System Integration (ITIS-6177)

Final Project

Azure- Cognitive Services for Language (Sentiment Analysis API)

-Pallavi Bhadri
801276258

Introduction:

This project uses NodeJs express to provide APIs that can communicate with the Microsoft Azure service while also requiring authentication for cognitive services. You must first create or log into an Azure account before continuing. You might be able to receive the service for free if you already have an account.

How to Sign Up through the Azure Portal:

- Visit https://azure.microsoft.com/en-us/products/cognitive-services/language-service/
- Use your school email address to establish a subscription to receive free credit.
- Select Continue to develop your resource on the Select more features page that displays.
- **Enter the following data on the Create language screen:**
    - ➢ Subscription
    - ➢ Resrource Group
    - ➢ Region
    - ➢ Name
    - ➢ Pricing Tier
    - ➢ The Responsible AI Notice option must be selected.
    - ➢ At the bottom of the page, click Review + Create.
    - ➢ Verify that your information was submitted accurately and that the validation has been successful on the screen that shows.
    - ➢ Next, choose Create.
- Creating a resource group, Choose the "Cognitive services" area.
- To connect the file path across Azure semantic analysis services, look for and create a "cognitive Services" resource in the resource group.
- A key and an endpoint that are ready for use will be produced.

What is sentiment analysis?

Each statement in the API receives a sentiment prediction for each of the three sentiment classes (Positive, Negative, and Neutral). By looking for indications of positive or negative emotion in raw text, you may learn what customer thinks about your brand or subject. For each document, this API gives a sentiment score between 0 and 1, with 1 being the highest favorable opinion.

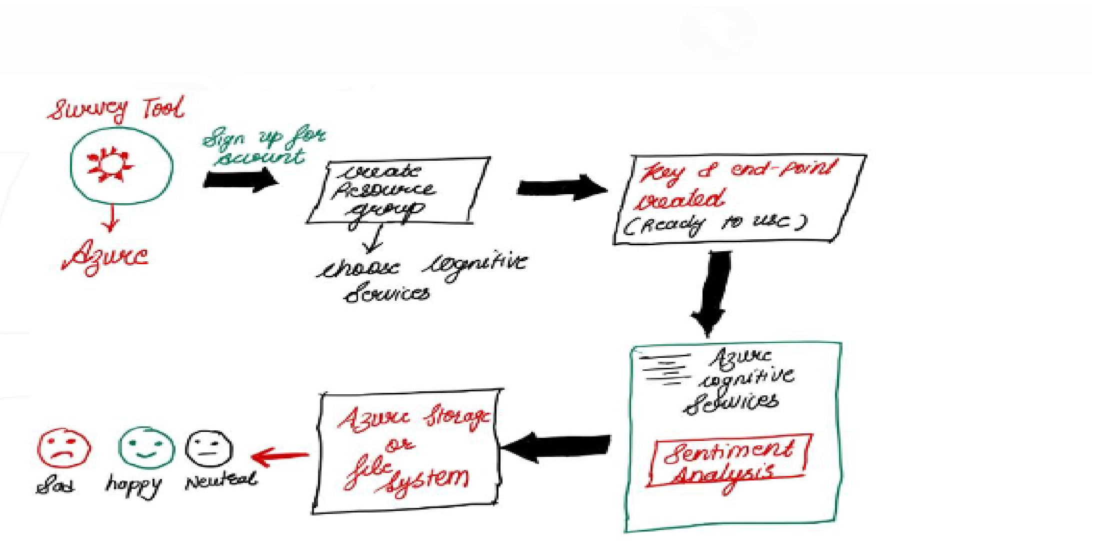
As more data from social networks, particularly Twitter, became available, sentiment analysis engines first debuted in the early 2000s and gained popularity.

