A Project Activity Report
Submitted for
MACHINE LEARNING
(UML 501)

FAKE NEWS PREDICTION

Submitted By:
RAVNEET KAUR - 102203202
HARDETYA GILL - 102203213

Submitted To:
DR. ARUN PUNDIR

III year, BE COE

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(A DEEMED TO BE UNIVERSITY),
PATIALA, PUNJAB, INDIA
Semester V: Jul - Dec 2024

TABLE OF CONTENTS

INTRODUCTION

With the proliferation of digital platforms, the spread of misinformation has become a significant concern. Fake news not only misleads individuals but also has the potential to influence elections, social movements, and public perception. This project addresses this challenge by leveraging state-of-the-art machine learning models to classify news articles as real or fake.

A fake news predictor is necessary because of the growing prevalence and impact of misinformation in today's digital landscape. Here are key reasons highlighting its importance:

1. Combating Misinformation

Fake news spreads rapidly through social media and digital platforms, often reaching large audiences before it can be corrected. A predictor helps identify and flag false information early, reducing its potential to mislead.

2. Preserving Public Trust

Misinformation undermines trust in credible news sources, institutions, and organizations. A reliable tool to detect fake news can help restore confidence in media and information channels.

3. Mitigating Social and Political Consequences

Fake news has been used to:

- Influence elections.
- Incite violence or hatred.
- Spread propaganda.
- By detecting such articles, a predictor can help prevent these harmful outcomes.

4. Supporting Fact-Checking Organizations

Automating fake news detection reduces the burden on human fact-checkers, enabling them to focus on nuanced cases while improving the efficiency of content validation processes.

5. Promoting Media Literacy

A fake news predictor raises awareness about misinformation and teaches users to critically evaluate the authenticity of information they encounter.

6. Improving Content Moderation

Platforms like Facebook, Twitter, and YouTube face challenges in moderating fake content. Predictors can assist in flagging dubious content for further review, maintaining healthier information ecosystems.

7. Economic Implications

Misinformation can disrupt economies by spreading false information about companies, markets, or products. Accurate detection helps mitigate financial losses caused by such disruptions.

8. Strengthening Democratic Processes

In democracies, informed decision-making by the public is vital. Fake news undermines this by spreading biased or false narratives. Predictors safeguard the integrity of information in democratic discourse.

9. Public Safety

During crises, fake news can lead to panic or harmful actions. For example, misinformation about health (e.g., COVID-19 remedies) has had serious consequences. Detecting fake news helps protect public health and safety.

By deploying a fake news predictor, we can mitigate the risks associated with misinformation and promote a more informed, safe, and trustworthy digital environment.

OBJECTIVE

Fake news detection has emerged as a critical field of research. Traditional methods for verifying news rely on human fact-checkers, which are time-intensive and impractical for large-scale analysis. Automated solutions have gained prominence due to advancements in machine learning and NLP. Techniques such as BERT, transformers, and synthetic data generation have shown promise in understanding the semantic nuances of language, which is crucial for identifying misinformation. This project builds on these developments by incorporating modern methodologies and tools.

Our objectives when carrying out this project were:

- Develop a machine learning model capable of detecting fake news with high accuracy.
- Create a balanced dataset comprising real and fake news using both publicly available databases and synthetic data.
- Integrate a BERT-based architecture for superior contextual understanding.
- Deploy an API-driven system for real-time news classification.
- Provide a user-friendly interface to facilitate easy interaction with the model.

METHODOLOGY

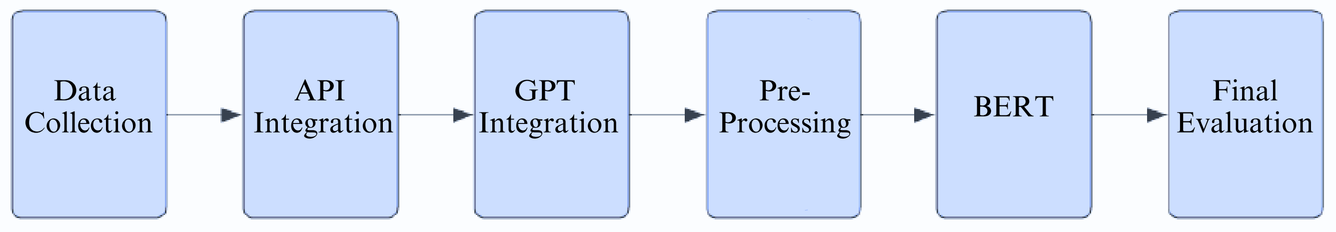

Fig 1: Flowchart of the methodology for data processing and training.

