COMBAT ONLINE PLAGIARISM

STEPS:

1.**Text Preprocessing**: Cleaning and preparing the text data.

2.**Feature Extraction**: Converting text to a numerical format that a machine learning model can understand.

3. **Model Training**: Training a machine learning model to detect plagiarism.

4. **Deployment**: Setting up a simple web server to allow users to upload documents and check for plagiarism.

CODE:

```
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from flask import Flask, request, jsonify
# Download necessary NLTK data
nltk.download('stopwords')
```

```
nltk.download('punkt')
# Sample Data (Replace with your actual data)
documents = [
    "The internet is flooded with content, making it challenging to spot plagiarism.",
    "Our AI-powered tool can help authors and news organizations quickly detect copied content.",
    "Don't let plagiarism go unnoticed; empower yourself with our plagiarism detection software.",
    "This is an original piece of content written by an author.",
    "This content is a copied version of another document."
]
# Preprocessing function
def preprocess(document):
    tokens = nltk.word_tokenize(document)
    tokens = [word.lower() for word in tokens if word.isalpha() and word not in nltk.corpus.stopwords.words('english')]
    return ' '.join(tokens)

# Preprocess documents
documents = [preprocess(doc) for doc in documents]
```

```
# Vectorize the documents
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(documents)
# Flask app for deployment
app = Flask(__name__)
@app.route('/check_plagiarism', methods=['POST'])
def check_plagiarism():
    data = request.json
    new_document = data.get('document')
    if new_document:
        # Preprocess the new document
        preprocessed_doc = preprocess(new_document)
        new_doc_vector =
vectorizer.transform([preprocessed_doc])

        # Compute similarity
        similarity_scores = cosine_similarity(new_doc_vector,
tfidf_matrix)
        max_similarity = max(similarity_scores[0])
        # Determine if the document is plagiarized
```

```
        if max_similarity > 0.5:  # Threshold can be adjusted
            result = 'Plagiarized'
        else:
            result = 'Original'
        return jsonify({'document': new_document,
'similarity_score': max_similarity, 'result': result})
    else:
        return jsonify({'error': 'No document provided'}), 400
if __name__ == '__main__':
    app.run(debug=True)
```

Instructions for Running the Code:

1. **Install Dependencies:**

```
pip install scikit-learn Flask
```

2. **Save the Code:**

- Save the code in a file named minimal_plagiarism_detection.py.

3. **Run the Flask App:**

```
python minimal_plagiarism_detection.py
```

4. **Test the API:**

- Use curl, Postman, or another HTTP client to send a POST request to http://127.0.0.1:5000/check_plagiarism.