

**SARANATHAN COLLEGE OF ENGINEERING**

**Venkateshwara Nagar,Panjappur,Trichy-12**

**IBM-NAALAYA THIRAN**

**(project based experiential learning program)**

PROJECT TITLE -- Fake News Detecting Using NLP

DOMAIN -- Artificial Intelligence

SUBJECT CODE -- SB3001

PROJECT ID -- Proj\_203471\_Team\_4

TEAM LEADER

MUTHAMIL S -- [saravananmuthamil@gmail.com](mailto:saravananmuthamil@gmail.com)

(au813821106071)

TEAM MEMBERS

PRIYANKA K -- [priyankakumar7641@gmail.com](mailto:priyankakumar7641@gmail.com)

(au813821106085)

SWETHA S -- [swetha311203@gmail.com](mailto:swetha311203@gmail.com)

(au813821106119)

SHANALIN J -- [shanalinpkt1607@gmail.com](mailto:shanalinpkt1607@gmail.com)

(au813821106104)

GITHUB: https://github.com/Muthamil1510/fake\_news\_detection\_phase1\_IBM\_au813821106071.git

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TABLE OF CONTENT** | **PG.NO** |
| 1 | **INTRODUCTION** | 1 |
| 1.1 | PROJECT OVERVIEW | 1 |
| 1.2 | PURPOSE | 1 |
| 2 | **LITERATURE SURVEY** | 4 |
| 2.1 | EXISTING PROBLEM | 4 |
| 2.2 | REFERENCE | 6 |
| 2.3 | PROBLEM STATEMENT | 7 |
| 3 | **IDEATION AND PROPOSED SOLUTION** | 8 |
| 3.1 | EMPATHY MAP CANVAS | 8 |
| 3.2 | IDEATION AND BRAINSTORMING | 9 |
| 3.3 | PROPOSED SOLUTION | 10 |
| 3.4 | PROBLEM SOLUTION | 11 |
| 4 | **PROJECT DESIGN** | 12 |
| 4.1 | DATA FLOW DIAGRAM | 12 |
| 4.2 | SOLUTION AND TECHNICAL  ARCHITECTURE | 15 |
| 5 | **PROJECT PLANNING AND SCHEDULE** | 16 |
| 5.1 | SPRINT DELIVERY SCHEDULE | 16 |
| 5.2 | SPRINT PLANNING AND ESTIMATION | 17 |
| 5.3 | REPORT FROM JIRA | 18 |
| 6 | **RESULTS** | 19 |
| 7 | **ADVANTAGES** | 20 |
| 8 | **DISADVANTAGES** | 21 |
| 9 | **CONCLUSION** | 22 |
| 10 | **FUTURE SCOPE** | 23 |
| 11 | **APPENDIX** | 24 |

**Abstract:**

With the proliferation of digital media and the ease of information dissemination on the internet, the spread of fake news has become a major concern in today's society. Fake news can not only misinform the public but also have significant social and political consequences. To combat this issue, this research focuses on the development of a robust and effective fake news detection system using Natural Language Processing (NLP) techniques. This study leverages the power of NLP to analyze and classify textual content as either genuine or fake news. The methodology involves a multi-step process, including data collection, preprocessing, feature extraction, and machine learning model implementation. A diverse and extensive dataset of both real and fake news articles is collected to train and test the model, ensuring its adaptability to various forms of misinformation. Key NLP techniques such as tokenization, text vectorization, and word embeddings are employed to convert the textual data into numerical representations, making it suitable for machine learning models. Feature engineering is used to identify linguistic patterns and characteristics that differentiate fake news from genuine news articles. Various machine learning algorithms, such as support vector machines, recurrent neural networks, and transformer models, are employed to build and evaluate the detection system's performance. The model is trained on labeled data to classify news articles as authentic or deceptive based on the extracted features and linguistic cues. The results of this study demonstrate the effectiveness of NLP-based techniques in identifying fake news with high accuracy and efficiency. The proposed system offers a valuable tool for media organizations, fact-checking platforms, and social media networks to combat the dissemination of misinformation and safeguard the integrity of the information ecosystem. As fake news continues to pose a threat to the credibility of information sources, the development of robust NLP-based detection systems becomes increasingly vital in maintaining a well-informed Society.

**1)Introduction**

* 1. **Project Overview:**

In an age of digital information overload, the spread of fake news and misinformation has become a critical issue. This project aims to develop Fake News Detection system using Natural Language Processing (NLP) techniques. The objective is to create a model that can automatically identify and flag news articles, social media posts, or any textual content that contains false or misleading information.

* 1. **Purpose:**

**1. Preserving Information Integrity:** The primary purpose is to preserve the integrity of information in the digital age. Fake news, misinformation, and disinformation can have serious consequences, ranging from influencing public opinion and causing social unrest to impacting financial markets and public health. Fake news detection using NLP aims to identify and mitigate the spread of false or misleading information.

**2.Enhancing Media Literacy:** Fake news detection tools can also serve an educational purpose. By providing users with the means to check the authenticity of news articles and content, it promotes media literacy. Users can learn to be more critical consumers of information and better equipped to distinguish between credible and questionable sources.

**3.Aiding Fact-Checking Organizations:** Fact-checking organizations and journalists can benefit from NLP-based tools to expedite the process of identifying false claims and verifying information. These tools can assist in more efficient fact-checking, especially when dealing with large volumes of information.

**4.Promoting Accountability:** Fake news detection can contribute to holding publishers and content creators accountable for the accuracy of their information. When fake news is identified and exposed, it can lead to greater transparency and ethical reporting practices.

**5.Supporting Policymaking:** In the context of public policy and governance, fake news can influence decision-making and public sentiment. By detecting and addressing fake news, NLP tools can help policymakers make more informed decisions and mitigate potential crises.

**6.Ensuring Ethical AI and NLP Development:** As NLP models and AI systems become more influential in the information ecosystem, it is essential to ensure that they are used responsibly and ethically. Fake news detection using NLP underscores the importance of creating AI systems that prioritize accuracy, fairness, and transparency.

