FAKE NEWS DETECTION USING NLP

Problem Statement :

Develop a fake news detection system using Natural Language Processing (NLP) to accurately identify and categorize news articles or information as either real or fake, contributing to the mitigation of misinformation and its harmful effects on society.

Design Thinking Process :

1. **Empathize:**
    a. Understand the impact of fake news on society.
    b. Identify the needs of end-users and stakeholders in combating fake news.
2. **Define:**
    a. Define the specific goals of the fake news detection project.
    b. Clarify the types of fake news to target (e.g., text-based, image-based).
3. **Ideate:**
    a. Brainstorm NLP techniques to detect fake news (e.g., sentiment analysis, text classification)Explore data sources and collection methods.
4. **Prototype:**
    a. Develop NLP models and algorithms for fake news detection.
    b. Create a prototype of the fake news detection system.
5. **Test:**
    a. Evaluate the performance of NLP models using labeled datasets.
    b. Collect feedback from users and refine the detection system.
6. **Iterate:**
    a. Improve the NLP models based on testing results.

b. Continuously update the system to adapt to evolving fake news techniques.

7. **Implement:**

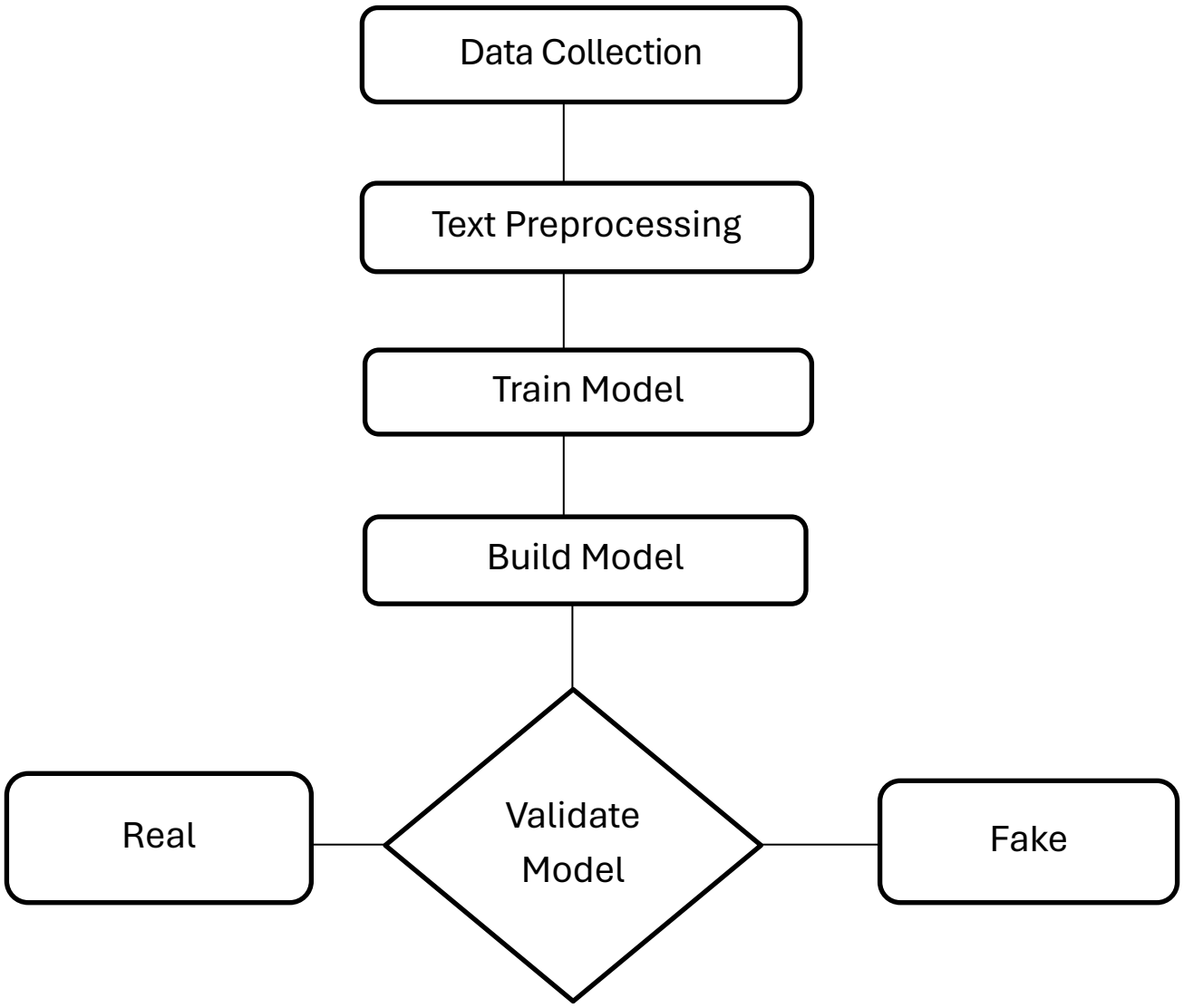
   a. Deploy the final fake news detection system in real-world applications.

   b. Monitor its performance and gather user feedback for ongoing improvements.

