LABORATORY REPORT

Application Development Lab (CS33002)

B.Tech Program in ECSc

Submitted By

Name: Shreyaa Venkateswaran

Roll No: 2230120

Kalinga Institute of Industrial Technology
(Deemed to be University)
Bhubaneswar, India

Spring 2024-2025

Table of Content

Experiment Number	8
Experiment Title	Sentiment Prediction API Using FastAPI and X (formerly Twitter) Tweets
Date of Experiment	18-03-2025
Date of Submission	21-03-2025

1. Objective:

The objective of this lab experiment is to create a sentiment prediction API using FastAPI, which analyzes Twitter tweets for positive, negative, or neutral sentiment. This labintegrates natural language processing (NLP) techniques with a lightweight and high-performing API.

2. Procedure:

1. Install the required Python libraries: FastAPI, Tweepy, TextBlob, scikit-learn, pandas, and uvicorn.

2. Create an X Developer account.

3. Create a new application to obtain API keys

4. Use the Tweepy library to authenticate with the Twitter API.

5. Write a function to search for tweets containing a specific keyword or hashtag.

6. Fetch a specified number of recent tweets and return their text and metadata.

7. Use TextBlob or a similar NLP library to perform sentiment analysis on tweet text.

8. Define categories for sentiment (e.g., Positive, Negative, Neutral) based on the polarity score.

9. Create a function that takes text as input and returns the sentiment category.

10. Initialize a FastAPI application.

11. Define endpoints:

    1. A root endpoint (e.g., /) to confirm the API is running.
    2. A POST endpoint (e.g., /fetch_tweets/) to accept user inputs such as keyword and number of tweets to fetch.

12. Ensure the /fetch_tweets/ endpoint integrates the tweet-fetching and sentiment analysis functions.

13. Run the API using uvicorn in development mode (--reload flag for auto- updates).

14. Use a tool like Postman, CURL, or a web browser to test:

    1. The root endpoint for a welcome message.
    2. The POST endpoint by providing a sample keyword and tweet count in request payload.