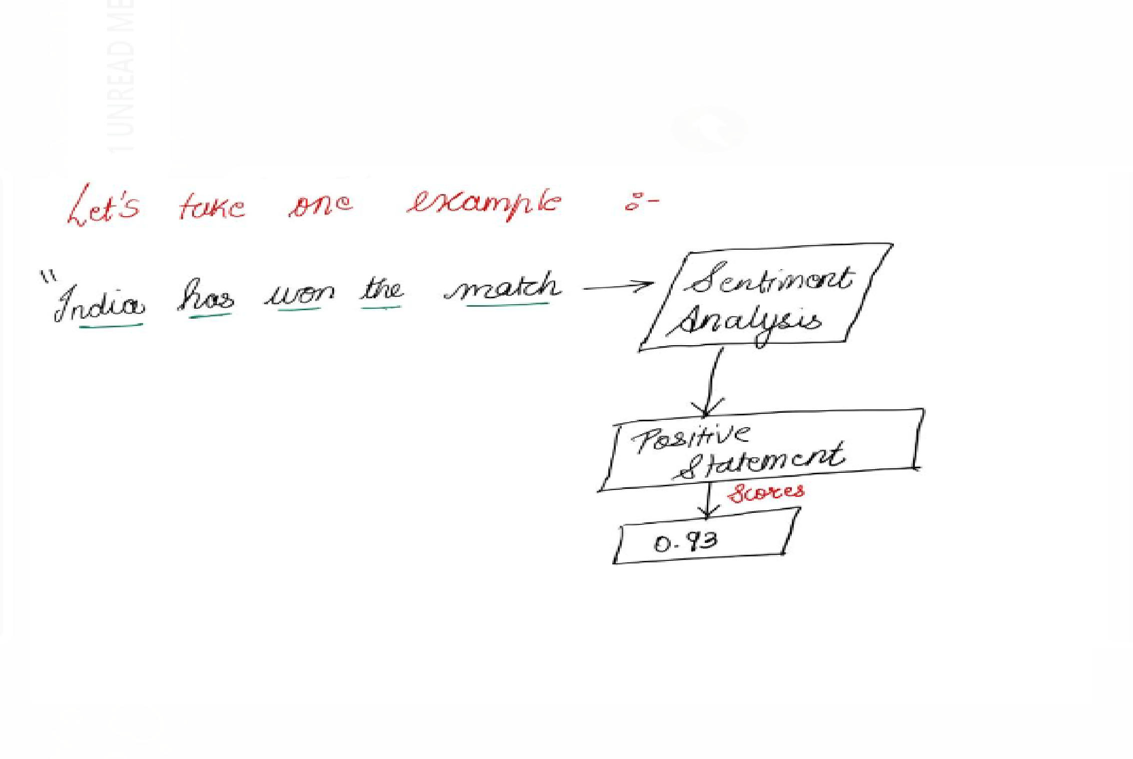
To extract subjective information from online material, such as texts, tweets, blogs, social media posts, news stories, reviews, and comments, sentiment analysis is becoming a common practice.

In order to better grasp how it functions, let's look at an example:

Sentiment	Label
The document has at least one uplifting statement. The remaining clauses are all neutral.	Positive
The text contains at least one unfavorable sentence. The remaining clauses are all neutral.	Negative
In the text, every statement is neutral.	Neutral

Sentiment Example	Label
http://142.93.49.248:3001/sentimentanalysis?keyword=%22UNC%20Charlotte%20has%20won%20the%20match%22	Positive(UNC Charlotte won the match, shows the positive clause "WON")
http://142.93.49.248:3001/sentimentanalysis?keyword=%E2%80%9DI%20am%20sad%20today%E2%80%9D	Negative(I am Sad today, shows the negative clause" SAD")





Installation:

- To execute, [Node.js](https://nodejs.org/) is needed for my project.
- Start the server after installing the dependencies with npm install.

Dependencies used in this project:

- Javascript's compiler - NodeJs
- Running web server and API framework: Express js.
- Microsoft Azure's Cognitive Services feature
- Node.js package manager are used by users.

Status Code:

☐ 200: OK
☐ 401: Bad Error

How to consume this Semantic Analysis API?