8. This iterative design thinking process ensures that the fake news detection system remains effective and responsive to emerging challenges.

INNOVATION

To implement this, collecting the fake news data set from the kaggle .



1. **Data Collection**: In the first step, you gather a diverse and representative dataset of news articles. This dataset should include both reliable, authentic news articles and fake, deceptive ones. The accuracy and performance of your fake news detection model heavily depend on the quality and quantity of the data collected. A larger and more balanced dataset helps the model learn the distinguishing features between real and fake news.

2. **Text Preprocessing**: Text preprocessing is essential to clean and prepare the data for analysis. This step involves tasks like removing HTML tags, special characters, and punctuation, tokenization (splitting the text into words or tokens), converting text to lowercase, and stemming or lemmatization to standardize words. Additionally, stop words (common words like "the" or "and") may be removed to reduce noise in the dataset. Text preprocessing ensures that the text data is in a suitable format for NLP analysis.

3. **Train Model**: In this step, you split your preprocessed dataset into training and testing sets. The training data is used to teach the machine learning model the patterns and characteristics of both real and fake news articles. Various NLP techniques are applied here, such as feature extraction, vectorization, and the creation of word embeddings using methods like Word2Vec or TF-IDF. The model, which can be a deep neural network, a support vector machine, or other algorithms, learns to distinguish between real and fake news based on these features.

4. **Building Model**: The model-building phase involves selecting and implementing a machine learning algorithm. This algorithm is trained on the preprocessed data and learns to classify news articles as real or fake based on the features extracted during the training phase. You can fine-tune the model's parameters and architecture to improve its performance, and you may also consider techniques like ensembling or deep learning architectures for enhanced accuracy.

5. **Validate Model**: Once the model is built, it needs to be evaluated for its performance. This is done by testing the model on a separate validation dataset (different from the training data). You can use various evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to assess how well the model is at detecting fake news. If the model performs well on the validation dataset, it can then be deployed for real-time fake news detection.

Real News Sample Head

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

Fake News Sample Head

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

Loading the Dataset:

Downloaded the **train.csv** dataset from the Kaggle

https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

We loaded a train.csv dataset using the **pandas** library.

```
[8]: import pandas as pd

[9]: dataFrame = pd.read_csv('train.csv')
     dataFrame.head()
```

[9]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

Text Preprocessing:

Replacing null values in the dataset with empty string

```
[5]: dataFrame.isnull().sum()

[5]: id          0
     title     558
     author   1957
     text       39
     label       0
     dtype: int64

[6]: #to remove the null values with an empty string
     dataFrame = dataFrame.fillna('')

[7]: dataFrame.isnull().sum()

[7]: id        0
     title     0
     author    0
     text      0
     label     0
     dtype: int64
```

Removing Unnecessary Columns in the dataFrame

```
[8]: #remove unnecessary columns
     dataFrame.drop(['id','title','author'],axis=1)
```

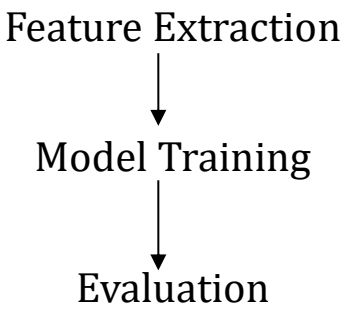
[8]:

	text	label
0	House Dem Aide: We Didn't Even See Comey's Let...	1
1	Ever get the feeling your life circles the rou...	0
2	Why the Truth Might Get You Fired October 29, ...	1
3	Videos 15 Civilians Killed In Single US Airstr...	1
4	Print \nAn Iranian woman has been sentenced to...	1
...	...	...