15. Verify the output includes fetched tweets with their respective sentiment analysis.

3. Code:

app.py:

```python
import os
import tweepy
from fastapi import FastAPI, HTTPException
from fastapi.responses import FileResponse
from pydantic import BaseModel
from textblob import TextBlob
import uvicorn
from dotenv import load_dotenv

load_dotenv()
app = FastAPI()

API_KEY = os.getenv("API_KEY")
API_SECRET = os.getenv("API_SECRET")
ACCESS_TOKEN = os.getenv("ACCESS_TOKEN")
```

```
ACCESS_SECRET = os.getenv("ACCESS_SECRET")

auth = tweepy.OAuthHandler(API_KEY, API_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_SECRET)
api = tweepy.API(auth, wait_on_rate_limit=True)

class TweetRequest(BaseModel):
    keyword: str
    count: int = 10

def analyze_sentiment(text: str) -> str:
    analysis = TextBlob(text)
    polarity = analysis.sentiment.polarity
    if polarity > 0:
        return "Positive"
    elif polarity < 0:
        return "Negative"
    else:
        return "Neutral"

def fetch_tweets(keyword: str, count: int):
    try:
        tweets = api.search_tweets(q=keyword, count=count,
lang="en", tweet_mode="extended")
        results = []
        for tweet in tweets:
            sentiment = analyze_sentiment(tweet.full_text)
            results.append({
                "text": tweet.full_text,
                "sentiment": sentiment
            })
        return results
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))

@app.get("/")
def root():
    return {"message": "Sentiment Prediction API is running!"}

@app.post("/fetch_tweets/")
def get_tweets(request: TweetRequest):
    return fetch_tweets(request.keyword, request.count)
```

```
@app.get("/frontend")
def serve_frontend():
    return FileResponse("index.html")

if __name__ == "__main__":
    uvicorn.run(app, host="0.0.0.0", port=8000, reload=True)
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sentiment Pulse | Twitter Analysis</title>
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600;700&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <style>
        :root {
            --primary: #4361ee;
            --secondary: #3f37c9;
            --positive: #4cc9f0;
            --neutral: #f8961e;
            --negative: #f94144;
            --light: #f8f9fa;
            --dark: #212529;
        }

        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body {
            font-family: 'Poppins', sans-serif;
```

```
                  background: linear-gradient(135deg, #f5f7fa 0%,
#c3cfe2 100%);
            min-height: 100vh;
            padding: 2rem;
            color: var(--dark);
        }

        .container {
            max-width: 800px;
            margin: 2rem auto;
            background: white;
            border-radius: 20px;
            box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);
            overflow: hidden;
            animation: fadeIn 0.5s ease-out;
        }

        @keyframes fadeIn {
            from { opacity: 0; transform: translateY(20px); }
            to { opacity: 1; transform: translateY(0); }
        }

        header {
             background: linear-gradient(to right, var(--primary),
var(--secondary));
            color: white;
            padding: 2rem;
            text-align: center;
            position: relative;
        }

        header h1 {
            font-size: 2.5rem;
            margin-bottom: 0.5rem;
        }

        header p {
            opacity: 0.9;
            font-weight: 300;
        }

        .logo {
            position: absolute;
            top: 20px;
            left: 20px;
            font-size: 1.5rem;
            color: white;
        }

        .input-section {
            padding: 2rem;
            background: white;
        }

        .input-group {
            margin-bottom: 1.5rem;
        }

        label {
            display: block;
            margin-bottom: 0.5rem;
            font-weight: 600;
            color: var(--dark);
        }

        input {
            width: 100%;
            padding: 15px;
            border: 2px solid #e9ecef;
            border-radius: 10px;
            font-size: 1rem;
            transition: all 0.3s ease;
        }

        input:focus {
            border-color: var(--primary);
            outline: none;
            box-shadow: 0 0 0 3px rgba(67, 97, 238, 0.2);
        }

        button {
            width: 100%;
            padding: 15px;
             background: linear-gradient(to right, var(--primary),
var(--secondary));
            color: white;
```

```css
            border: none;
            border-radius: 10px;
            font-size: 1.1rem;
            font-weight: 600;
            cursor: pointer;
            transition: all 0.3s ease;
            display: flex;
            align-items: center;
            justify-content: center;
            gap: 10px;
        }

        button:hover {
            transform: translateY(-2px);
            box-shadow: 0 5px 15px rgba(67, 97, 238, 0.3);
        }

        button:active {
            transform: translateY(0);
        }

        .results {
            padding: 0 2rem 2rem;
            max-height: 500px;
            overflow-y: auto;
        }

        .tweet {
            background: white;
            border-radius: 12px;
            padding: 1.5rem;
            margin-bottom: 1rem;
            box-shadow: 0 3px 10px rgba(0, 0, 0, 0.05);
            border-left: 4px solid;
            transition: all 0.3s ease;
            animation: slideIn 0.5s ease-out;
            animation-fill-mode: both;
        }

        @keyframes slideIn {
            from { opacity: 0; transform: translateX(-20px); }
            to { opacity: 1; transform: translateX(0); }
        }
```