1. Data Collection:

Instead of using publicly available datasets such as LIAR and Kaggle, we bulit our own dataset. The dataset had 4 columns- Author Name, Title, Content and Label. We added the real news entries using snippets available online and the fake news articles were procured form ChatGPT. In addition, synthetic news articles were also generated using GPT-based APIs to diversify the dataset.

2. API Integration:

APIs for fetching news articles and real-time predictions were integrated. So, our model not only trained on the dataset made by us, but also on the articles being added simultaneously by the API.

3. GPT Integration:

GPT is a generative model trained to produce coherent and contextually relevant text. It's major functionalities are:

Text Coherence: Produces high-quality articles that resemble real news, making the detection task more challenging.

Prompt Flexibility: Can generate outputs aligned with specific themes or styles.

4. Preprocessing:

Text cleaning included removing HTML tags, stopwords, and punctuations. Tokenization and vectorization were performed using BERT embeddings to capture semantic context.

5. Model Training- BERT:

BERT is used here to extract embeddings (dense numerical representations) of news articles. It captures contextual nuances of language in the input text by analyzing all words in both directions (left-to-right and right-to-left). It also generates representations that are particularly useful for classification tasks like real vs. fake news detection.

6. Evaluation:

Metrics such as accuracy, precision, recall, and F1-score were computed. Confusion matrices were used for detailed error analysis.

In a crux, the following steps are being done :

1. Import Dependencies & Libraries
2. Create our Dataset
3. Incorporate New Data Articles:
   - API: real news articles
   - GPT: Fake news articles
4. Data Cleaning
5. Model Training: Classification by BERT Model
6. Evaluating

OBSERVATION

1. DATA COLLECTION:

	A	B	C	D
1	AUTHOR	TITLE	CONTENT	LABEL
2	Jake Hoffman	13 US states set to have women governors next year, a record	For the first time, 13 US states will be led by women governors in 2025, following Kelly Ayotte's victory i	0
3	Alex Monroe	Scientists Claim to Have Found a Cure for Aging	Scientists from a secret lab claim they have discovered a method to reverse aging. The study shows pa	1
4	Sarah Kline	Mars Rovers Uncover Alien Civilization Evidence	Recent Mars rover missions have allegedly found remnants of an alien civilization. Images show artifa	1
5	Tim Rogers	Time Traveler Arrested for Insider Trading	A man claiming to be from 2075 was reportedly arrested for using future knowledge in stock trading. D	1
6	Jamie Patel	New Energy Drink Gives Superhuman Strength	A recently launched energy drink is said to provide superhuman strength, allowing users to lift heavy o	1
7	Lauren Chen	Ancient Egyptian Tomb Reveals Modern Technology	Archaeologists claim a tomb in Egypt contains a 5,000-year-old device resembling a laptop. Some bel	1
8	Daniel Ortiz	Lost City Found Beneath Antarctic Ice	Researchers claim to have found a lost city under Antarctica, with structures visible in ice-penetrating	1
9	Priya Das	Government to Release UFO Documents Next Week	Officials reportedly plan to release documents confirming UFO sightings and encounters. The files all	1
10	Carlos Gomez	Underwater Pyramid Discovered Near Bermuda	Divers have reportedly discovered a pyramid-like structure near the Bermuda Triangle. Some believe i	1
11	Emily Tang	Elusive Bigfoot Spotted on Security Camera	Security footage allegedly captures Bigfoot roaming a remote forest area, stirring excitement among c	1
12	Victor Ivanov	Global Elites Plotting Mars Colony for Themselves	A whistleblower claims global elites plan to establish a Mars colony to escape Earth's crises. Docume	1
13	John Smith	Aliens Spotted in Central Park	An amateur astronomer claims to have seen UFOs flying over Central Park last night. The strange light	1
14	Jane Doe	Chocolate Proven to Extend Lifespan by 20 Years	Researchers from an unknown institute announced a study claiming that eating chocolate daily can e:	1
15	Alex Turner	Dinosaurs Alive in Remote Amazon Jungle	Adventurers in the Amazon claim to have spotted dinosaur-like creatures in unexplored areas. Photos	1
16	Emily Clark	City Announces Plan to Control Weather for Perfect Summers	City officials have reportedly created a new technology to control weather patterns, promising endles	1

```
[ ]  from google.colab import files
     uploaded = files.upload()

     Choose files  No file chosen      Upload widget is only available when the cell has been
     Saving News - Sheet1.csv to News - Sheet1.csv
```