Using **Natural Language ToolKit** Library for Text Preprocessing

This code processes the 'text' column in the DataFrame by applying text preprocessing operations like stemming, lowercase conversion, and stopword removal to each text entry in that column.

```
[9]: import nltk
     nltk.download('stopwords')
     from nltk.corpus import stopwords
     from nltk.stem.porter import PorterStemmer

     import re

     [nltk_data] Downloading package stopwords to
     [nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!

[10]: portStem = PorterStemmer()

[11]: def stemming(content):
          con = re.sub('[^a-zA-Z]',' ',content)
          con = con.lower()
          con = con.split()
          con = [portStem.stem(word) for word in con if not word in stopwords.words('english')]
          con = ''.join(con)
          return con

[*]: dataFrame['text'] = dataFrame['text'].apply(stemming)
```



Feature Extraction:

It involves tasks such as removing non-alphabetic characters, converting text to lowercase, splitting it into words, removing **stopwords**, and applying **stemming**. These preprocessing steps are essential for preparing text data for classification or other NLP tasks..

```
[17]: def stemming(content):
          con=re.sub('[^a-zA-Z]', ' ', content)
          con=con.lower()
          con=con.split()
          con=[port_stem.stem(word) for word in con if not word in stopwords.words('english')]
          con=' '.join(con)
          return con
```

```
In [37]: df['text']= df['text'].apply(stemming)

In [38]: x=df['text']

In [39]: y=df['label']
```

Model Training:

Training the Model

1. **train_test_split(x, y, test_size=0.25)**: This function is used to split your dataset x and corresponding labels y into training and testing sets. The test_size parameter specifies the proportion of the dataset that should be allocated to the testing set (in this case, 25% of the data). It returns x_train, x_test, y_train, and y_test.

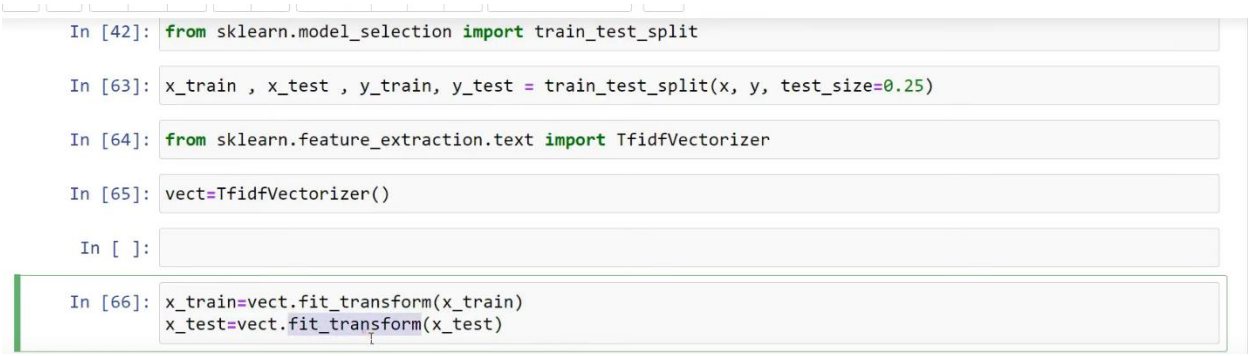
2. **from sklearn.feature_extraction.text import TfidfVectorizer**: This line imports the **TfidfVectorizer** class from scikit-learn (a popular machine learning library). **TfidfVectorizer** is used for text data preprocessing and feature extraction, particularly for

3. converting text into numerical features using the TF-IDF (Term Frequency-Inverse Document Frequency) method.

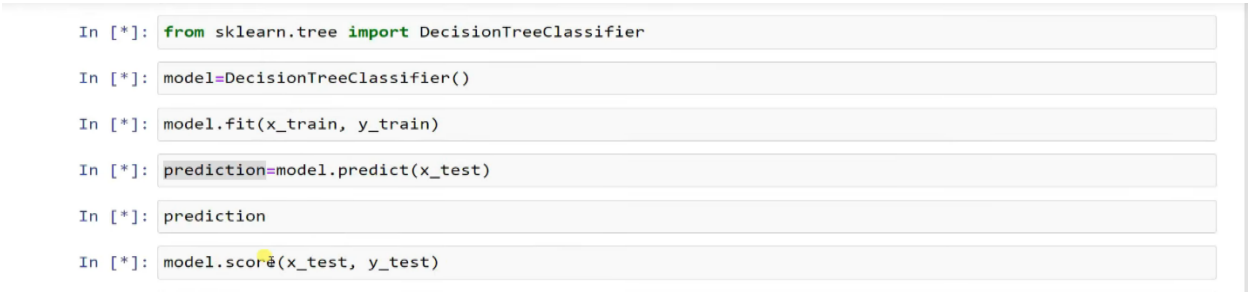
4. **vect = TfidfVectorizer():** This line creates an instance of the TfidfVectorizer class. vect will be used to transform your text data into TF-IDF features.