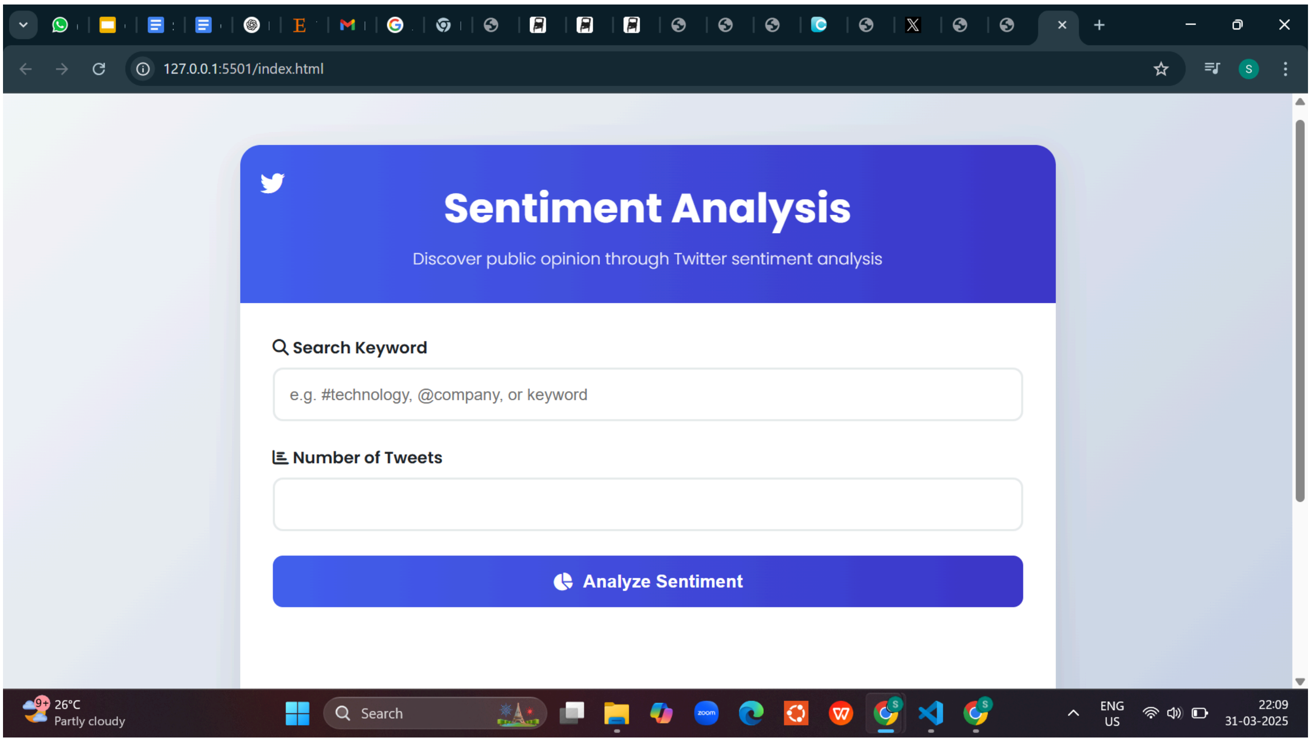
```
    .tweet:hover {
        transform: translateY(-3px);
        box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
    }

    .tweet.positive {
        border-color: var(--positive);
    }

    .tweet.neutral {
        border-color: var(--neutral);
    }

    .tweet.negative {
        border-color: var(--negative);
    }

    .sentiment {
        display: inline-block;
        padding: 5px 10px;
        border-radius: 20px;
        font-size: 0.8rem;
        font-weight: 600;
        margin-bottom: 10px;
        color: white;
    }

    .positive .sentiment {
        background-color: var(--positive);
    }

    .neutral .sentiment {
        background-color: var(--neutral);
    }

    .negative .sentiment {
        background-color: var(--negative);
    }

    .tweet-text {
        margin-bottom: 10px;
        line-height: 1.5;
    }

    .tweet-meta {
        display: flex;
        align-items: center;
        gap: 10px;
        font-size: 0.9rem;
        color: #6c757d;
    }

    .loading {
        display: none;
        text-align: center;
        padding: 2rem;
    }

    .spinner {
        width: 50px;
        height: 50px;
        border: 5px solid rgba(67, 97, 238, 0.2);
        border-radius: 50%;
        border-top-color: var(--primary);
        animation: spin 1s ease-in-out infinite;
        margin: 0 auto 1rem;
    }

    @keyframes spin {
        to { transform: rotate(360deg); }
    }

    .stats {
        display: flex;
        justify-content: space-around;
        margin-bottom: 2rem;
        text-align: center;
    }

    .stat-card {
        background: white;
        padding: 1.5rem;
        border-radius: 12px;
        box-shadow: 0 3px 10px rgba(0, 0, 0, 0.05);
        flex: 1;
        margin: 0 10px;
        transition: all 0.3s ease;
    }

    .stat-card:hover {
        transform: translateY(-5px);
        box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
    }

    .stat-value {
        font-size: 2rem;
        font-weight: 700;
        margin: 10px 0;
    }

    .positive-stat {
        color: var(--positive);
    }

    .neutral-stat {
        color: var(--neutral);
    }

    .negative-stat {
        color: var(--negative);
    }

    .empty-state {
        text-align: center;
        padding: 3rem;
        color: #6c757d;
    }

    .empty-state i {
        font-size: 3rem;
        margin-bottom: 1rem;
        opacity: 0.5;
    }

    @media (max-width: 768px) {
        .container {
            margin: 1rem;
            border-radius: 15px;
        }

        header h1 {
            font-size: 2rem;
        }

        .stats {
            flex-direction: column;
        }

        .stat-card {
            margin: 10px 0;
        }
    }
</style>
</head>
<body>
<div class="container">
    <header>
        <div class="logo">
            <i class="fab fa-twitter"></i>
        </div>
        <h1>Sentiment Analysis</h1>
        <p>Discover public opinion through Twitter sentiment
analysis</p>
    </header>

    <div class="input-section">
        <div class="input-group">
            <label for="keyword"><i class="fas
fa-search"></i> Search Keyword</label>
            <input type="text" id="keyword" placeholder="e.g.
#technology, @company, or keyword" required>
        </div>

        <div class="input-group">
            <label for="count"><i class="fas
fa-chart-bar"></i> Number of Tweets</label>
            <input type="number" id="count" value="10"
min="1" max="100" required>
        </div>

        <button onclick="analyzeSentiment()">
```

```
                <i class="fas fa-chart-pie"></i> Analyze Sentiment
            </button>
        </div>

        <div class="loading" id="loading">
            <div class="spinner"></div>
            <p>Analyzing tweets...</p>
        </div>

        <div class="stats" id="stats" style="display: none;">
            <div class="stat-card">
                <h3>Positive</h3>
                <div class="stat-value positive-stat" id="positive-count">0</div>
                <p>Tweets</p>
            </div>
            <div class="stat-card">
                <h3>Neutral</h3>
                <div class="stat-value neutral-stat" id="neutral-count">0</div>
                <p>Tweets</p>
