Aggressive Classifier performed well in classifying news articles as fake or real based on their content.

Overall, the section on Fake/Real News Classification thoroughly examines various classification methods and how well they can accurately classify news articles as fake or real. By using multiple classifiers and evaluation metrics, the performance of the classification models is carefully assessed, helping to make informed decisions when identifying fake news articles.

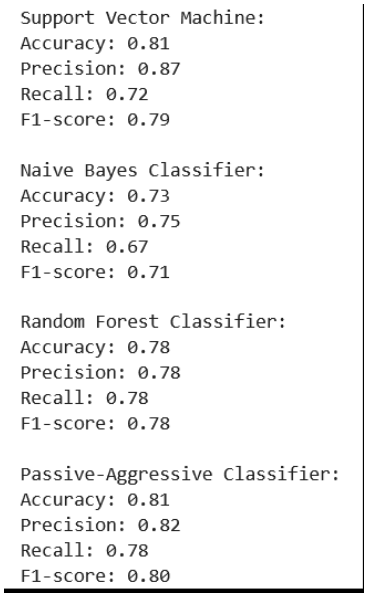

Figure 11: Evaluation Result for Fake/Real News Detection *on Both Body and Title of News*

1986 words

5 EVALUATION

Exploratory Data Analysis (EDA) Insights

Analyzing the dataset gave us important information about the features and trends of fake and real news articles:

- **Thematic Priorities:** The analysis of word frequencies in news titles showed that fake and real news have different thematic priorities. Fake news titles often highlighted political figures like "Hillary" and "Clinton," suggesting a focus on exaggerated political stories. On the other hand, real news titles focused on broader political discussions and events, featuring terms like "Trump" and "debate" prominently.
- **Narrative Angles:** The presence of specific terms in fake and real news titles indicated that news outlets adopt different narrative angles. Fake news tended to focus on political figures, while real news concentrated on reporting facts and discussing statements made by political figures.
- **Sensationalism and Prejudice:** Terms like "Muslim" and "terrorist" were exclusively associated with fabricated news articles, suggesting a potential trend of sensationalism or biased reporting aimed at provoking fear or prejudice. This emphasizes the importance of critically evaluating news sources, especially when it comes to sensitive topics.
- **Editorial Styles:** Analysis of title lengths suggested variations in editorial styles or content emphasis between fake and real news articles. Fake news titles were slightly longer on average, possibly reflecting a tendency towards more sensational or attention-grabbing headlines.
- **Consistency in Real News Titles:** Real news titles exhibited a consistent pattern of length, with the majority hovering around 60 characters. This consistency could imply a focus on clarity and conciseness in conveying news topics.

Machine Learning Classification Performance

The evaluation of the machine learning classifiers for fake news detection yielded varying results based on different feature sets and classification approaches.

- **Fake/Real News Detection Based on News Body:**
    - Support Vector Machine (SVM) exhibited the highest accuracy at 81%, with a balanced precision and recall (F1-score: 80%).
    - Naive Bayes Classifier achieved an accuracy of 73%, with relatively lower precision and recall (F1-score: 71%).
    - Random Forest Classifier and Passive-Aggressive Classifier performed moderately well, with accuracies of 78% and 76% respectively.
- **Fake/Real News Detection Based on News Title:**
    - Naive Bayes Classifier showed the highest accuracy at 65%, with a notable precision-recall trade-off (F1-score: 70%).
    - SVM and Passive-Aggressive Classifier had lower accuracies at 54% and 59% respectively, indicating challenges in classifying news based solely on titles.
    - Random Forest Classifier exhibited an accuracy of 62% with a lower F1-score of 46%.
- **Fake/Real News Detection Based on Both Body and Title of News:**
    - SVM and Passive-Aggressive Classifier demonstrated the highest accuracy at 81%, with robust precision and recall scores (F1-score: 79% and 80% respectively).
    - Naive Bayes Classifier achieved an accuracy of 73% with a balanced precision-recall performance (F1-score: 71%).
    - Random Forest Classifier maintained a reasonable accuracy of 78%, with consistent precision and recall (F1-score: 78%).

The results highlight how crucial it is to choose the right features and models when dealing with the challenging job of detecting fake news. Improving and experimenting with different feature engineering methods could boost the performance of classifiers, making social media news verification systems more efficient.

503 words

6 CONCLUSION

Reflection on Aims and Goals

Upon reflecting on the aims and goals of the project, several significant insights and observations become evident:

- **Goal Achievement**: The main aim of the project was to use machine learning methods to confirm if social media news articles are genuine. By using different classification algorithms and thorough evaluation, we have moved closer to reaching this goal. The classifiers showed different levels of success in identifying fake and real news articles, giving us useful information on how machine learning can help fight misinformation.