- **Host Address:** 142.93.49.248
- **Port:** 3001

Request URL:

http://142.93.49.248:3001/sentimentanalysis?keyword=%22UNC%20Charlotte%20has%20won%20the%20match%22

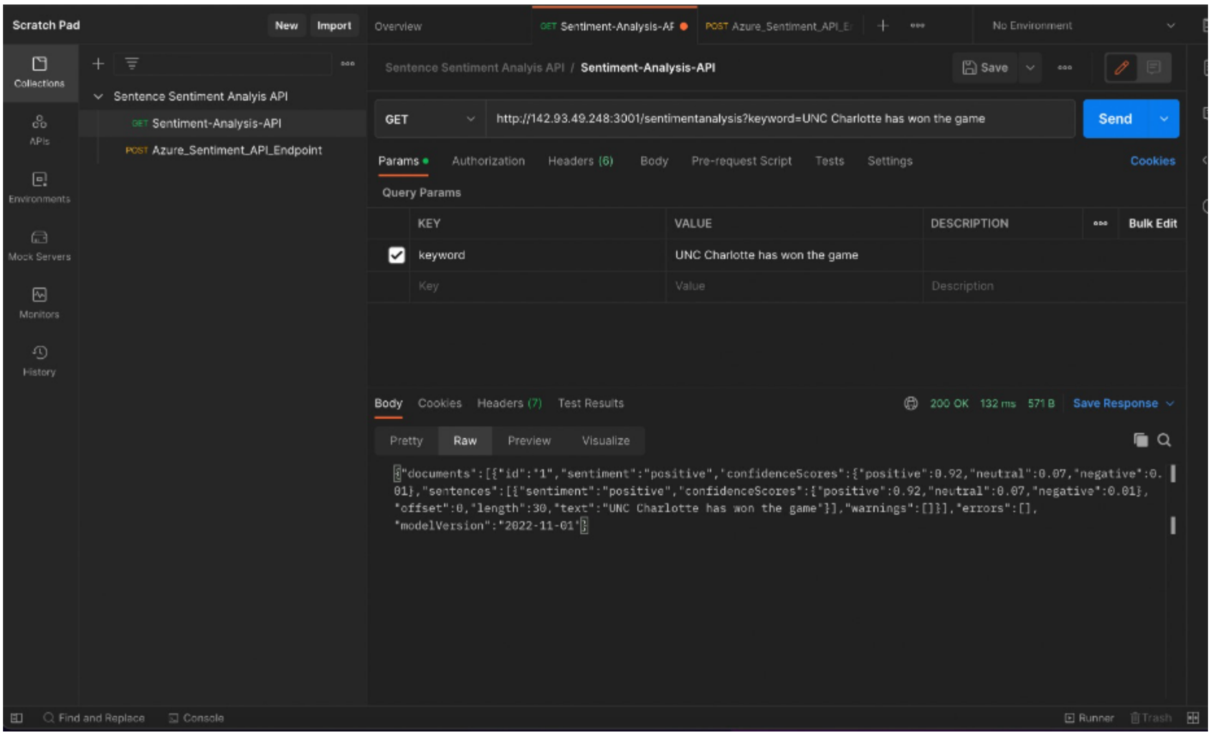
Request Header:

```
        "content-type": "application/json",
```

```
        "ocp-apim-subscription-key": process.env.token,
```

Sentiment Analysis API :Postman

Users can use query params as a keyword and value to deliver the text to the API. As an input parameter, text is supplied. Let's imagine the user writes, "UNC Charlotte won the match," which will be interpreted **positively** and may appear as follows in the body:



GET http://142.93.49.248:3001/sentimentanalysis?keyword=UNC Charlotte has won the game

	KEY	VALUE	DESCRIPTION
☑	keyword	UNC Charlotte has won the game	
	Key	Value	Description

