PLAGIARISM DETECTION BETWEEN DOCUMENT PAIRS

EX.NO : 9

DATE : / /2025

DETECT PLAGIARISM BETWEEN PAIRS OF DOCUMENTS USING COSINE SIMILARITY OR JACCARD COEFFICIENT

AIM:

To write a program to detect plagiarism between two documents using Cosine Similarity (TF-IDF) and Jaccard Coefficient (shingle/ n-gram overlap), and to decide if documents are likely plagiarized based on similarity thresholds.

ALGORITHM:

Step 1: Start
Step 2: Import necessary libraries.
Step 3: Load and preprocess news dataset.
Step 4: Clean articles (lowercase, remove punctuation, stopwords).
Step 5: Convert text to TF■IDF vectors.
Step 6: Split into train and test sets.
Step 7: Train Logistic Regression model.
Step 8: Evaluate accuracy and classification report.
Step 9: Test with a custom input article.

PROGRAM:

```
# cosine_plagiarism.py

import re

import numpy as np

from sklearn.feature_extraction.text import

TfidfVectorizer

from sklearn.metrics.pairwise import

cosine_similarity

import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords



STOP = set(stopwords.words('english'))



def preprocess_text(text, remove_stopwords=True):

    text = text.lower()

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

normalize whitespace

    text = re.sub(r'[^a-z0-9\s]', ' ', text)      #

remove punctuation (keep numbers optionally)
```

```
    if remove_stopwords:
        words = [w for w in text.split() if w not in
STOP]
        return " ".join(words)
    return text

def cosine_plagiarism_score(doc1, doc2,
remove_stopwords=True):
    docs = [preprocess_text(doc1, remove_stopwords),
preprocess_text(doc2, remove_stopwords)]
    tfidf = TfidfVectorizer(ngram_range=(1,2))  #
unigrams + bigrams
    vecs = tfidf.fit_transform(docs)
    sim = cosine_similarity(vecs[0], vecs[1])[0,0]
    return sim

if __name__ == "__main__":
    d1 = """Machine learning is a field of
artificial intelligence that uses statistical
techniques to give computer systems the ability to
learn from data."""
    d2 = """Machine learning is a branch of AI which
uses statistics to allow computers to learn from
data."""
    score = cosine_plagiarism_score(d1, d2)
    print(f"Cosine similarity (TF-IDF) =
{score:.4f}")
    # Threshold guide
    if score > 0.75:
        print("High similarity — possible
plagiarism")
    elif score > 0.45:
        print("Moderate similarity — review for
```

```
paraphrase or partial copying")

    else:

        print("Low similarity")
```

RESULT:
Thus, a program has been successfully executed.