**7.Preserving Trust in Information Sources**: Trust in information sources is crucial for a functioning democracy and an informed society. Detecting and exposing fake news can help maintain trust in news outlets, social media platforms, and other sources of information.

**8.Mitigating the Spread of Disinformation:** In times of crisis, such as a pandemic or natural disaster, the rapid spread of false information can have

direct consequences. NLP-based fake news detection can play a critical role in minimizing the dissemination of disinformation that may cause panic or harm.

In **summary**, the purpose of fake news detection using NLP is to safeguardthe reliability of information, foster media literacy, and address the challengesposed by misinformation and disinformation in our digital world. It servesboth as a preventive measure to combat the spread of fake news and as atool for improving the overall quality of information consumption and dissemination.

# Fake news detection using NLP is a complex and ongoing field of research and development, as the methods and technologies used to spread misinformation continue to evolve. It plays a crucial role in combating the spread of false information and ensuring the accuracy and credibility of news and information sources.

**2)Literature Survey**

**2.1) Existing Problem:**

Fake news detection is a challenging and evolving field, and there are several existing problems and limitations associated with it:

**Data Quality and Availability:** Limited availability of labeled datasets for training, especially in languages other than English, hampers the development of accurate fake news detection models. Additionally, the quality and diversity of the available data can significantly impact the model's performance.

**Rapidly Evolving Techniques:** Those spreading fake news continually adapt their tactics, making it difficult for detection algorithms to keep up. New forms of deception, such as deepfakes and sophisticated misinformation campaigns, pose significant challenges to traditional detection methods.

**Context and Satire:** Fake news often relies on satire or altered context, making it difficult for algorithms to distinguish between genuine satire and actual misinformation. Understanding nuanced linguistic and cultural cues is a complex task for automated systems**.**

**Biases in Data:** Training data can contain biases present in the sources from which it's collected. Biased data can lead to biased models, potentially misclassifying certain types of news based on the biases present in the training data.

**Adversarial Attacks:** Adversarial attacks involve intentionally manipulating the input data to deceive machine learning models. Fake news creators can employ these techniques to evade detection systems, making them less effective in real-world scenarios.

**Multimodal Challenges:** With the rise of multimedia content, detecting fake news in the form of images and videos becomes a more intricate problem. Analyzing visual and auditory cues alongside textual information is a complex task and requires sophisticated multimodal models.

**Explainability:** Deep learning models, particularly neural networks, are often considered black boxes, making it difficult to explain their decisions. Interpretable models are crucial for building trust, especially in applications where the consequences of misclassification are significant.

**Misinformation Spread Dynamics:** Understanding how misinformation spreads through social networks and how it evolves over time is vital. Real-time analysis of social media interactions and identifying influential nodes in the network is challenging but essential for effective detection and mitigation.

**Legal and Ethical Challenges:** Determining the boundaries between free speech, satire, and misinformation raises legal and ethical questions. Striking a balance between addressing misinformation and preserving freedom of expression is a complex societal challenge.

**Resource Constraints:** Small organizations or fact-checking initiatives might lack the resources to implement and maintain advanced fake news detection systems, limiting the widespread application of effective detection techniques.

**2.2) Reference:**

1)"Fake News Detection on Social Media: A Data Mining Perspective" by Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2017). This paper discusses various data mining and NLP techniques for detecting fake news on social media.

2)"Fake News Detection on Twitter: A Data Mining Approach “by Ruchasky, N., Seo, S., & Liu, Y. (2017). This paper focuses on fake news detection specifically, on Twitter and highlights the importance of linguistic and network features.

3) "A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities" by Zubiaga, A., Liakata, M., Procter, R., Bontcheva, K., & Tolmie, P. (2018). This survey paper provides a comprehensive overview of the various aspects of fake news, including detection methods and opportunities.

4) "BERT for Detecting Fake News by Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). This paper demonstrates how the BERT (Bidirectional Encoder Representations from Transformers)model can be used for fake news detection

5)"Hierarchical Attention Network for Fake News Detection"by Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.& Hovy, E. H. (2016). This paper introduces a hierarchical attention network for detecting fake news, focusing on the attention mechanisms.

**2.3)Problem Statement:**

The problem is to develop a fake news detection model using a Kaggle dataset. The goal is to distinguish between genuine and fake news articles based on their titles and text. This project involves using natural language processing (NLP) techniques to preprocess the text data, building a machine learning model for classification, and evaluating the model's performance.

**Data Source**: Choose the fake news dataset available on Kaggle, containing articles titles and text, along with their labels (genuine or fake).

**Data Preprocessing:** Clean and preprocess the textual data to prepare it for analysis.

**Feature Extraction:** Utilize techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings to convert text into numerical features.