5. **x_train = vect.fit_transform(x_train):** It applies the TF-IDF transformation to the training data x_train, converting the text data into numerical features. This will replace the original x_train with the transformed data.

6. **x_test = vect.fit_transform(x_test):** Similarly, it applies the TF-IDF transformation to the testing data x_test, converting it into numerical features. However, this line should be corrected to use transform instead of fit_transform. So, it should be x_test = vect.transform(x_test) to ensure that the testing data is transformed using the same TF-IDF settings as the training data.



Evaluation:



1. **from sklearn.tree import DecisionTreeClassifier**: This line imports the Decision Tree Classifier class from scikit-learn. A decision tree classifier is a type of machine learning model that makes decisions by learning simple decision rules inferred from the training data.

2. **model = DecisionTreeClassifier():** This creates an instance of the Decision Tree Classifier model.

3. **model.fit(x_train, y_train):** It trains the Decision Tree Classifier using the training data x_train and corresponding labels y_train. The model learns to make predictions based on this training data.

4. **prediction = model.predict(x_test):** This line makes predictions on the testing data x_test using the trained decision tree model and stores the predictions in the prediction variable.

5. **model.score(x_test, y_test):** This calculates the accuracy of the trained model on the testing data. It compares the model's predictions (based on x_test) to the actual labels y_test and returns the accuracy score, which represents the proportion of correctly classified instances in the testing data.

In fake news detection with NLP, Pickle can be used to save and load machine learning models trained to classify news articles as real or fake. This allows for efficient model storage and reuse, making it practical to deploy and maintain the fake news detection system.

```
[53]: import pickle

[54]: pickle.dump(vect, open('vector.pkl', 'wb'))

[56]: pickle.dump(model, open('model.pkl', 'wb'))

[57]: vector_form=pickle.load(open('vector.pkl', 'rb'))

[58]: load_model=pickle.load(open('model.pkl', 'rb'))

[ ]:
```

Writing function to detect a news is fake or not

```
[59]: def fake_news(news):
          news=stemming(news)
          input_data=[news]
          vector_form1=vector_form.transform(input_data)
          prediction = load_model.predict(vector_form1)
          return prediction
```

```
[60]: val=fake_news("""In these trying times, Jackie Mason is the Voice of Reason. [In this week's exclusive clip for Breitbart News, Jackie discusses the loom
```

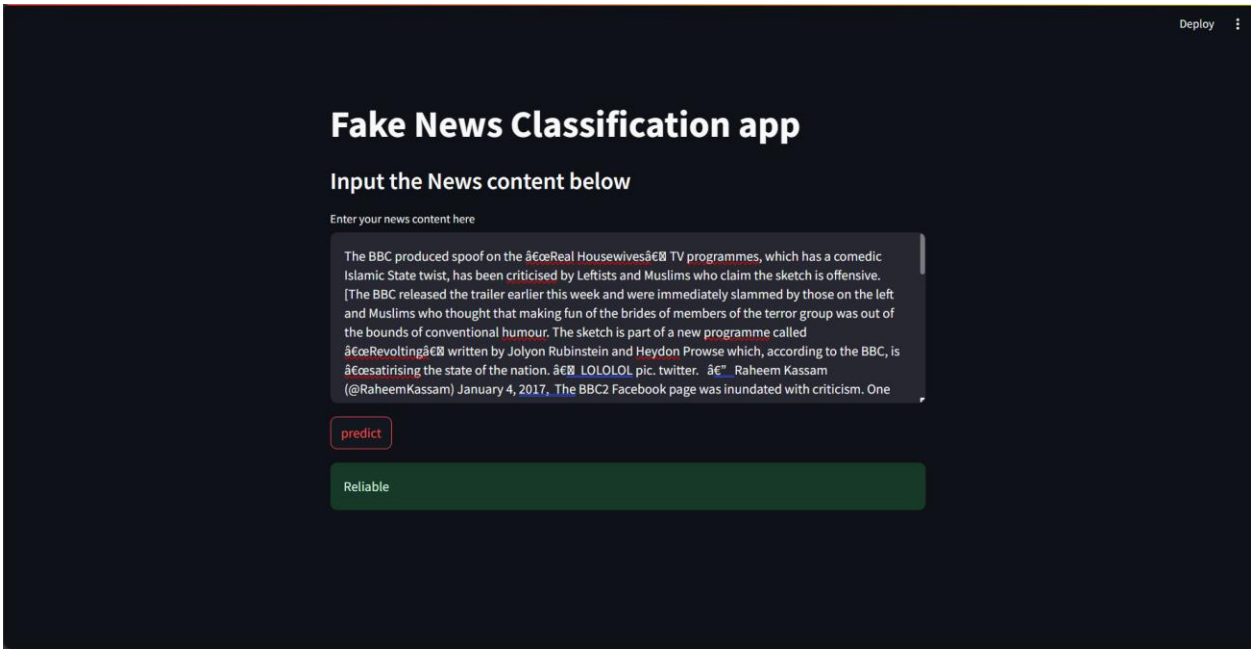
```
[61]: if val==[0]:
          print('reliable')
      else:
          print('unreliable')
```

