**COLLEGE CODE :** 5113

COLLEGE NAME : Kingston Engineering College

DOMAIN : AI –Artificial Intelligence

PROJECT TITLE : Fake News Detection using NLP

PROJECT MEMBERS:

R.Saravanan (511321104088),

K.Tharun(511321104102),

**G.Seeralan(**511321104306**),**

**D.Jagan(**511321104306**)**

**Introduction:**

Fake detection using Natural Language Processing (NLP) is a critical application that has gained prominence in the digital age. With the proliferation of online information and content, distinguishing between genuine and fraudulent text, be it news articles, product reviews, or social media posts, has become a pressing challenge. Design thinking, a human-centered approach to problem-solving, offers a structured and innovative framework to tackle this problem effectively.

**Design Thinking Process:**

Design thinking is a human-centered, iterative problem-solving approach that emphasizes empathy, collaboration, and creative ideation. It typically consists of the following phases:

1. Empathize:
   * In this phase, the focus is on understanding the problem from the perspective of the end users or stakeholders.
   * Activities may include conducting interviews, observations, and surveys to gain insights into users' needs, motivations, and pain points.
2. Define:
   * In the definition phase, you synthesize the information gathered in the empathize phase to create a clear and specific problem statement.
   * This involves reframing the problem in a way that guides the design process.
3. Ideate:
   * Ideation is a creative phase where you generate a wide range of potential solutions to the defined problem.
   * Brainstorming and other creative techniques are often used to encourage diverse and innovative ideas.
4. Prototype:
   * Prototyping involves developing low-fidelity representations of potential solutions.
   * These prototypes can be physical or digital and are used to test and refine ideas quickly.
5. Test:
   * In the testing phase, prototypes are presented to users or stakeholders for feedback and evaluation.
   * The goal is to gather insights on what works, what doesn't, and how the solution can be improved.
6. Implement:
   * Once a viable solution is identified and refined, it is implemented in the real world.
   * This phase involves scaling up the solution and integrating it into the existing systems or processes

**Phases of Development:**

The phases of development refer to the steps taken to bring a solution from concept to execution. These phases can vary depending on the nature of the project, but they often include:

1. Research and Analysis:
   * In this phase, further research may be conducted to validate the chosen solution.
   * Detailed planning and analysis are essential to understand the technical, financial, and logistical aspects of the solution.
2. Design:
   * Design involves creating detailed plans, blueprints, or specifications for the solution.
   * This can include visual design, user interface design, and architectural design, depending on the nature of the project.
3. Development:
   * This is the phase where the actual solution is built or developed.
   * It may involve coding software, manufacturing products, or constructing physical structures.
4. Testing and Quality Assurance:
   * The solution is rigorously tested to ensure it meets the defined requirements and functions as intended.
   * Bugs and issues are identified and addressed during this phase.
5. Deployment:
   * The solution is deployed or released to the intended users or market.
   * This may involve a phased rollout or a full-scale launch, depending on the project's complexity.
6. Maintenance and Iteration:
   * After deployment, ongoing maintenance and updates are essential to ensure the solution continues to meet user needs.
   * Iteration may also be required to improve the solution based on feedback and changing requirements.

**Choice of Classification Algorithm and Model Training Process:**

* **Classification Algorithm:** Consider algorithms like Logistic Regression for simplicity and interpretability in text classification tasks. Alternatively, explore more complex models like Random Forest or deep learning models for improved accuracy.
* **Model Training Process:** Split the preprocessed data into training and testing sets. Train the selected model on the training data and evaluate its performance on the test data. Iterate on the model and preprocessing steps based on evaluation results.
* **Code Files:** Include all code files for data preprocessing, feature extraction, model training, and evaluation.
* **README File:** Create a well-structured README explaining how to run the code, specifying any dependencies, and providing a brief project overview.
* **Dataset Source:** Mention the dataset source (Kaggle) in the README and include a brief description of the dataset.

**CODE :**

import pandas as pd

import numpy as np

import re

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

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import MultinomialNB

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

from tensorflow import keras

from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

# Load the dataset

data = pd.read\_csv("C:\\Users\\prath\\Downloads\\Fake.csv") # Update the path to the dataset file

# Combine the 'title' and 'text' columns into one text column

data['text'] = data['title'] + " " + data['text']

# Remove unnecessary columns

data = data[['text', 'subject']]

# Data cleaning

def clean\_text(text):

text = re.sub(r'\S\*@\S\*\s?', '', text) # Remove emails

text = re.sub(r'\s+', ' ', text) # Remove extra whitespace

text = re.sub(r'[^a-zA-Z]', ' ', text) # Remove non-alphabetic characters

text = text.lower() # Convert to lowercase

return text

data['text'] = data['text'].apply(clean\_text)

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data['text'], data['subject'], test\_size=0.2, random\_state=42)

tfidf\_vectorizer = TfidfVectorizer(max\_features=5000)

X\_train\_tfidf = tfidf\_vectorizer.fit\_transform(X\_train)

X\_test\_tfidf = tfidf\_vectorizer.transform(X\_test)

nb\_classifier = MultinomialNB()

nb\_classifier.fit(X\_train\_tfidf, y\_train)

y\_pred = nb\_classifier.predict(X\_test\_tfidf)

# Evaluation

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

confusion = confusion\_matrix(y\_test, y\_pred)

classification\_rep = classification\_report(y\_test, y\_pred)

print("Accuracy: ", accuracy)

print("Confusion Matrix:\n", confusion)

print("Classification Report:\n", classification\_rep)

**OUTPUT:**

Accuracy: 0.5786672344049393

Confusion Matrix:

[[ 20 0 66 1 26 203]

[ 1 20 21 57 14 46]

[ 3 0 1741 0 34 43]

[ 5 70 26 8 5 46]

[ 15 0 199 0 123 560]

[ 22 0 356 0 160 806]]

Classification Report:

precision recall f1-score support

Government News 0.30 0.06 0.10 316

Middle-east 0.22 0.13 0.16 159

News 0.72 0.96 0.82 1821

US\_News 0.12 0.05 0.07 160

left-news 0.34 0.14 0.20 897

politics 0.47 0.60 0.53 1344

accuracy 0.58 4697

macro avg 0.36 0.32 0.31 4697

weighted avg 0.51 0.58 0.52 4697

**CONCLUSION:**

Fake detection using NLP is a multifaceted challenge in the digital age, and the design thinking process provides a structured approach to tackle this problem while keeping users' needs at the forefront. The development phases encompass data collection, preprocessing, feature engineering, model selection, training, deployment, and ongoing monitoring and improvement to create a robust and effective solution for identifying fake content.