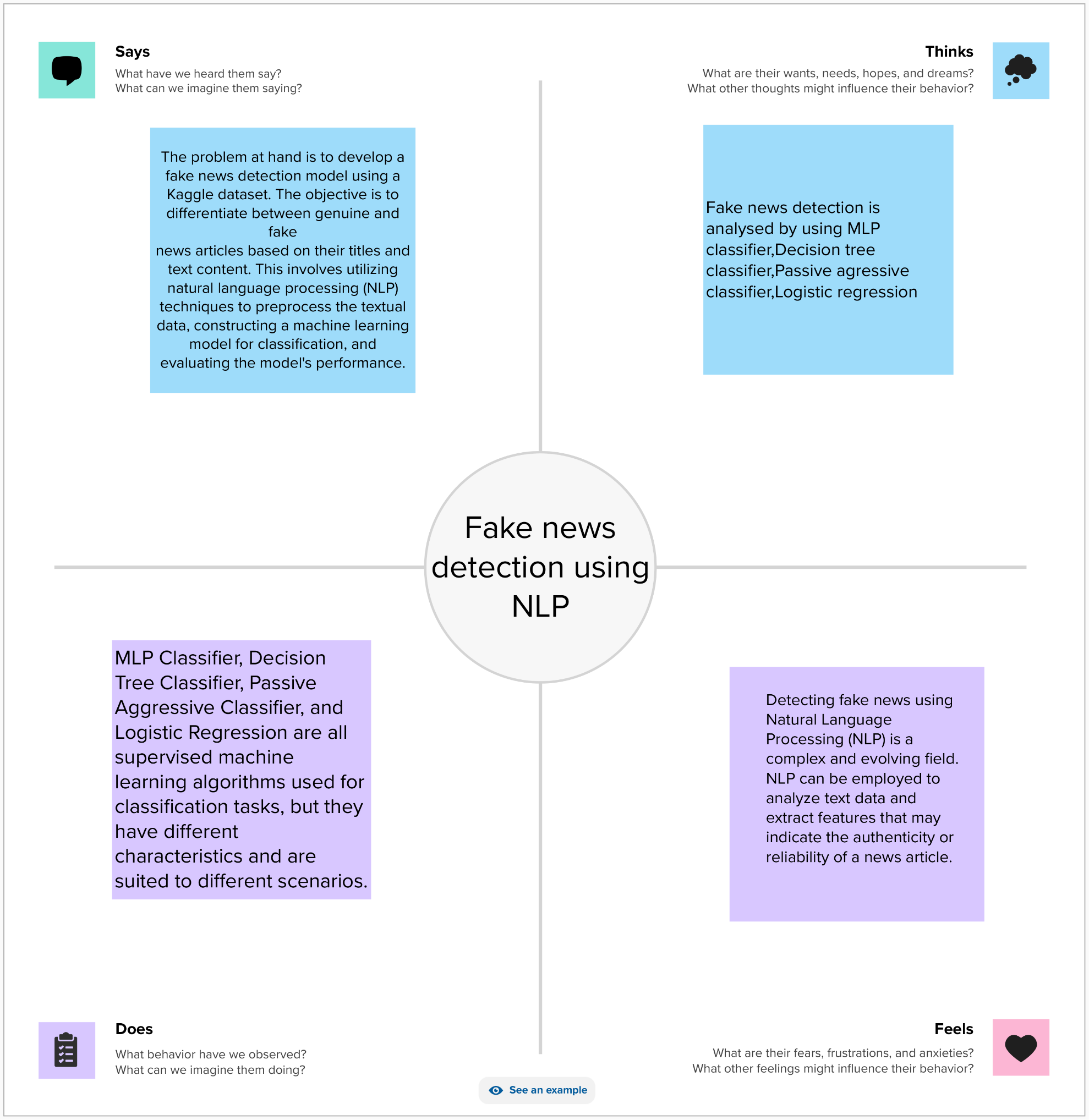
**Model Selection:** Select a suitable classification algorithm (e.g., Logistic Regression, Random Forest, or Neural Networks) for the fake news detection task.

**Model Training:** Train the selected model using the preprocessed data.

**Evaluation:** Evaluate the model's performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC

**3)Ideation and proposed solution**

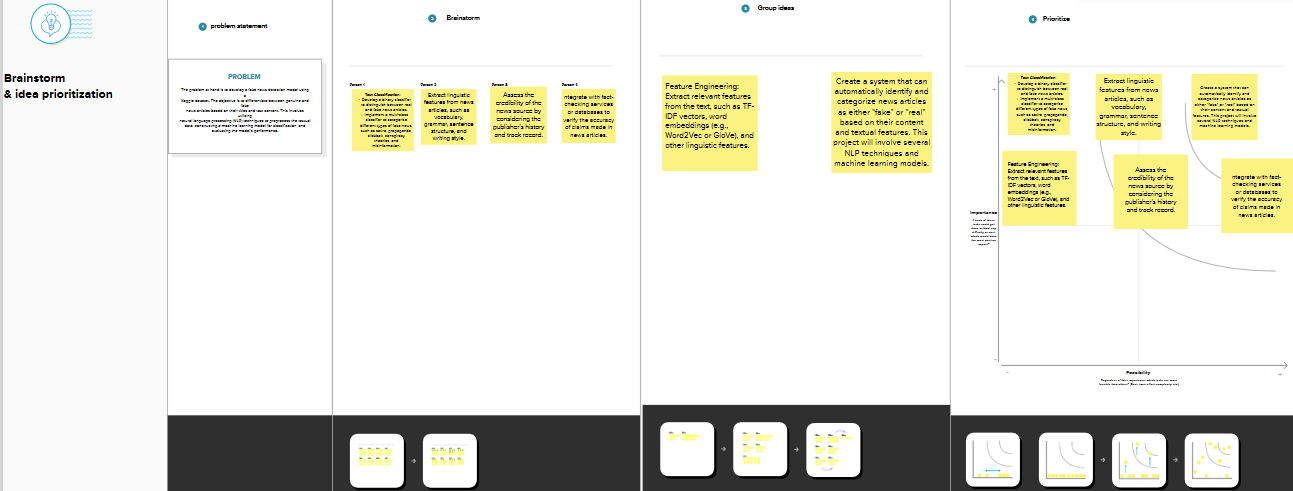
**3.1) Empathy map canvas**



In the digital age, the proliferation of fake news has emerged as a significant societal challenge, necessitating innovative solutions that blend technology and human understanding. One such approach is the application of Natural Language Processing (NLP) in conjunction with an Empathy Map Canvas. This method transcends conventional algorithms, incorporating human insights and emotions into the process of fake news detection.