            </div>
            <div class="stat-card">
                <h3>Negative</h3>
                <div class="stat-value negative-stat" id="negative-count">0</div>
                <p>Tweets</p>
            </div>
        </div>

        <div class="results" id="results">
            <div class="empty-state">
                <i class="fas fa-comment-dots"></i>
                <h3>No analysis yet</h3>
                <p>Enter a keyword and click "Analyze Sentiment" to see results</p>
            </div>
        </div>
    </div>

    <script>
        async function analyzeSentiment() {
            const keyword = document.getElementById("keyword").value;
            const count = document.getElementById("count").value;

            if (!keyword) {
                alert("Please enter a keyword to analyze");
                return;
            }

            // Show loading state
                document.getElementById("loading").style.display = "block";
            document.getElementById("results").innerHTML = "";
                  document.getElementById("stats").style.display = "none";

            try {
                const response = await fetch("http://127.0.0.1:8000/fetch_tweets/", {
                    method: "POST",
                    headers: { "Content-Type": "application/json" },
                    body: JSON.stringify({ keyword: keyword, count: parseInt(count) })
                });

                const data = await response.json();
                displayResults(data);
            } catch (error) {
                console.error("Error:", error);
                document.getElementById("results").innerHTML = `
                    <div class="empty-state">
                        <i class="fas fa-exclamation-triangle"></i>
                        <h3>Error loading data</h3>
                        <p>${error.message || "Please try again later"}</p>
                    </div>
                `;
            } finally {
                  document.getElementById("loading").style.display = "none";
            }
```

```
        }

        function displayResults(data) {
            const resultDiv = document.getElementById("results");
            const statsDiv = document.getElementById("stats");

            if (!data || data.length === 0) {
                resultDiv.innerHTML = `
                    <div class="empty-state">
                        <i class="fas fa-comment-slash"></i>
                        <h3>No tweets found</h3>
                             <p>Try a different keyword or search
term</p>
                    </div>
                `;
                return;
            }

            // Calculate stats
            let positive = 0, neutral = 0, negative = 0;
            data.forEach(tweet => {
                if (tweet.sentiment === "positive") positive++;
                         else if (tweet.sentiment === "neutral")
neutral++;
                else negative++;
            });

            // Update stats
             document.getElementById("positive-count").textContent
= positive;
              document.getElementById("neutral-count").textContent
= neutral;
             document.getElementById("negative-count").textContent
= negative;
            statsDiv.style.display = "flex";

            // Display tweets
            resultDiv.innerHTML = "";
            data.forEach((tweet, index) => {
                                           const tweetElement =
document.createElement("div");
                                 tweetElement.className = `tweet
${tweet.sentiment}`;
```

```
                tweetElement.innerHTML = `
                                                                  <div
class="sentiment">${tweet.sentiment.toUpperCase()}</div>
                    <p class="tweet-text">${tweet.text}</p>
                    <div class="tweet-meta">
                                <span><i class="far fa-user"></i>
${tweet.username || "Unknown"}</span>
                         <span><i class="far fa-calendar-alt"></i>
${tweet.date || ""}</span>
                    </div>
                `;
                    tweetElement.style.animationDelay = `${index *
0.1}s`;
                resultDiv.appendChild(tweetElement);
            });
        }
    </script>
</body>
</html>
```

