Velammal College of Engineering and Technology, Madurai

(Autonomous)

Department of Artificial Intelligence &Data Science

Academic Year: 2025-2026

VCET Regulation 2021

LAB MANUAL

21AD307-NATURAL LANGUAGE PROCESSING LABORATORY

SUBJECT NAME	NATURAL LANGUAGE PROCESSING LABORATORY
SUBJECT CODE	21AD307
BRANCH	ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
SEMESTER	VI– EVEN
YEAR	III
BATCH	2023 – 2027

Prepared by — **D.VIJAYADURGA, AP/AI&DS**

Verified by — **Dr. S. SASIKALA, HoD/AI&DS**

VISION

To emerge and sustain as academic excellence in Artificial Intelligence and Data Science to produce ethical professionals through innovative research and education.

MISSION

- To promote industry ready graduates by acquiring intelligent data analytical skills.
- To empower the graduates towards research and application-oriented knowledge for higher studies.
- To equip the graduates with entrepreneurship skills to serve the needs of society.

Table of Contents

Ex. No 1. IMPLEMENTATION OF RESUME SCREENING

Aim:

To design and implement a resume screening system using Python and Natural Language Processing techniques that automatically matches resumes with a given job description using TF-IDF vectorization and cosine similarity.

Tools Required: PYTHON IDE.

Program:

```
import os

import nltk

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine_similarity

job_desc = """Looking for a Python developer with experience in NLP, Machine Learning, and data analysis."""

resumes = [

    "Python developer with machine learning and NLP experience",

    "Web developer skilled in HTML CSS JavaScript",

    "Data analyst with Python and data visualization skills"

]

vectorizer = TfidfVectorizer(stop_words='english')

vectors = vectorizer.fit_transform([job_desc] + resumes)

similarity = cosine_similarity(vectors[0:1], vectors[1:])

for i, score in enumerate(similarity[0]):

    print(f"Resume {i+1} Match Score: {score:.2f}")
```

OUTPUT:

```
Resume 1 Match Score: 0.78
Resume 2 Match Score: 0.12
Resume 3 Match Score: 0.45
```

Result:

Thus the resume screening process is implemented and the output is shown successfully.

Ex. No 2 DEVELOPMENT OF SENTIMENT ANALYSIS WITH PYTHON

AIM:

To develop a sentiment analysis program using Python that determines whether a given text expresses positive, negative, or neutral sentiment based on polarity score.

PROGRAM:

```
from textblob import TextBlob

text = "The movie was amazing and very interesting"

analysis = TextBlob(text)
print("Sentiment Polarity:", analysis.sentiment.polarity)

if analysis.sentiment.polarity > 0:
    print("Positive sentiment")
elif analysis.sentiment.polarity < 0:
    print("Negative sentiment")
else:
    print("Neutral sentiment")
```

OUTPUT

```
Sentiment Polarity: 0.6
Positive sentiment
```

Result:

Thus the Sentiment analysis process is implemented and the output is shown successfully.

EX. NO 3 KEYWORD EXTRACTION USING PYTHON

Aim:

To implement a keyword extraction system using Python and NLP techniques to identify and extract important keywords from a given text by removing stopwords.

PROGRAM

```
import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

text = "Natural Language Processing is an important field of Artificial Intelligence"

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

words = word_tokenize(text.lower())

keywords = [word for word in words if word.isalnum() and word not in stop_words]

print("Keywords:", keywords)
```

Output

```
Keywords: ['natural', 'language', 'processing', 'important', 'field', 'artificial', 'intelligence']
```

RESULT

Thus the Sentiment analysis process is implemented and the output is shown successfully.

EX. NO: 4 NLP for Other Languages (Tamil / Hindi)

Aim:

To design and implement an NLP-based program using Python to detect the language of non-English text such as Tamil or Hindi.

Program:

```
from textblob import TextBlob
text = "நான் இன்று மிகவும் மகிழ்ச்சியாக இருக்கிறேன்"
blob = TextBlob(text)
print("Original Text:", text)
print("Detected Language:", blob.detect_language())
```

OUTPUT:

```
Original Text: நான் இன்று மிகவும் மகிழ்ச்சியாக இருக்கிறேன்

Detected Language: ta
```

RESULT:

Thus NLP-based program using Python to detect the language of non-English text such as Tamil or Hindi is done and the output is verified successfully.

EX. NO: 5 NLP for WhatsApp Chat Analysis

Aim:

To analyze WhatsApp chat data using Python and NLP techniques in order to identify frequently used words and understand communication patterns.

Program:

```
import re
from collections import Counter
with open("chat.txt", encoding="utf-8") as f:
    chat = f.read()
messages = re.findall(r": (.+)", chat)
words = " ".join(messages).lower().split()
common_words = Counter(words).most_common(10)
print("Most common words:", common_words)
```

Output:

```
Most common words: [('hi', 10), ('hello', 7), ('ok', 5), ('yes', 4), ('thanks', 3)]
Frequently used words in the WhatsApp chat are identified successfully.
```

Result:

Thus the WhatsApp chat data using Python and NLP techniques in order to identify frequently used words and understand communication patterns is done and the output is shown successfully.

EX. NO: 6 CHATBOT IMPLEMENTATION (RULE-BASED)

Aim:

To design and implement a simple rule-based chatbot using Python that interacts with users and responds to predefined queries.

Program:

```
def chatbot(user_input):
    user_input = user_input.lower()

    if "hello" in user_input:
        return "Hi! How can I help you?"
    elif "name" in user_input:
        return "I am a simple chatbot."
    elif "bye" in user_input:
        return "Goodbye!"
    else:
        return "Sorry, I didn't understand."

while True:
    user = input("You: ")
    if user.lower() == "exit":
        break
    print("Bot:", chatbot(user))
```

OUTPUT:

```
You: hello

Bot: Hi! How can I help you?



You: name

Bot: I am a simple chatbot.



You: bye

Bot: Goodbye!
```

Result:

Thus chatbot is implemented and the output is shown successfully.

EX. NO: 7 NEXT WORD PREDICTION MODEL (N-GRAM MODEL)

Aim:

To implement a next-word prediction system using Python based on the N-gram language model to predict the most probable next word in a given sentence.

Program:

```
from collections import defaultdict
import random
text = "machine learning is fun and machine learning is powerful"
words = text.split()
ngrams = defaultdict(list)
for i in range(len(words)-1):
    ngrams[words[i]].append(words[i+1])
def predict_next(word):
    if word in ngrams:
        return random.choice(ngrams[word])
    else:
        return "No prediction"
print("Next word:", predict_next("machine"))
```

OUTPUT:

```
Next word: learning
(Output may vary because prediction is random)
```

Inference

The model predicts the next word using a bigram language model.

Result:

Thus the next-word prediction system using Python based on the N-gram language model to predict the most probable next word in a given sentence is executed and the output is verified successfully.