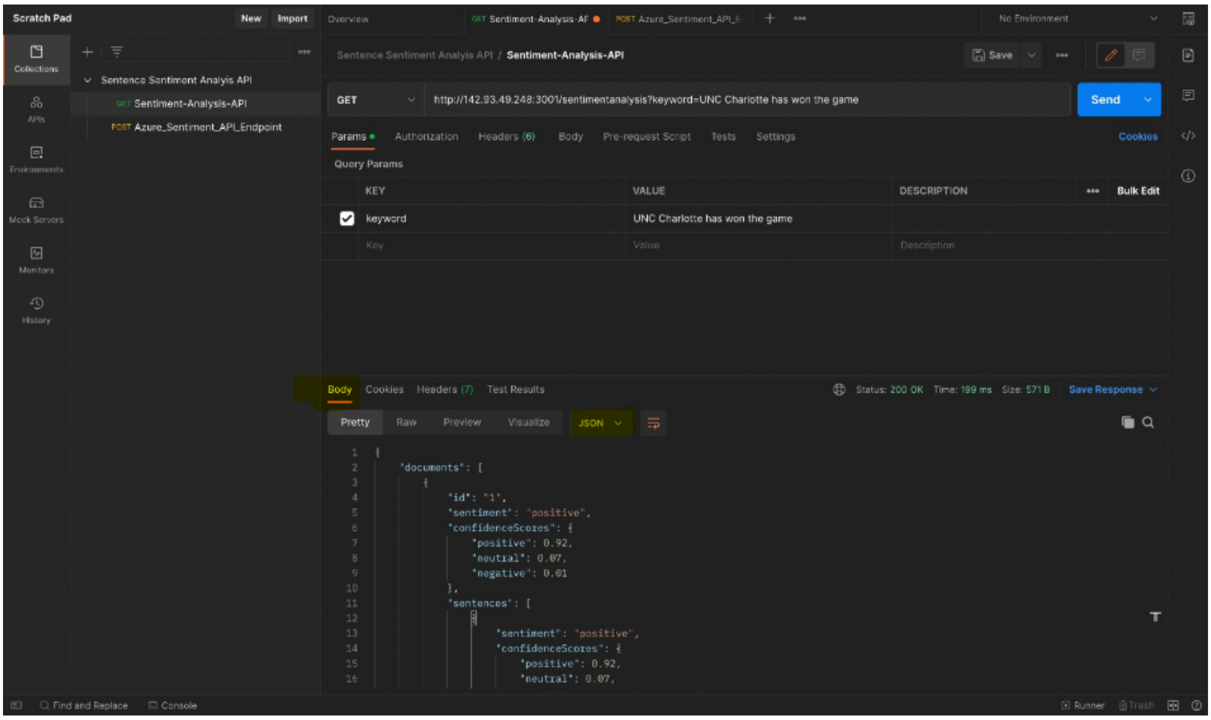
200 OK 132 ms 571 B

```
{"documents":[{"id":"1","sentiment":"positive","confidenceScores":{"positive":0.92,"neutral":0.07,"negative":0.
01},"sentences":[{"sentiment":"positive","confidenceScores":{"positive":0.92,"neutral":0.07,"negative":0.01},
"offset":0,"length":30,"text":"UNC Charlotte has won the game"}],"warnings":[]}],"errors":[],
"modelVersion":"2022-11-01"}
```

Users can also view the body in JSON Format by selecting the JSON from the drop-down button and can get the following output for the same:

OUTPUT

```
{
   "documents": [
       {
            "id": "1",
            "sentiment": "positive",
            "confidenceScores": {
                "positive": 0.92,
                "neutral": 0.07,
                "negative": 0.01
            },
            "sentences": [
                {
                    "sentiment": "positive",
                    "confidenceScores": {
                        "positive": 0.92,
                        "neutral": 0.07,
                        "negative": 0.01
                    },
                    "offset": 0,
                    "length": 30,
                    "text": "UNC Charlotte has won the game"
                }
            ],
            "warnings": []
        }
    ],
    "errors": [],
    "modelVersion": "2022-11-01"
}
```