**3.2) Ideation and Brainstorming:**

Brainstorming is a creative and collaborative problem-solving technique used to generate a large number of ideas or potential solutions to a specific problem or challenge**.**

****

**3.3) Proposed solution:**

|  |  |  |
| --- | --- | --- |
| **SNO.** | **PARAMETER** | **DESCRIPTION** |
| 1. | Problem statement (Problem  to be Solved) | To detect whether the news is fake or not using NLP |
| 2. | Idea/ solution description | By using nlp classifier,decision tree classifier,passive aggressive classifier,logistic regression to detect the accuracy |
| 3. | Novelty/Uniqueness | Detecting fake news using classifiers |
| 4. | Social Impact/Customer  Satisfaction | It enhances public trust in media and  information sources. |
| 5. | Business Model (Revenue  Model)  . | It improves the accuracy and reliability of fake news detection system. |

**:**

**3.4)Problem solution:**

|  |  |  |
| --- | --- | --- |
| 1. **Data Collection:**   Collect a large dataset of news articles, including both credible and fake news. The dataset should be diverse and representative | **2)Data Processing:**  Preprocess the text data, including lowercasing, vectorisation, and removing stop words and punctuation | **3)Labelling:**  Manually or semi-automatically label the collected data  as either credible or  fake news**.** |

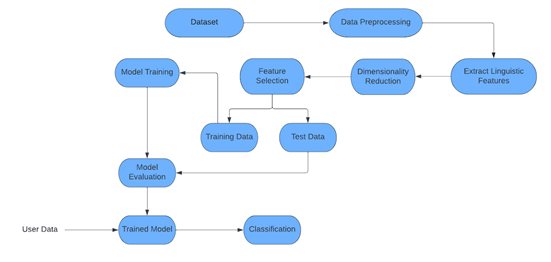
|  |  |  |
| --- | --- | --- |
| **4)Data Splitting:**  Split the dataset into  Training, Validation  and test sets to  evaluate the  models’ performance | **5)Feature Extraction:**  Use the pre-processed data to extract features for model training | **6)Model Selection:**  Choose the best-performing model based on validation results |

|  |  |  |
| --- | --- | --- |
| **7)Evaluation:**    Evaluate the model's performance on the test dataset to assess its ability to detect fake news accurately**.** | **8)Continuous Monitoring**:  Continuously monitor the system's performance and retrain the model as new data becomes available**.** | **9)User Interface:**  Develop a user-friendly interface for users to submit news articles and receive fake news detection results |

**4)Project Design**

**4.1) Data Flow Diagram:**

A Data Flow Diagram (DFD) is a graphical representation used to visualize the flow of data within a system or process. It's a structured way to illustrate how data moves through various components of a system, including processes, data stores, data sources, and data destinations. DFDs are commonly used in system analysis and design to model information flow and to understand the interactions between different elements of a system.

****

**1.Start**: The process begins here.

**2.Input Text**: The input to the system is the news article or text that needs to be checked for authenticity.

**3.Preprocessing**:

* + Text Cleaning: Remove any irrelevant characters, punctuation, and special symbols.
  + Tokenization: Split the text into individual words or tokens.
  + Stopword Removal: Eliminate common words that do not carry much meaning (e.g., "the," "is," "and").

**4.Feature Extraction**:

* + Extract features from the preprocessed text. These features might include word frequencies, TF-IDF scores, or n-grams.

**5.Machine Learning Model**:

* + This is a simplified step. In a real-world scenario, this would be a machine learning or deep learning model trained on labeled data.
  + The model takes the extracted features as input and classifies the news as either "Real" or "Fake."

**6.Prediction**:

* + The model makes a prediction regarding the authenticity of the news.
  + If the prediction score is high for "Real," the news is classified as real;
  + if the score is high for "Fake," it's classified as fake.

**7.Output**:

* + Display the prediction result, indicating whether the news is "Real" or "Fake."

**8.End**: The process ends here.

**Explanation:**

1. The process starts with the input of the news text that you want to analyze.
2. Preprocessing involves cleaning and simplifying the text to prepare it for analysis. Tokenization breaks the text into words, and stopword removal eliminates common words that don't add much value for distinguishing real from fake news.
3. Feature extraction is the process of converting the text into numerical features. These features can be used as input to a machine learning model. Common features include word frequencies (count of words in the text), TF-IDF (Term Frequency-Inverse Document Frequency) scores, and n-grams (sequences of words).
4. The machine learning model, which is typically trained on a dataset of labeled real and fake news articles, takes these features and makes a prediction. In a more sophisticated system, this model could be a neural network or other advanced algorithms.
5. Based on the model's prediction, the system outputs whether the news is classified as "Real" or "Fake."
6. The process ends, and the result is displayed.

**4.2) Solution and Technical Architecture:**

training data

Data pre-processing

Data augmentation

Data cleansing

Pre-processed data

word vector

Load word embedding

Train model

Build deep learning model

Save model

**5)Project planning and schedule:**

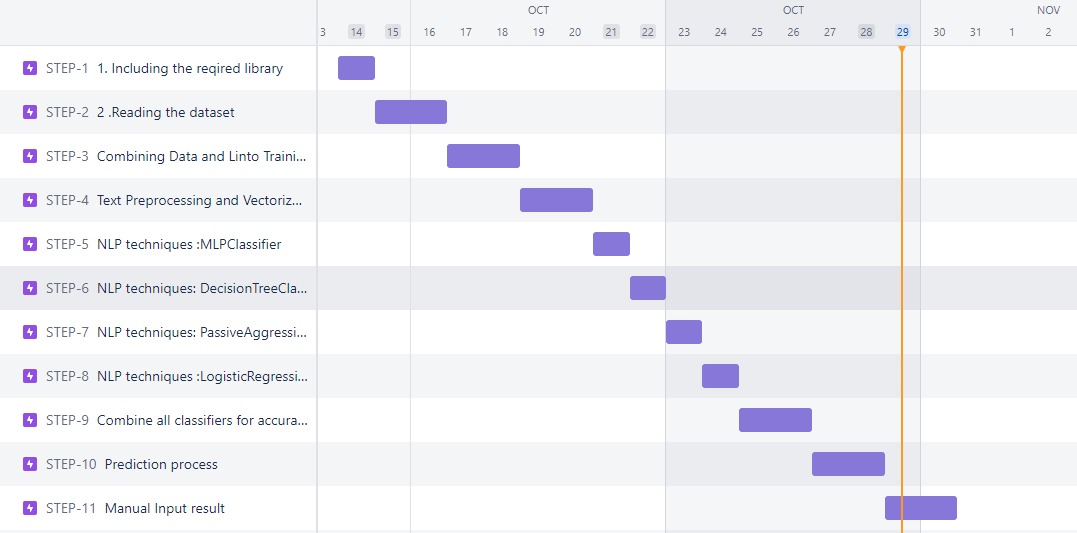
**5.1)Sprint delivery and schedule:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| sprint | Total  Story  points | Duration | sprint  start  date | sprint  end  date | Story  points  completed  (as on  planned  end date | sprint  release  date  (actual) |
| sprint-1 | 20 | 3days | 26sep  2023 | 29sep  2023 | 20 | 29sep  2023 |
| sprint-2 | 20 | 7days | 4oct  2023 | 11oct  2023 | 20 | 11oct  2023 |
| sprint-3 | 20 | 6days | 12oct  2023 | 18oct  2023 | 20 | 18oct  2023 |
| sprint-4 | 20 | 7days | 19oct  2023 | 26oct  2023 | 20 | 26oct  2023 |