- **Challenges Faced**: During the project, I faced various difficulties and hurdles that needed careful thinking and finding solutions. These difficulties involved preparing text data, choosing suitable features, adjusting hyperparameters, and assessing model performance. Overcoming these challenges demanded a mix of technical knowledge, experimentation, and continuous improvement of methods. By conquering these obstacles, I made substantial strides in reaching our project objectives and enhancing our comprehension of machine learning techniques for detecting fake news.
- **Limitations and Constraints**: Despite the advancements achieved, it is crucial to recognize the project's restrictions and boundaries. One constraint is the dependence on a solitary dataset, which might not completely encompass the variety of news articles and situations found in real-life scenarios. Moreover, the classifiers' effectiveness could be affected by factors like data imbalance, feature selection, and model complexity, necessitating additional research and improvement.
- **Implications:**
    - **Media Literacy:** It is crucial to have media literacy skills to distinguish between fake and real news in today's fast-changing digital media world.
    - **Editorial Responsibility:** The differences in themes and editorial styles between fake and real news show how editorial choices can influence public opinions and stories.
- **Future Directions**:
    - **Advanced Classification Techniques:** Future studies may investigate sophisticated classification methods like deep learning models or ensemble techniques to enhance the precision and resilience of fake news detection algorithms.
    - **Multimodal Analysis:** Incorporating extra elements like metadata, social media engagement metrics, or image analysis could boost the thoroughness and dependability of fake news detection systems
    - **Real-time Monitoring:** Implementing developed models for real-time monitoring of news feeds could assist in early identification and reduction of fake news spread, supporting the overall initiatives in fighting misinformation and upholding media integrity.
- **Ethical Considerations**: It's important to think about the ethical issues of using machine learning to detect fake news. Even though fighting misinformation is good, there are risks and unintended outcomes with automated content control and censorship. So, it's vital to take a responsible and open approach to creating and using models, making sure they are fair, accountable, and respect freedom of speech.

In conclusion, analyzing our objectives and targets gives us important information about the advancements, obstacles, and possibilities for further research and improvement in the area of detecting fake news through machine learning methods. Through a thorough assessment of our accomplishments and reflecting on the wider societal impacts, we can enhance our knowledge and efficiency in combating the intricate issue of misinformation in the modern era.

492 words

7 REFERENCES

[1] Figueira, Á., & Oliveira, L. 2017. The current state of fake news: challenges and opportunities. Procedia Computer Science, 121, 817-825. https://doi.org/10.1016/j.procs.2017.11.106
[2] Nagaraja, A., K N, S., Sinha, A., Rajendra Kumar, J. V., & Nayak, P. 2021. Fake news detection using machine learning methods. Proceedings of the International Conference on Data Science, E-learning and Information Systems (DATA'21), 185–192. Association for Computing Machinery. https://doi.org/10.1145/3460620.3460753
[3] Agarwal, V., Sultana, H. P., Malhotra, S., & Sarkar, A. 2019. Analysis of classifiers for fake news detection. Procedia Computer Science, 165, 377-383. https://doi.org/10.1016/j.procs.2020.01.035
[4] Mikolov, T., Chen, K., Corrado, G., & Dean, J. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
[5] Oshikawa, R., Qian, J., & Wang, W. Y. 2020. A Survey on Natural Language Processing for Fake News Detection. arXiv:1811.00770v2.
[6] Sahin, M. E., Tang, C., & Al-Ramahi, M. A. 2022. Fake news detection on social media: A word embedding-based approach. Computer Information Systems Faculty Publications, 9. https://digitalcommons.tamusa.edu/cis_faculty/
[7] Balshetwar, S. V., RS, A., & R, D. J. 2023. Fake news detection in social media based on sentiment analysis using classifier techniques. Multimedia Tools and Applications, 82(20), 35781–35811. https://doi.org/10.1007/s11042-023-14883-3https://doi.org/10.1007/s11042-023-14883-3
[8] Joseph, A. M., Fernandez, V., Kritzman, S., Eaddy, I., Cook, O. M., Lambros, S., Jara Silva, C. E., Arguelles, D., Abraham, C., Dorgham, N., Gilbert, Z. A., Chacko, L., Hirpara, R. J., Mayi, B. S., & Jacobs, R. J. 2022. COVID-19 misinformation on social media: A scoping review. Cureus, 14(4), e24601. https://doi.org/10.7759/cureus.24601
[9] Chauhan, T., & Palivela, H. 2021. Optimization and improvement of fake news detection using deep learning approaches for societal benefit. International Journal of Information Management Data Insights, 1(2), 100051. https://doi.org/10.1016/j.jjimei.2021.100051
[10] Muhammed, T. S., & Mathew, S. K. 2022. The disaster of misinformation: A review of research in social media. International Journal of Data Science and Analytics, 13(4), 271–285. https://doi.org/10.1007/s41060-022-00311-6
[11] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. arXiv:1809.01286.
[12] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. 2017. Fake News Detection on Social Media: A Data Mining Perspective. ACM SIGKDD Explorations Newsletter, 19(1), 22-36.
[13] Shu, K., Wang, S., & Liu, H. 2017. Exploiting Tri-Relationship for Fake News Detection. arXiv preprint arXiv:1712.07709

Total words count: 7120 of 9000 words