Building a custom dataset for the project demonstrates a tailored approach that aligns with our specific requirements. Below is a detailed explanation of our dataset creation process, emphasizing its structure, sources, and advantages:

The following four columns were made to capture essential attributes of each news entry:

a. **Author Name:** Represents the individual or organization credited with writing the article.
b. **Title:** The headline or title of the news article.
c. **Content**: The main body of the article.
d. **Label:** Indicates whether the article is real or fake. A binary value that serves as the target variable for training your classification model.

1. Real News Articles

SOURCE:
Real news entries were gathered using snippets available online, such as those from reputable news websites, online publications, and archives.
PROCESS:
- Selected credible sources known for their journalistic integrity (e.g., well-known newspapers, magazines, and verified online portals).
- Copied titles, content, and author names from real articles, ensuring they represented diverse topics (e.g., politics, health, technology, and sports).
- Organized and cleaned the entries to maintain consistency, removing irrelevant data such as advertisements or unrelated content.

ADVANTAGES:
- Guarantees that the real news entries are authentic and accurate, providing a strong foundation for training the model.
- Ensures coverage of various writing styles and tones, enhancing the model's ability to generalize across real-world examples.

2. Fake News Articles

SOURCE:
Fake news entries were procured from ChatGPT, a generative AI model capable of producing coherent and realistic-looking fake news articles.
PROCESS:
ChatGPT was prompted to generate fake news articles on diverse topics, ensuring coverage of common fake news themes, such as conspiracy theories, sensational claims, and misleading information.
ADVANTAGES:
- Provides realistic yet fabricated articles that mimic the language and style of genuine news, making them challenging for the model to distinguish.
- Highlights the subtle cues of fake news, such as sensationalism, emotional tone, and lack of credible sources.

2. API INTEGRATION:

```python
import requests
import pandas as pd

def fetch_real_news(api_key, query="latest news", page_size=50):
    url = f"https://newsapi.org/v2/everything?q={query}&pageSize={page_size}&apiKey={api_key}"
    response = requests.get(url)
    if response.status_code == 200:
        data = response.json()
        articles = [
            {
                "title": article.get("title", "No Title"),  # Fetch title or fallback to "No Title"
                "content": article.get("content", "No Content"),  # Fetch content or fallback to "No Content"
                "author": article.get("author", "Unknown Author"),  # Fetch author or fallback to "Unknown Author"
                "label": 0  # Real news
            }
            for article in data.get("articles", [])
        ]
        return pd.DataFrame(articles)
    else:
        print("Error:", response.status_code, response.text)
        return pd.DataFrame()

# Replace 'your_api_key' with your actual API key
api_key = 'fab6126bbff14e988861b5728a0a203e'
new_real_news = fetch_real_news(api_key, query="technology", page_size=50)

# Display the first few rows
print("New Real News Articles Fetched Successfully:")
print(new_real_news.head())
```

1. Data Augmentation and Dataset Expansion

The API is used to generate additional real news articles, contributing to a larger and more diverse dataset. This serves multiple purposes:

Fetching Real News Articles:

- The API fetches or generates legitimate news articles by leveraging trusted external sources or AI models like GPT. These articles add variety to the dataset by covering different topics, writing styles, and perspectives.

2. Simulating Real-World Scenarios

The API enables the creation of articles that replicate real-world scenarios. By incorporating these scenarios, the model learns to recognize the nuanced patterns of fake news that mimic legitimate sources.

3. Dynamic Dataset Updates

APIs facilitate the dynamic updating of your dataset by:

- Regularly fetching new real news articles from trusted sources (e.g., RSS feeds or news APIs like NewsAPI).
- Prompting GPT or similar models to create new fake news samples in real-time

4. Introducing Variety in Data

APIs, particularly those integrated with GPT-based models, can generate highly diverse text samples. This variety prevents the model from learning only surface-level features and helps it focus on deeper contextual and semantic cues.

