1. **Implement a basic information retrieval system using TF-IDF (Term Frequency-Inverse Document Frequency) for document ranking using python.**

**Aim:**

To implement a basic information retrieval system using TF-IDF (Term Frequency-Inverse Document Frequency) for document ranking using python.

**Code:**

import nltk

from sklearn.feature\_extraction.text import TfidfVectorizer

# Sample documents

documents = []

n = int(input("Enter number of documents: "))

for i in range(n):

doc = input(f"Enter document {i+1}: ")

documents.append(doc)

# Create TF-IDF model

vectorizer = TfidfVectorizer()

tfidf\_matrix = vectorizer.fit\_transform(documents)

# Display TF-IDF scores

print("\nTF-IDF Scores:")

feature\_names = vectorizer.get\_feature\_names\_out()

for i, doc in enumerate(documents):

print(f"\nDocument {i+1}:")

for word, score in zip(feature\_names, tfidf\_matrix[i].toarray()[0]):

if score > 0:

print(f"{word}: {score:.4f}")

**Input:**

Enter number of documents: 3

Enter document 1: The cat sat on the mat

Enter document 2: The dog barked at the cat

Enter document 3: The mat was near the door

**Output:**

TF-IDF Scores:

Document 1:

cat: 0.3565

mat: 0.3565

on: 0.4687

sat: 0.4687

the: 0.5536

Document 2:

at: 0.4484

barked: 0.4484

cat: 0.3410

dog: 0.4484

the: 0.5297

Document 3:

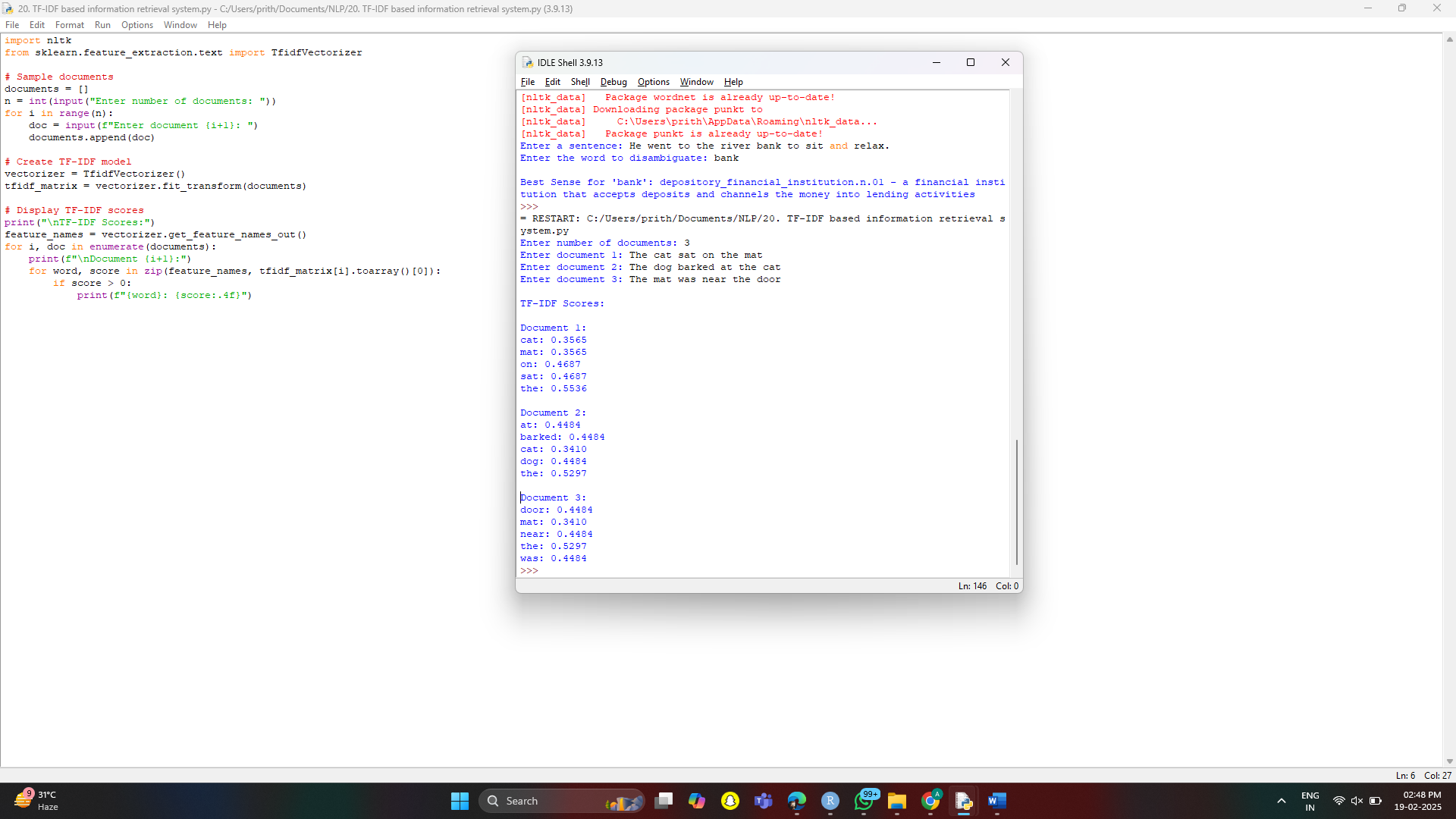
door: 0.4484

mat: 0.3410

near: 0.4484

the: 0.5297

was: 0.4484

****