FAKE NEWS DETECTION USING NLP

1. **Define the Problem:**

   Clearly define the problem you want to solve. In this case, the problem is to detect fake news articles using NLP techniques.

2. **Data Collection:**

   Gather a dataset of news articles labeled as either real or fake. There are several publicly available datasets for fake news detection, such as the LIAR-PLUS, FakeNewsNet, or Kaggle's Fake News Detection dataset. Make sure to split the data into training and testing sets.

3. **Data Preprocessing:**

   Preprocess the text data to make it suitable for NLP tasks. This may include:

   - Text cleaning (removing HTML tags, special characters, and punctuation).
   - Tokenization (splitting text into words or subword units).
   - Stop word removal (common words like "the," "and" that don't carry much information).
   - Stemming or lemmatization (reducing words to their base form).
   - Text vectorization (converting text data into numerical format).

4. **Feature Engineering:**

   Create meaningful features that can help the model differentiate between real and fake news. Some common techniques include:

   - TF-IDF (Term Frequency-Inverse Document Frequency) vectors.
   - Word embeddings (e.g., Word2Vec, GloVe).
   - N-grams (sequences of 'n' words).
   - Sentiment analysis (using sentiment scores as features).

5. **Model Selection:**

   Choose an appropriate machine learning or deep learning model for fake news detection. Common choices include:

   - Logistic Regression
   - Naive Bayes
   - Random Forest
   - Support Vector Machine
   - Recurrent Neural Networks (RNN)
   - Convolutional Neural Networks (CNN)
   - Transformers (e.g., BERT)

6. **Model Training:**

   Train the selected model on the preprocessed data. Use the training dataset to teach the model how to distinguish between real and fake news.

7. **Model Evaluation:**

   Assess the model's performance using evaluation metrics such as:

   - Accuracy
   - Precision
   - Recall
   - F1-score
   - ROC-AUC
   - Confusion Matrix

8. **Hyperparameter Tuning:**

   Fine-tune your model by adjusting hyperparameters. You can use techniques like grid search or random search to find the best hyperparameter values.

9. **Cross-Validation:**

   Perform cross-validation to ensure your model's generalization and avoid overfitting.

10. **Testing and Deployment:**

    Test your model on the testing dataset to ensure its real-world performance. If the model meets your criteria, you can deploy it for real-time fake news detection.

11. **Continuous Improvement:**

    Keep monitoring and updating your model as new data becomes available. Fake news patterns may change over time, so the model should be adaptable.

12. **Ethical Considerations:**

    Consider the ethical implications of your model and its potential biases. Ensure that your fake news detection system is fair and transparent.

CODING;

```
import pandas as pd
import matplotlib.pyplot as plt
import spacy
from spacy.util import minibatch, compounding
import random

nlp = spacy.load('el__core__news__md')
df1 = pd.read__csv('../data/jtp__fake__news.csv')
df1.replace(to__replace='[ \ n \ r \ t]', value=' ', regex=True, inplace=True)

def load__data(train__data, limit=0, split=0.8):
    random.shuffle(train__data)
    train__data = train__data[-limit:]
    texts, labels = zip(*train__data)
    cats = [{"REAL": not bool(y), "FAKE": bool(y)} for y in labels]
    split = int(len(train__data) * split)

    return (texts[:split], cats[:split]), (texts[split:], cats[split:])  -
def evaluate(tokenizer, textcat, texts, cats):
    docs = (tokenizer(text) for text in texts)
    tp = 0.0  # True positives
    fp = 1e-8  # False positives
```

```
    fn = 1e-8  # False negatives
    tn = 0.0  # True negatives
    for i, doc in enumerate(textcat.pipe(docs)):
        gold = cats[i]
        for the label, score in doc.cats.items():
            if the label is not in gold:
                continue
            if label = = "FAKE":
                continue
            if score > = 0.5 and gold[label] > = 0.5:
                tp + = 1.0
            elif score > = 0.5 and gold[label] < 0.5:
                fp + = 1.0
            elif score < 0.5 and gold[label] < 0.5:
                tn + = 1
            elif score < 0.5 and gold[label] > = 0.5:
                fn + = 1
    precision = tp / (tp + fp)
    recall = tp / (tp + fn)
    if (precision + recall) = = 0:
        f__score = 0.0
    else:
        f__score = 2 * (precision * recall) / (precision + recall)
    return {"textcat__p": precision, "textcat__r": recall, "textcat__f": f__score
    }
In [3]:
df1.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total five columns):
 #   Column   Non-Null Count  Dtype
 0   title    100 non-null    object
 One text     100 non-null    object
 Two sources 100 non-null    object
 Three url      100 non-null    object
 4   is__fake  100 non-null    int64
dtypes: int64(1), object(4)
memory usage: 4.0+ KB
textcat=nlp.create__pipe( "textcat", config={"exclusive__classes": True, "ar
    chitecture": "simple__cnn"})
```

```
nlp.add__pipe(textcat, last=True)
nlp.pipe__names
['tagger', 'parser', 'ner', 'textcat']
textcat.add__label("REAL")
textcat.add__label("FAKE")
df1['tuples'] = df1.apply(lambda row: (row['text'], row['is__fake']), axis=1)
train = df1['tuples'].tolist()
(train__texts, train__cats), (dev__texts, dev__cats) = load__data(train, split=
    0.9)

train__data = list(zip(train__texts,[{'cats': cats} for cats in train__cats]))
n__iter = 20
other__pipes = [pipe for pipe in nlp.pipe__names if pipe != 'textcat']
with nlp.disable__pipes(*other__pipes):  # only train textcat
    optimizer = nlp.begin__training()

    print("Training the model...")
    print('{:^5}\t{:^5}\t{:^5}\t{:^5}'.format('LOSS', 'P', 'R', 'F'))
```

OUTPUT;

array([1716, 1722, 122, 363, 311, 322, 236, 228, 220, 226, 223, 220, 206, 202, 283, 282, 280, 278, 275, 266, 266, 261, 262, 256, 255, 253, 252, 215, 211, 213, 237, 233, 232, 232, 230, 226, 228, 225, 221, 223, 222, 222, 220, 226, 228, 227, 226, 221, 222, 220, 206, 208, 206, 205, 201, 203, 202, 202, 200, 66, 68, 67, 66, 65, 61, 63, 62, 60, 86, 88, 87, 86, 81, 83, 82, 76, 78, 77, 76, 75, 71, 73, 72, 72, 70, 66, 68, 67, 66, 65, 61, 63, 62, 62, 60, 56, 58, 57, 56, 55, 51, 53, 52, 52, 50, 16, 18, 17, 16, 15, 11, 13, 12, 12, 10, 36, 38, 37, 36, 35, 31, 33, 32, 32, 30, 26, 28, 27, 26, 25, 21, 23, 22, 221, 223, 222, 222, 220, 226, 228, 227, 226, 221, 222, 220, 206, 208, , 280, 278, 275, 266, 266, 261, 262, 256, 255, 253, 252, 215, 211, 213, 237, 233, 232, 232, 230, 226, 228, 225, 221, 223, 222, 222, 220, 226, 228, 227, 226, 221, 222, 206, 205, 201, 203, 202, 202, 200, 66, 68, 67, 66, 65, 61, 63, 62, 60, 86, 88, 87, 86, 81, 83, 82, 76, 78, 77, 76, 22, 20, 26, 28, 27, 26, 25, 21, 23, 22, 22, 20, 6, 8, 7, 6, 5, 1, 3, 2, 2])

R.DHIWAKAR

NANDHA COLLEGE OF TECHNOLOGY
26.10.2023