RAJASTHAN TECHNICAL UNIVERSITY, KOTA

IV Year-VII&VIII

Semester: B.Tech. Computer Science & Engineering (Data Science)

7CDS4-22: Social Media Analytics Lab

Credit:2	**Max.Marks:100(IA:60,ETE:40)**
0L+0T+4P	**End Term Exam: 2 Hours**

1) Preprocessing Text Documents using NLTK covering:

a. **Stopword Elimination**
b. **Stemming**
c. **Lemmatization**
d. **POS Tagging**

Step-by-step Implementation Using NLTK

Step 1: Install and Import NLTK

```
import nltk
nltk.download('punkt_tab')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger_eng')
nltk.download('omw-1.4')

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk import pos_tag
```

Input Text

```
text = "The striped bats are hanging on their feet for best."
```

---

a. Stopword Elimination

```
stop_words = set(stopwords.words('english'))
words = word_tokenize(text)
filtered_words = [word for word in words if word.lower() not in stop_words]

print("After Stopword Removal:", filtered_words)
```

---

b. Stemming

```
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in filtered_words]

print("After Stemming:", stemmed_words)
```

---

c. Lemmatization

```
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(word) for word in filtered_words]

print("After Lemmatization:", lemmatized_words)
```

---

d. POS Tagging

```
pos_tags = pos_tag(filtered_words)

print("POS Tagging:", pos_tags)
```

Output Sample (may vary based on input):

For the input `"The striped bats are hanging on their feet for best."`, the output would be something like:

```
After Stopword Removal: ['striped', 'bats', 'hanging', 'feet', 'best', '.']
After Stemming: ['stripe', 'bat', 'hang', 'feet', 'best', '.']
After Lemmatization: ['striped', 'bat', 'hanging', 'foot', 'best', '.']
POS Tagging: [('striped', 'JJ'), ('bats', 'NNS'), ('hanging', 'VBG'), ('feet',
'NNS'), ('best', 'JJS'), ('.', '.')]
```

Let me know if you want this wrapped into a function or want to apply it to a full document (e.g., `.txt` or `.csv` file).

2) **Sentiment Analysis on Customer Reviews** using **NLTK in Python**.

Objective

Analyze whether a customer review is **positive**, **negative**, or **neutral**.

Steps Using NLTK

1. Install and Import Libraries

```
import nltk
nltk.download('vader_lexicon')

from nltk.sentiment import SentimentIntensityAnalyzer
```

2. Initialize the Sentiment Analyzer

```
sia = SentimentIntensityAnalyzer()
```

3. Example: Analyze a Customer Review

```
review = "The product quality is excellent and delivery was super fast!"

# Analyze sentiment
score = sia.polarity_scores(review)

print("Sentiment Scores:", score)
```

4. Interpret the Sentiment

```
compound_score = score['compound']

if compound_score >= 0.05:
    sentiment = "Positive"
elif compound_score <= -0.05:
    sentiment = "Negative"
else:
    sentiment = "Neutral"

print("Overall Sentiment:", sentiment)
```

Test with Multiple Reviews

```
reviews = [
    "Absolutely love this product! Works perfectly.",
    "It's okay, not the best but not the worst either.",
    "Terrible experience. The item broke after one use."
]

for r in reviews:
    score = sia.polarity_scores(r)
    compound = score['compound']
    sentiment = "Positive" if compound >= 0.05 else "Negative" if compound <= -0.05 else "Neutral"
    print(f"Review: {r}\nSentiment: {sentiment}\n")
```

About VADER (Used by NLTK)

- VADER (Valence Aware Dictionary and sEntiment Reasoner) is specifically attuned to sentiments expressed in social media and reviews.
- It outputs 4 scores:
    - **neg**: Negative
    - **neu**: Neutral
    - **pos**: Positive
    - **compound**: Overall score between -1 (most negative) and +1 (most positive)

Optional: Analyze Reviews from a File

---

3) **Web Analytics** focusing on:

Web analytics involves collecting, measuring, analyzing, and reporting web data to understand and optimize web usage.

---

a. Web Usage Data

1. Web Server Log Data

- Logs generated by web servers that record **requests** made to the website.
- **Typical Information Captured:**
  - IP address
  - Timestamp
  - Requested URL
  - HTTP method (GET, POST)
  - Status code (200, 404, etc.)
  - User-agent (browser/device)

Use Cases:

- Track user behavior
- Analyze peak access times
- Identify errors (e.g., 404 pages)

📄 Sample Log Entry:

```
192.168.1.1 - - [20/May/2025:14:15:32] "GET /product/123 HTTP/1.1" 200 "Mozilla/5.0"
```

---

2. Clickstream Analysis

- Sequence of clicks (pages visited) by a user during a session.
- Often visualized as a **path** or **funnel**.

Use Cases:

- Understand user navigation patterns
- Detect drop-off points (e.g., cart abandonment)
- Improve UX design

🔍 Example:

```
Homepage → Category Page → Product Page → Cart → Checkout
```

---

b. Hyperlink Data

This includes the **structure of links** between web pages, both **internal** and **external**.

Types of Hyperlink Data:

- **Inbound Links (Backlinks)**: Links from other websites to yours.
- **Outbound Links**: Links from your site to other websites.
- **Internal Links**: Links within the same website (used for SEO and navigation).

Use Cases:

- Web structure mining
- PageRank and authority scoring
- Site architecture optimization

🔗 Applications:

- **Search engines** use hyperlink data for ranking (e.g., Google's PageRank).
- **Crawlers** use it to discover new pages.