4. Results/Output:



127.0.0.1:5501/index.html

Sentiment Analysis

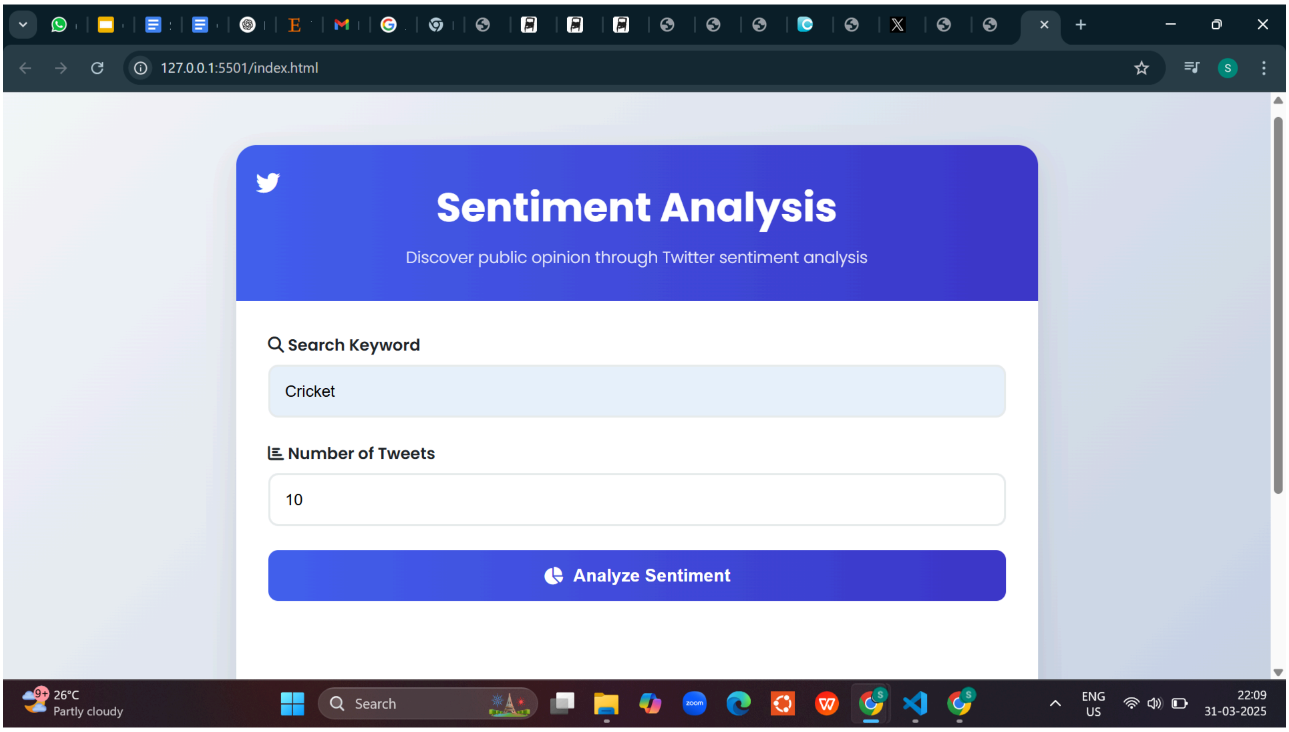
Discover public opinion through Twitter sentiment analysis

Search Keyword

e.g. #technology, @company, or keyword

Number of Tweets

Analyze Sentiment



127.0.0.1:5501/index.html

Sentiment Analysis

Discover public opinion through Twitter sentiment analysis

Search Keyword

Cricket

Number of Tweets

10

5. Remarks:

Created a sentiment analysis prediction API using FastAPI which analyses the sentiment of twitter tweets. However there is an error in retrieving the data as X is denying access even though the API was obtained through X developer mode. Otherwise the code should work properly.

Website link: Twitter_Sentiment_Analysis

GitHub link: GitHub

Shreyaa Venkateswaran

Signature of the Lab Coordinator