EX: 5

NLP Task : part of speech Tagging and information retrieval

AIM:

By using NLP libraries to perform parts of speech tagging

PROGRAM CODE :

```
import pandas as pd
import re
import spacy
nlp = spacy.load("en_core_web_sm")
text = "AI-driven platforms personalise learning
        paths and help student grasp
        concepts"
doc = nlp(text)
for token in doc:
    print(f"{token.text:15} → {token.pos_}")
from sklearn.feature_extraction.text import
        vectorizer
from sklearn.metrics.pairwise import cosine_
        similarity

documents = [
    "AI tools analyse student performance and
    provide real time feedback.",
    "Intelligent tutoring system adapt to
    each student's learning style".
```

output:

AI → PROPN

- → PUNCT

driven → VERB

platforms → NOUN

personalize → VERB

learning → VERB

paths → NOUN

and → CCONJ

help → VERB

Students → NOUN

grasp → VERB

concepts → NOUN

faster → ADV

. → FUNCT

```
query = "How does AI support students in
                                learning ?"
corpus = documents + [query]
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(corpus)
similarities = cosine_similarity(tfidf_matrix[-1],
                tfidf_matrix[:-1]).flatten()

ranked_docs = sorted(zip(similarities,
                documents), reverse=TRUE)
print("\n Top relevant documents: \n")
    for score, doc in ranked_docs:
        print(f"Score : {score:.3} → {doc}").
```

RESULT:

Thus nlp programme has been executed successfully.

2025/10/25

9/9/25

Ex no: 6 Exploratory Data Analysis with Python

AIM:

To do exploratory data Analysis with python

Program:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv ("netflix_titles.csv")

print (df.info())
print (df.head())
print (df.describe (include = 'all'))
print ("Number of unique countries", df['country'].
                                    nunique())
print ("Number of unique directors:", df['director'].
                                    nunique())

print (df['type'].value_counts())
print (df['release_year'].value_counts().head(5))
print(df.groupby (['country', 'type']).size().
                    sort_values (ascending = False).
                    head (10))

df['date_added'] = pd.to_datetime (df['date_add
                    format = 'mixed', errors = 'co
```

2025/10/25

output :

<class 'pandas.core.frame.DataFrame'>

RangeIndex : 8807 entries, 0 to 8806

Data columns (total 12 columns):

#	column	Non-Null Count	Dtype
0	show_id	8807	non-null object
1	type	8807	non-null object
2	title	8807	non-null object
3	director	6173	non-null object
4	cast	7982	non-null object
5	country	7976	non-null object
6	date_added	8797	non-null object
7	release_year	8807	non-null int64
8	rating	8803	non-null object

dtypes : int64 (1), object (11)

Number of unique countries : 748

Number of unique directors : 4528

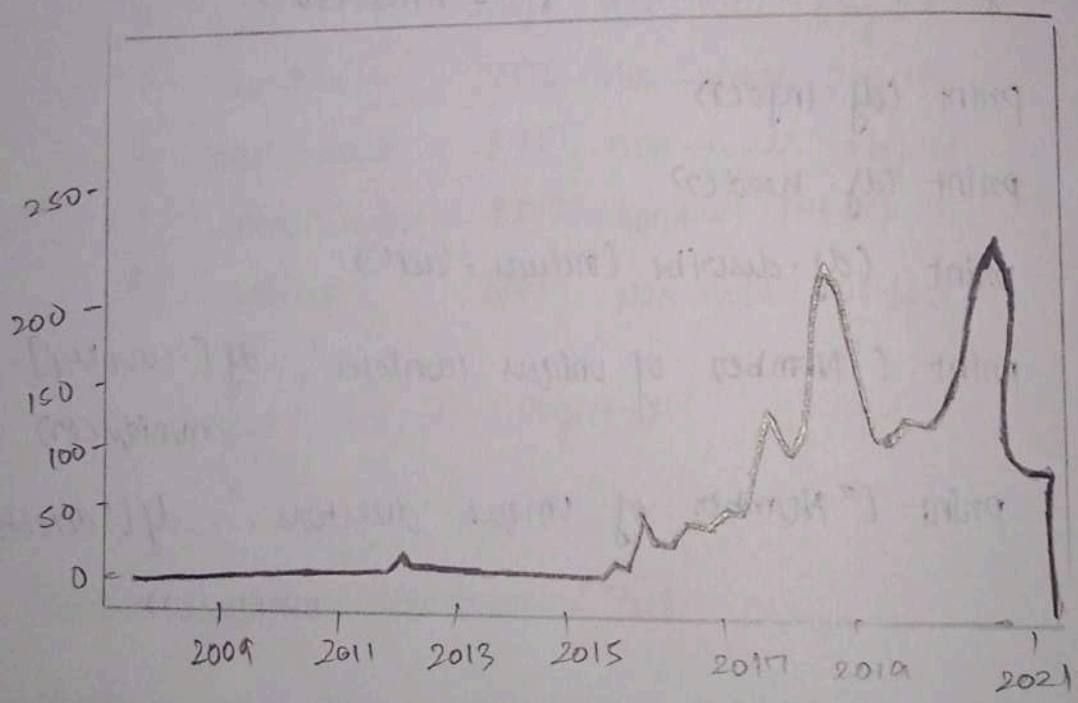
type

Movie 6131

TV show 2676

Name: count, dtype : int64

```
df.set_index('date_added', inplace=True)
monthly_content = df.resample('M').size()
plt.figure(figsize=(12,6))
monthly_content.plot()
plt.title("Netflix Content Added Over Time")
plt.xlabel("Date")
plt.ylabel("Number of Titles Added")
plt.grid(True)
plt.show()
plt.figure(figsize=(10,5))
sns.histplot(df['release_year'], bins=30, kde=False)
plt.title("Distribution of Release Years")
plt.xlabel("Release Years")
plt.ylabel("Number of Titles")
plt.show()
sns.countplot(data=df, x='type', palette='Set2')
plt.title("Count of Movies and TV Shows")
plt.show()
top_countries = df['country'].value_counts().head(10)
top_countries.plot(kind='bar', color='skyblue')
plt.title("Top 10 Countries by Number of Titles")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.show()
```


250
200
150
100
50
0
2009
2011
2013
2015
2017
2019
2021

2025/10/25 12:02