**5.2)Sprint Planning And Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement**  **(epic)** | **User**  **story**  **number** | **User story /**  **Task** | **Story**  **point** | **Priority** | **Team**  **members** |
| Sprint-1 | Data | USN-1 | Gather a dataset of  labeled news articles,  where each article is  classified as either "real"  or "fake." These labels  can be obtained from  credible fact-checking  organizations or  crowdsourced from  experts. | 20 | HIGH | Muthamil S  Priyanka K  Swetha S  Shanalin J |
| Sprint-2 | Processing | USN-2 | Clean and preprocess  the text data by removing  stopwords, punctuation,  and converting text to  lowercase | 20 | HIGH | Muthamil S  Priyanka K  Swetha S  Shanalin J |
| Sprint-3 | Model | USN-3 | The models used to detect  the news are  MLP classifier,Passive  Aggressive classifier,  Decision tree classifier,  Logistic regression. | 20 | HIGH | Muthamil S  Priyanka K  Swetha S  Shanalin J |
| Sprint-4 | Final | USN-4 | The news given is finally  Detected whether it is true  Or fake and the classifiers  Also used to measure  Accuracy. | 20 | HIGH | Muthamil S  Priyanka K  Swetha S  Shanalin J |

**5.3)Report from jira**

****

**6) Results:**

It is used to detect whether the given news is fake or real and

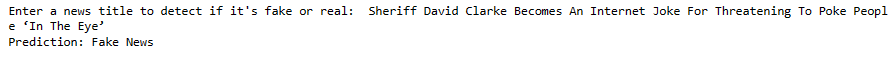
it is also used to measure the accuracy.

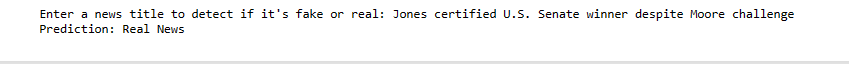
**Examples** for Fake news detection is given below.

When the program is executed , it will display a box . Enter the news you want to detection then it will display whether the news is fake or real.



**Fake news:**



**Real news :**

**7)Advantages**

1. NLP algorithms can automatically analyze and process large volumes of text data, which is essential given the vast amount of news and information available online. This automation makes it possible to detect fake news in real-time and at scale
2. NLP can process information rapidly, allowing for quick identification of potentially misleading or false news stories. This is particularly important in a fast-paced digital news environment where misinformation can spread rapidly
3. NLP models can be trained on vast amounts of data, making them capable of learning patterns and characteristics of fake news. They can adapt and improve over time as more data becomes available
4. NLP models are becoming increasingly sophisticated, allowing for more accurate fake news detection through advanced language understanding and pattern recognition**.**

**8)Disadvantages**

1. Fake news creators can deliberately manipulate text to bypass NLP-based detection systems. Adversarial attacks can be designed to fool the models by making slight, often imperceptible changes to the text.
2. As fake news detection technology improves, so does the sophistication of fake news creators. This creates a constant arms race, with both sides continually adapting and evolving their methods.
3. NLP models can exhibit bias in their predictions, leading to misclassification of legitimate news as fake and vice versa. The bias can stem from the training data used to develop the model, resulting in unfair or incorrect outcomes.
4. Fake news can exist in multiple languages and include various forms of media such as images and videos. Detecting fake news in these contexts is more challenging and requires complex, multilingual, and multimodal models.

**9)Conclusion**

We can print a confusion matrix to get an overview of the number of true and false positives and negatives. fake news detection techniques can be divided into style- and content-based or fact-checking techniques.

It is often assumed that poor writing (poor spelling, poor punctuation, limited vocabulary, overused terminology, incorrect grammar, etc.) is a sure sign of fake news.

More than ever, this is the case when a machine's opinion must be supported by clear and fully verifiable indications on which to base its decisions, within the events have been verified and the competent authority determines the authenticity of each event. Collecting data once will not be enough given the rapid spread of information in today's connected world and the number of articles produced. I hope this was helpful to you. You can comment in the comments section for any question

**10)Future Scope**

1. **Real-Time Detection:** The ability to detect fake news in real-time, as it's published, will become more important. This can help prevent the rapid spread of false information.
2. **Data Labeling and Annotation:** An ongoing challenge is the creation of high-quality labeled datasets for training and testing. Developing more efficient and accurate methods for labeling fake news data is essential.
3. **Ethical Considerations:** As fake news detection becomes more accurate and widespread, it's important to address ethical concerns such as privacy, bias, and freedom of expression.
4. **Deepfakes and Synthetic Content:** Detecting increasingly sophisticated deepfakes and synthetic content is a growing challenge, and NLP models will need to evolve to address this threat.
5. **Behavioral Analysis:** In addition to content analysis, analyzing user behavior, such as sharing patterns and engagement, can be valuable for detecting fake news.
6. **Global Challenges:** Fake news is a global issue, and solutions need to be adaptable to various languages, cultures, and regions.
7. **Open Source Tools and Datasets:** The development of open-source fake news detection tools and datasets will promote collaboration and innovation in this field.