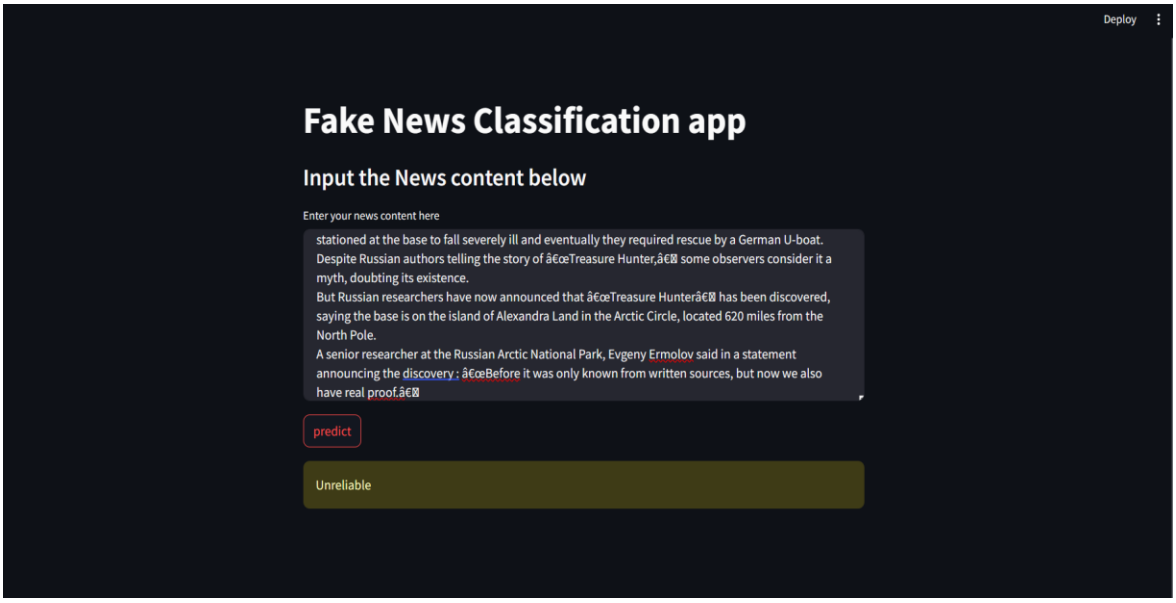
```
reliable
```

Using Streamlit Library to Deploy

Streamlit is an open-source Python library used for creating web applications with minimal effort. It allows developers and data scientists to turn data scripts into interactive web apps quickly. Streamlit is known for its simplicity and ease of use, as it doesn't require extensive web development knowledge. Users can build data dashboards, visualizations, and interactive tools by writing Python code in a straightforward manner. It's a popular choice for prototyping and deploying data-driven applications.

Streamlit can be used in fake news detection projects with NLP to create interactive web applications for users. It enables real-time testing of the NLP model, making it user-friendly and accessible. Users can input news articles, receive instant results, and better understand the model's performance. Streamlit simplifies the deployment and visualization of the NLP-based fake news detection system, enhancing user engagement and usability.





Conclusion:

In conclusion, a fake news detection project using NLP is a valuable application of natural language processing techniques to combat the spread of misinformation. By leveraging NLP models and techniques, we can analyze and classify news articles accurately. Implementing Streamlit for creating user-friendly web interfaces enhances the project's usability and accessibility. With the right tools and technology, we can contribute to a more informed and trustworthy information landscape, mitigating the impact of fake news.

Team:

MOHAMED ABDUL RAHMAN S (au951221104023)

YABESH JESILEN V (au951221104059)

SAM SELIN D (au951221104039)

RAHUL B (au951221104034)

NAGA MARI MUTHU A (au951221104028)