5. Supporting Real-Time Predictions

Beyond dataset generation, the API can enhance the user interaction component by supporting real-time predictions:

- Users input a news article via the system interface.
- The API processes the input by communicating with the machine learning model to provide instant feedback (real or fake).
- This setup ensures seamless and scalable predictions without exposing the internal workings of the model.

6. Enabling Synthetic Data Generation

Using GPT-powered APIs, you can generate synthetic data tailored to specific needs:

- Controlled Prompts: APIs allow you to provide precise prompts to create targeted content (e.g., "Write a fake news article on a breakthrough scientific discovery").
- Custom Variations: Generate variations of a single news article by tweaking prompts, helping the model learn to identify subtle changes that might indicate fake news.
- Balancing the Dataset: APIs can be instructed to create an equal number of real and fake articles, ensuring class balance in the training dataset.

3. GPT INTEGRATION

```
[ ] from transformers import GPT2LMHeadModel, GPT2Tokenizer

    model_name = "gpt2"  # You can also use "EleutherAI/gpt-neo-1.3B" for larger models
    tokenizer = GPT2Tokenizer.from_pretrained(model_name)
    model = GPT2LMHeadModel.from_pretrained(model_name)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens),
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100%  26.0/26.0 [00:00<00:00, 1.88kB/s]
vocab.json: 100%  1.04M/1.04M [00:00<00:00, 3.97MB/s]
merges.txt: 100%  456k/456k [00:00<00:00, 7.33MB/s]
tokenizer.json: 100%  1.36M/1.36M [00:00<00:00, 4.51MB/s]
config.json: 100%  665/665 [00:00<00:00, 39.9kB/s]
model.safetensors: 100%  548M/548M [00:04<00:00, 123MB/s]
generation_config.json: 100%  124/124 [00:00<00:00, 9.20kB/s]
```

1.Synthetic Data Generation

- This expands our dataset with a variety of writing styles, topics, and formats.
- Balances the dataset by creating equal amounts of real and fake news, especially when labeled real-world data is scarce.
- Introduces edge cases and variations, making your model more robust to diverse inputs.

2. Contextual Understanding During Data Augmentation

GPT can be fine-tuned or prompted to generate contextually diverse fake news articles. For example, generating fake news that mimics credible sources or sensational headlines.

Text Coherence: Produces high-quality articles that resemble real news, making the detection task more challenging.
Prompt Flexibility: Can generate outputs aligned with specific themes or styles (e.g., political fake news, health misinformation).

4. PRE-PROCESSING

```
[ ] # Rename columns in the new dataset
    new_real_news.rename(
        columns={
            "title": "TITLE",
            "content": "CONTENT",
            "author": "AUTHOR",
            "label": "LABEL"
        },
        inplace=True
    )
    print("Renamed New Real News Columns:")
    print(new_real_news.columns)

    Renamed New Real News Columns:
    Index(['TITLE', 'CONTENT', 'AUTHOR', 'LABEL'], dtype='object')
```

```
# Merge the datasets
combined_data = pd.concat([existing_data, new_real_news], ignore_index=True)

# Save the combined dataset to a CSV file
combined_data.to_csv("Updated_News_Dataset.csv", index=False)
print("Combined Dataset Saved Successfully.")

Combined Dataset Saved Successfully.
```

```
# Ensure the 'text' column is of string type and handle batches correctly
train_dataset = train_dataset.map(lambda x: {"text": [str(text) for text in x["text"]]}, batched=True)
test_dataset = test_dataset.map(lambda x: {"text": [str(text) for text in x["text"]]}, batched=True)

# Check for and remove any rows where 'text' is NaN or empty
train_dataset = train_dataset.filter(lambda x: x["text"] != "" and x["text"] is not None)
test_dataset = test_dataset.filter(lambda x: x["text"] != "" and x["text"] is not None)

# Retry tokenization
train_tokenized = train_dataset.map(tokenize_function, batched=True)
test_tokenized = test_dataset.map(tokenize_function, batched=True)
```

To get the final dataset, we add the fake news articles generated by API and the real news articles generated by GPT, to our original dataset. After we have made our final dataset, some data cleaning is performed as shown above. Since, the column names in the news generated by API and GPT differed from the ones in our dataset, column renaming is done.

Lastly, the dataset is divided into training set and testing set to further train the model.