**APPENDIX**

SOURCE CODE USING PYTHON:

import pandas as pd

import matplotlib.pyplot as plt

from simple\_nlp\_library import preprocessing, embeddings

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report

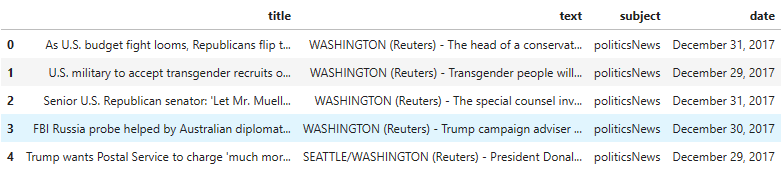
from sklearn.feature\_extraction.text import CountVectorizer

# Load the data

real = pd.read\_csv("True.csv")

fake = pd.read\_csv("Fake.csv")

# previewing the first rows of the real news

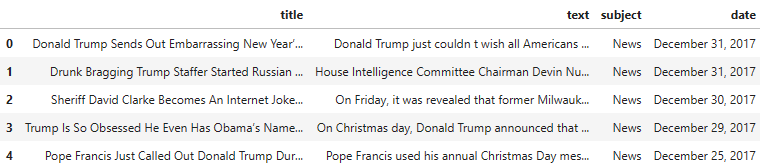
real.head()

**#**previewing the last rows of the real news dataset

real.tail()

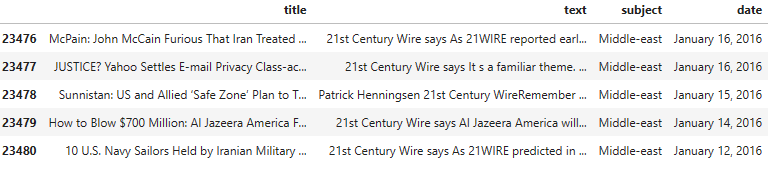
# previewing the first rows of the fake news dataset

fake.head()

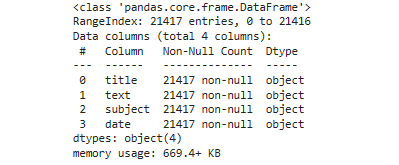


# previewing the last rows of the fake news dataset

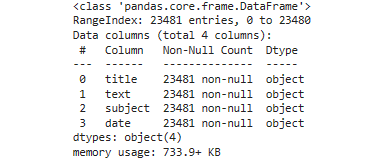
fake.tail()



# Getting information about the real news dataset

real.info()

# Getting information about the fake news dataset

fake.info()

# Combine real and fake news titles and labels

X = real["title"].tolist() + fake["title"].tolist()

y = [0] \* len(real) + [1] \* len(fake)

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Text preprocessing and vectorization

stop\_words = preprocessing.stop\_words()

vectors = embeddings.vectors()

# Vectorize the training and testing data

vectorizer = CountVectorizer(stop\_words=stop\_words)

X\_train\_vec = vectorizer.fit\_transform(X\_train)

X\_test\_vec = vectorizer.transform(X\_test)

# MLPClassifier model

from sklearn.neural\_network import MLPClassifier

mlp = MLPClassifier(random\_state=100)

mlp.fit(X\_train\_vec, y\_train)

# Predictions

y\_pred\_train = mlp.predict(X\_train\_vec)

y\_pred\_test = mlp.predict(X\_test\_vec)

train\_accuracy = accuracy\_score(y\_train, y\_pred\_train)

test\_accuracy = accuracy\_score(y\_test, y\_pred\_test)

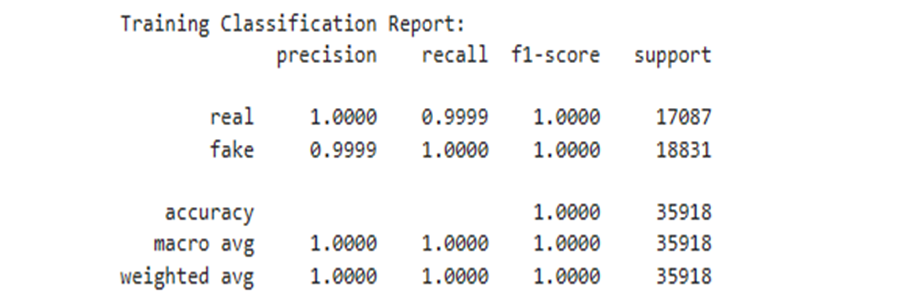
categories = ['real', 'fake']

print(f'Train accuracy-{train\_accuracy}\nTest accuracy-{test\_accuracy}')



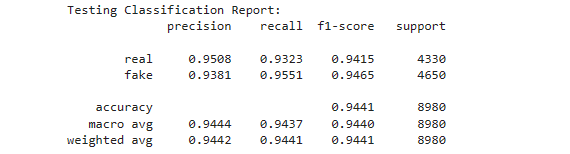
print("Training Classification Report:")

print(classification\_report(y\_train,y\_pred\_train,target\_names=categories,digits=4))



print("Testing Classification Report:")

print(classification\_report(y\_test, y\_pred\_test, target\_names**=**categories, digits**=**4))



x **=** ['Train Accuracy', 'Test Accuracy']

y **=** [train\_accuracy, test\_accuracy]

plt**.**figure(figsize**=**(8, 6))

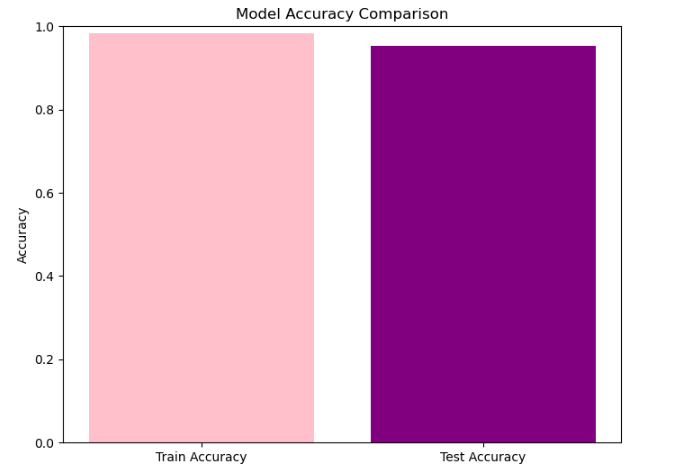
plt**.**bar(x, y, color**=**['pink', 'purple'])

plt**.**ylabel('Accuracy')

plt**.**title('Model Accuracy Comparison')

plt**.**ylim(0, 1) *# Set y-axis limit between 0 and 1 for accuracy values*

plt**.**show()