Tools Used for Web Analytics

Category	Tools
ver Log Analysis	Stats, Webalizer
kstream Tracking	ogle Analytics, Hotjar, Matomo
erlink Analysis	eaming Frog, Ahrefs, SEMrush

---

4) **Spamdexing** (a portmanteau of "spam" and "indexing") refers to unethical or manipulative practices used to improve a website's search engine rankings. These practices are against the guidelines of major search engines like Google and **should not be implemented** in real-world websites because:

- They violate search engine policies.
- They can get your site penalized or **blacklisted**.
- They create a poor experience for users.

Why You Should Not Implement Spamdexing

Spamdexing is **black-hat SEO**, and includes:

- **Keyword stuffing**: Repeating keywords excessively
- **Cloaking**: Showing different content to users vs. search engines
- **Hidden text/links**: Making text invisible but still present in the HTML
- **Doorway pages**: Low-quality pages stuffed with keywords that redirect to real pages
- **Link farms**: Creating large groups of interlinked pages for fake authority

---

Instead: Implement Ethical SEO (White-Hat SEO)

Here's how to do **ethical SEO** using Python for educational purposes:

Example: Keyword Optimization Using Python

```
from collections import Counter
import re

text = """
SEO is the practice of improving the ranking of a website on search engines.
Search engine optimization involves keyword research, technical optimization,
and content creation.
"""

# Tokenize and clean
words = re.findall(r'\b\w+\b', text.lower())
keyword_counts = Counter(words)

# Display top keywords
top_keywords = keyword_counts.most_common(5)
print("Top keywords:", top_keywords)
```

Instead of Spamdexing, You Should Focus On:

- Writing **high-quality, relevant content**
- Using proper **title tags** and **meta descriptions**
- Earning **backlinks** naturally
- Improving **page speed** and **mobile usability**
- Using **semantic HTML** and **structured data**

5) Using **Google Analytics** (GA4) to implement:

---

a. Conversion Statistics

What It Means:

Conversion statistics track how many users complete desired actions (like purchases, signups, downloads).

How to Implement in Google Analytics (GA4)

1. Set Up a Conversion Event

1. Go to **Admin → Events**.
2. Click **"Create event"** to define an action like `purchase`, `form_submit`, `signup`, etc.
3. After creating the event, mark it as a **Conversion**.

2. View Conversion Stats

- Navigate to **Reports → Engagement → Conversions**
- You'll see:
  - **Total conversions**
  - **Conversion rate**
  - **Events per session**

Example Events:

- `purchase`
- `generate_lead`
- `add_to_cart`
- `form_submission`

Tip:

Use **Google Tag Manager (GTM)** to send custom event data (e.g., `signup_complete`) from your website to GA4.

---

b. Visitor Profiles

What It Means:

Get demographic and behavioral insights about your visitors.

How to View in Google Analytics (GA4)

1. Enable Google Signals

1. Go to **Admin → Data Settings → Data Collection**
2. Turn on **Google Signals** for demographics and interest reporting

2. View Visitor Data

- Navigate to **Reports → User → Demographics**
  - View by **country, city, language, gender, age**
- Navigate to **Reports → Tech**
  - View by **device, OS, browser**

Additional Visitor Insights:

- **New vs Returning users**
- **Engagement time**
- **Source/medium** (e.g., traffic from Google vs Facebook)
- **User journeys and funnels**

Example Dashboard Sections:

Metric	What It Shows
sions	al visits
rs	que visitors
. engagement time	e spent on site
nce rate	f users who leave quickly
ice category	ktop / Mobile / Tablet

Tools Used: 1)Google Analytics 2) Google Tag Manager

6) To **implement and analyze Traffic Sources using Google Analytics (GA4)**, follow the steps below:

What It Means:

Traffic sources show where your website visitors are coming from (e.g., Google search, direct visits, social media, referrals).

✅ How to Implement Traffic Sources in GA4

Step 1: Set Up Google Analytics on Your Website

Make sure your website has GA4 tracking installed:

- Via the **GA4 tracking code** (gtag.js)
- Or through **Google Tag Manager**

Step 2: View Traffic Source Reports

Go to your GA4 property:

1. **Reports → Acquisition → Traffic acquisition**

You'll see:

Source	Mediu	Examples
gle	anic	rch engine
ect	ne	. typed directly or bookmark
ebook.com	erral	ks from another site
ail_campaign	ail	m email marketing tools
ner)	ner)	categorized sources

Sample Metrics in Traffic Source Report:

- **Sessions**: Number of site visits

- **Users**: Unique visitors
- **Engagement Rate**
- **Conversions**
- **Average Engagement Time**

Example Use Case:

If you're running a Facebook ad and an email campaign, GA4 helps you:

- See how many visitors came from **Facebook**
- See how many came from the **email newsletter**
- Track **which source led to conversions**

Optional: Add UTM Tags for Custom Source Tracking

To get **precise tracking**, use **UTM parameters** in your URLs.

Example URL:

```
https://yourwebsite.com?utm_source=facebook&utm_medium=social&utm_campaign=spr
ing_sale
```

Tools:

Use Google's Campaign URL Builder

▯ Summary

Action	Tool	Result
ck all visitor sources	4 → Reports → Acquisition	rce/Medium breakdown
tomize traffic labeling	M parameters	re detailed reporting
tralize tag management	ogle Tag Manager	y setup and updates
