FAKE NEWS DETECTION USING NLP

|  |  |
| --- | --- |
| *Date* | 29/10/2023 |
| *Team ID* | 394 |
| *Project name* | Fake news detection using nlp |

1. **DATA COLLECTION : Gather a dataset of news articles labeled as either real or fake. Several sources, such as Kaggle, offer datasets for this purpose.**
2. **TEXT PREPROCESSING : Clean and preprocess the text data. This includes tasks like removing punctuation, stop words, and stemming/lemmatizing words.**
3. **FEATURE EXTRACTION : Transform the text data into numerical features that can be used for machine learning. Common techniques include TF-IDF (Term Frequency- Inverse Document Frequency) or word embeddings like Word2Vec or GloVe.**
4. **MACHINE LEARNING MODEL ;**

SUPERVISED LEARNING : Train machine learning models, such as logistic regression, Naive Bayes, or decision trees, using the extracted features and labeled data.

DEEP LEARNING : Utilize deep learning models like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or transformer models like BERT for more advanced fake news detection.

1. **EVALUTION : Assess the model's performance using metrics like accuracy,**
2. **FINE TUNNING : Experiment with different models, hyperparameters, and feature extraction techniques to improve the model's performance.**
3. **DEPLOYMENT : Deploy the model for real-time or batch processing, depending on your application.**
4. **CONTINUOUS MONITORING :**

Regularly update and retrain the model to adapt to evolving fake news tactics.

1. **USER INTERFACE : Develop a user-friendly interface for users to input news articles or URLs for verification.**
2. **EXPLAINABLITY : Consider methods for explaining the model's decisions to build trust and transparency, such as LIME or SHAP values.**

PROGRAM :

import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

import plotly.express as px import plotly.graph\_objs as go

from plotly.subplots import make\_subplots

import nltk

from nltk.corpus import stopwords import tensorflow as tf

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.callbacks import ModelCheckpoint from sklearn.model\_selection import train\_test\_split from transformers import AutoTokenizer, TFAutoModelForSequenceClassification

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename)) nltk.download('stopwords')

OUTPUT :

True