# Plot the MLP training loss

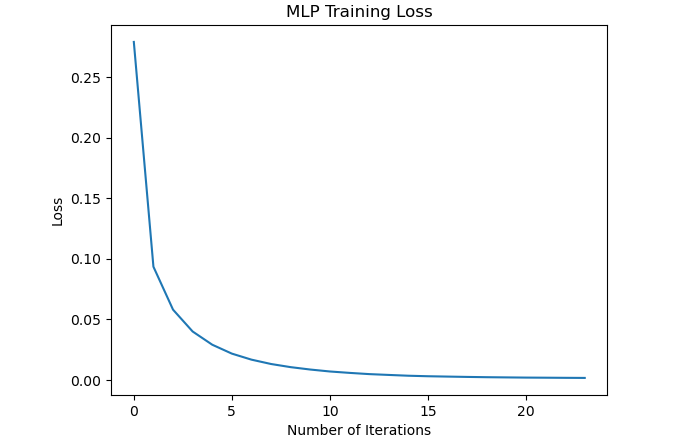
plt.plot(mlp.loss\_curve\_)

plt.title('MLP Training Loss')

plt.xlabel('Number of Iterations')

plt.ylabel('Loss')

plt.show()



#DecisionTreeClassifier

**from** sklearn.tree **import** DecisionTreeClassifier

dt\_classifier **=** DecisionTreeClassifier(random\_state**=**100)

dt\_classifier**.**fit(X\_train\_vec, y\_train)

*# predictions*

y\_pred\_train **=** dt\_classifier**.**predict(X\_train\_vec)

y\_pred\_test **=** dt\_classifier**.**predict(X\_test\_vec)

train\_accuracy **=** accuracy\_score(y\_train, y\_pred\_train)

test\_accuracy **=** accuracy\_score(y\_test, y\_pred\_test)

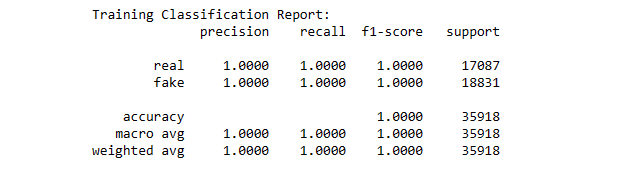
categories **=** ['real', 'fake']

print(f'Train accuracy - {train\_accuracy}\nTest accuracy - {test\_accuracy}')

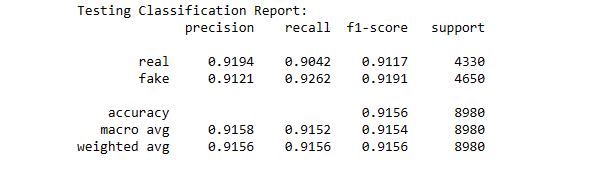


print("Training Classification Report:")

print(classification\_report(y\_train, y\_pred\_train, target\_names=categories, digits=4))



print("Testing Classification Report:") print(classification\_report(y\_test, y\_pred\_test, target\_names=categories, digits=4))



x = ['Train Accuracy', 'Test Accuracy']

y = [train\_accuracy, test\_accuracy]

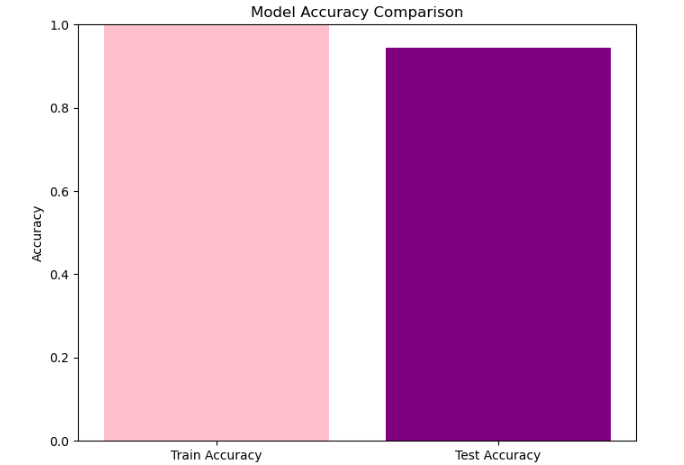
plt.figure(figsize=(8, 6))

plt.bar(x, y, color=['pink', 'purple'])

plt.ylabel('Accuracy')

plt.title('Model Accuracy Comparison')

plt.ylim(0, 1) # Set y-axis limit between 0 and 1 for accuracy



#PassiveAggressiveClassifier

from sklearn.linear\_model import PassiveAggressiveClassifier

pac = PassiveAggressiveClassifier(random\_state=100)

pac.fit(X\_train\_vec, y\_train)

# Predictions

y\_pred\_train = pac.predict(X\_train\_vec)

y\_pred\_test = pac.predict(X\_test\_vec)

train\_accuracy = accuracy\_score(y\_train, y\_pred\_train)

test\_accuracy = accuracy\_score(y\_test, y\_pred\_test)

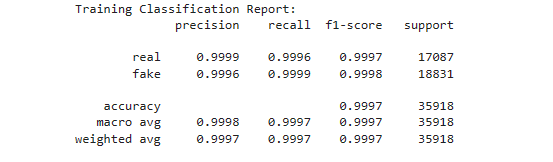
categories = ['real', 'fake']

print(f'Train accuracy - {train\_accuracy}\nTest accuracy - {test\_accuracy}')