5. BERT

```
[ ] from transformers import BertTokenizer

    # Load the BERT tokenizer
    tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

    # Tokenize the datasets
    def tokenize_function(examples):
        return tokenizer(examples["text"], padding="max_length", truncation=True)

    train_tokenized = train_dataset.map(tokenize_function, batched=True)
    test_tokenized = test_dataset.map(tokenize_function, batched=True)
```

```
tokenizer_config.json: 100%  48.0/48.0 [00:00<00:00, 1.05kB/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 4.12MB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 7.16MB/s]
config.json: 100%  570/570 [00:00<00:00, 16.2kB/s]
```

1. Feature Extraction for Classification

BERT is a bidirectional language model optimized for understanding the context of a given input text. In your project, BERT is used to extract embeddings (dense numerical representations) of news articles.

- Captures contextual nuances of language in the input text by analyzing all words in both directions (left-to-right and right-to-left).
- Generates representations that are particularly useful for classification tasks like real vs. fake news detection.

2. Fine-Tuning for Binary Classification

After extracting embeddings, BERT can be fine-tuned with labeled datasets for the specific task of classifying news as real or fake.

- It tailors BERT to your dataset, improving its performance for this specific task.
- Maintains pre-trained knowledge while adapting to domain-specific nuances in news articles.

Bidirectional Context: Provides a deeper understanding of sentence-level and document-level semantics.

Pre-Trained Knowledge: Leverages a vast corpus of general language knowledge for effective feature extraction.

6. EVALUATION

```
[ ] from sklearn.metrics import accuracy_score

    # Calculate accuracy
    accuracy = accuracy_score(predictions.label_ids, predicted_labels)
    print(f"Test Accuracy: {accuracy:.4f}")
```

```
Test Accuracy: 0.9794
```

```
from sklearn.metrics import precision_score, recall_score, f1_score, confusion_matrix

# Precision, Recall, F1-Score
precision = precision_score(predictions.label_ids, predicted_labels)
recall = recall_score(predictions.label_ids, predicted_labels)
f1 = f1_score(predictions.label_ids, predicted_labels)

print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")

# Confusion Matrix
cm = confusion_matrix(predictions.label_ids, predicted_labels)
print(f"Confusion Matrix:\n{cm}")
```

```
Precision: 1.0000
Recall: 0.9615
F1-Score: 0.9804
Confusion Matrix:
[[45  0]
 [ 2 50]]
```

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

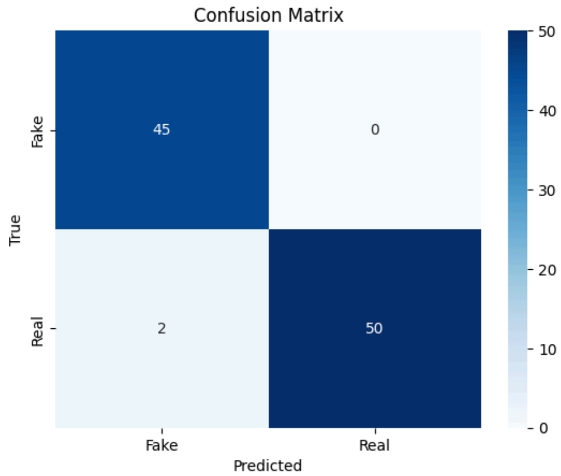

Fig 2: Creating the Confusion Matrix from the parameters obtained.

CONCLUSION & FUTURE WORK

There are a few limitations to this model we have designed:
· Dependence on the quality of synthetic data generation.
· Computational overhead during training due to BERT’s complexity.
· Limited performance for multilingual fake news articles.

Apart from those, the project successfully developed a scalable fake news detection system.

Future work includes expanding the dataset to include multilingual articles, integrating more advanced transformers like GPT-4, and enhancing the user interface with interactive visualizations.
We can also future prospects where we can add ChatGPT here also, to generate fake news articles to test them simultaneously.

BIBLIOGRAPHY

1. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

2. Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., & Choi, Y. (2019). Defending Against Neural Fake News. Advances in Neural Information Processing Systems.

3. Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

4. Shaik, S., Nelli, S. (2023). Chat Analysis and Spam Detection of WhatsApp using Machine Learning. ResearchGate.

5. Python 3.7 Documentation. Available at: https://docs.python.org/3.7/

6. Streamlit Documentation. Available at: https://docs.streamlit.io/

7. WordCloud for Python Documentation. Available at: https://amueller.github.io/word_cloud/

THANK YOU!