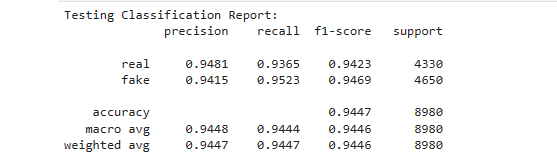


print("Training Classification Report:")

print(classification\_report(y\_train, y\_pred\_train, target\_names=categories, digits=4))



print("Testing Classification Report:") print(classification\_report(y\_test, y\_pred\_test, target\_names=categories, digits=4))



x = ['Train Accuracy', 'Test Accuracy']

y = [train\_accuracy, test\_accuracy]

plt.figure(figsize=(8, 6))

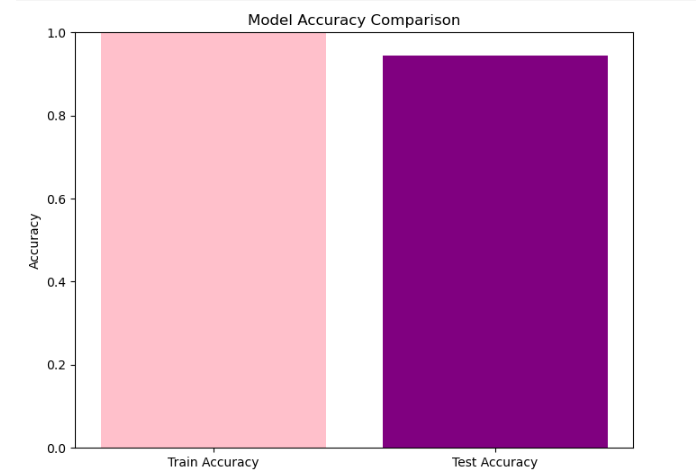
plt.bar(x, y, color=['pink', 'purple'])

plt.ylabel('Accuracy')

plt.title('Model Accuracy Comparison')

plt.ylim(0, 1) # Set y-axis limit between 0 and 1 for accuracy values

plt.show()



#LogisticRegression

from sklearn.linear\_model import LogisticRegression

logreg = LogisticRegression(random\_state=100)

logreg.fit(X\_train\_vec, y\_train)

# Predictions

y\_pred\_train = logreg.predict(X\_train\_vec)

y\_pred\_test = logreg.predict(X\_test\_vec)

train\_accuracy = accuracy\_score(y\_train, y\_pred\_train)

test\_accuracy = accuracy\_score(y\_test, y\_pred\_test)

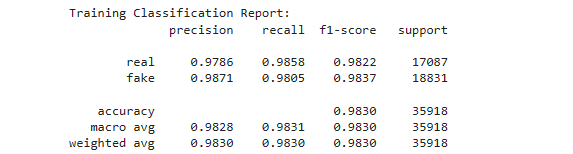
categories = ['real', 'fake']

print(f'Train accuracy - {train\_accuracy}\nTest accuracy - {test\_accuracy}')

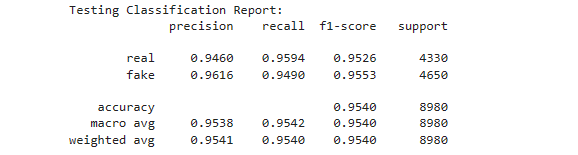


print("Training Classification Report:")

print(classification\_report(y\_train, y\_pred\_train, target\_names=categories, digits=4))



print("Testing Classification Report:") print(classification\_report(y\_test, y\_pred\_test, target\_names=categories, digits=4))



x = ['Train Accuracy', 'Test Accuracy']

y = [train\_accuracy, test\_accuracy]

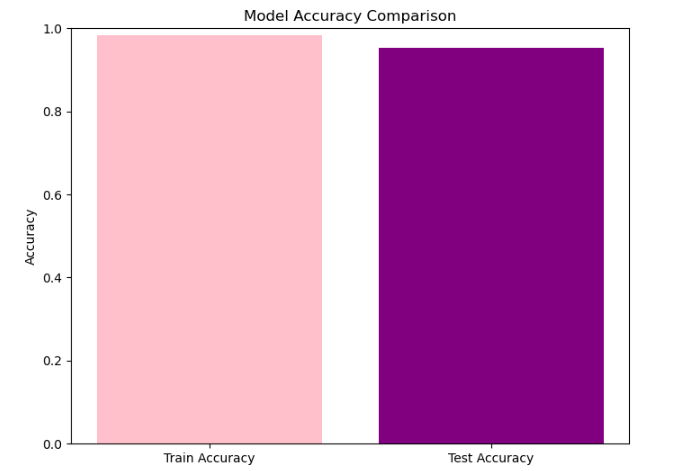
plt.figure(figsize=(8, 6))

plt.bar(x, y, color=['pink', 'purple'])

plt.ylabel('Accuracy')

plt.title('Model Accuracy Comparison')

plt.ylim(0, 1) # Set y-axis limit between 0 and 1 for accuracy values

plt.show() 

# Function to detect fake or real news from manual input

def detect\_fake\_or\_real\_news(input\_title):

input\_vec = vectorizer.transform([input\_title])

prediction = mlp.predict(input\_vec)

prediction = dt\_classifier.predict(input\_vec)

prediction = pac.predict(input\_vec)

prediction = logreg.predict(input\_vec)

if prediction[0] == 0:

return "Real News"

else:

return "Fake News"

#manual input example 1

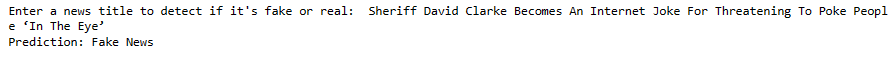
input\_title = input("Enter a news title to detect if it's fake or real: ")

result = detect\_fake\_or\_real\_news(input\_title)

print("Prediction:", result)

#enter any news in the box to find fake or real





#manual input example 2

input\_title = input("Enter a news title to detect if it's fake or real:” ")

result = detect\_fake\_or\_real\_news(input\_title)

print("Prediction:", result)