GET http://142.93.49.248:3001/sentimentanalysis?keyword=UNC Charlotte has won the game

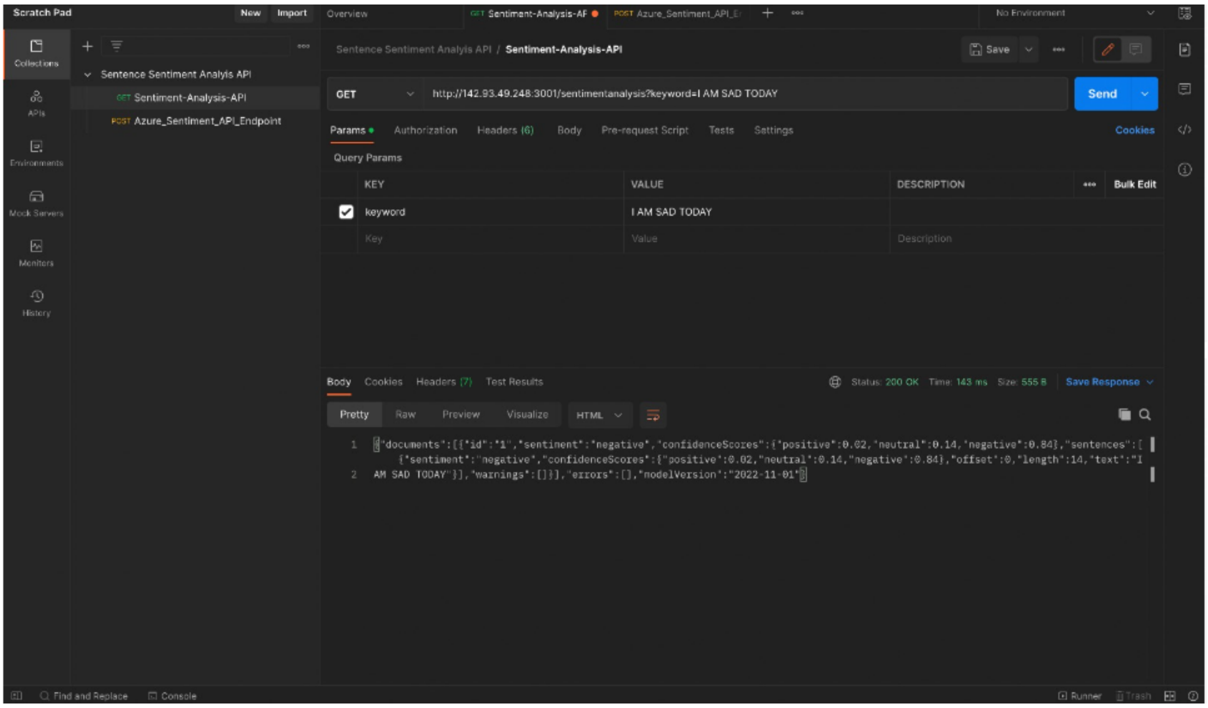
	KEY	VALUE	DESCRIPTION
☑	keyword	UNC Charlotte has won the game	
	Key	Value	Description

Status: 200 OK Time: 199 ms Size: 571 B

```
{
    "documents": [
        {
            "id": "1",
            "sentiment": "positive",
            "confidenceScores": {
                "positive": 0.92,
                "neutral": 0.07,
                "negative": 0.01
            },
            "sentences": [
                {
                    "sentiment": "positive",
                    "confidenceScores": {
                        "positive": 0.92,
                        "neutral": 0.07,
```

Let’s take one more example:

Users can use query params as a keyword and value to deliver the text to the API. As an input parameter, text is supplied. Let's imagine a user writes, "I feel sad today," which will be interpreted **negatively** and may appear as follows in the body:



GET http://142.93.49.248:3001/sentimentanalysis?keyword=I AM SAD TODAY

Query Params

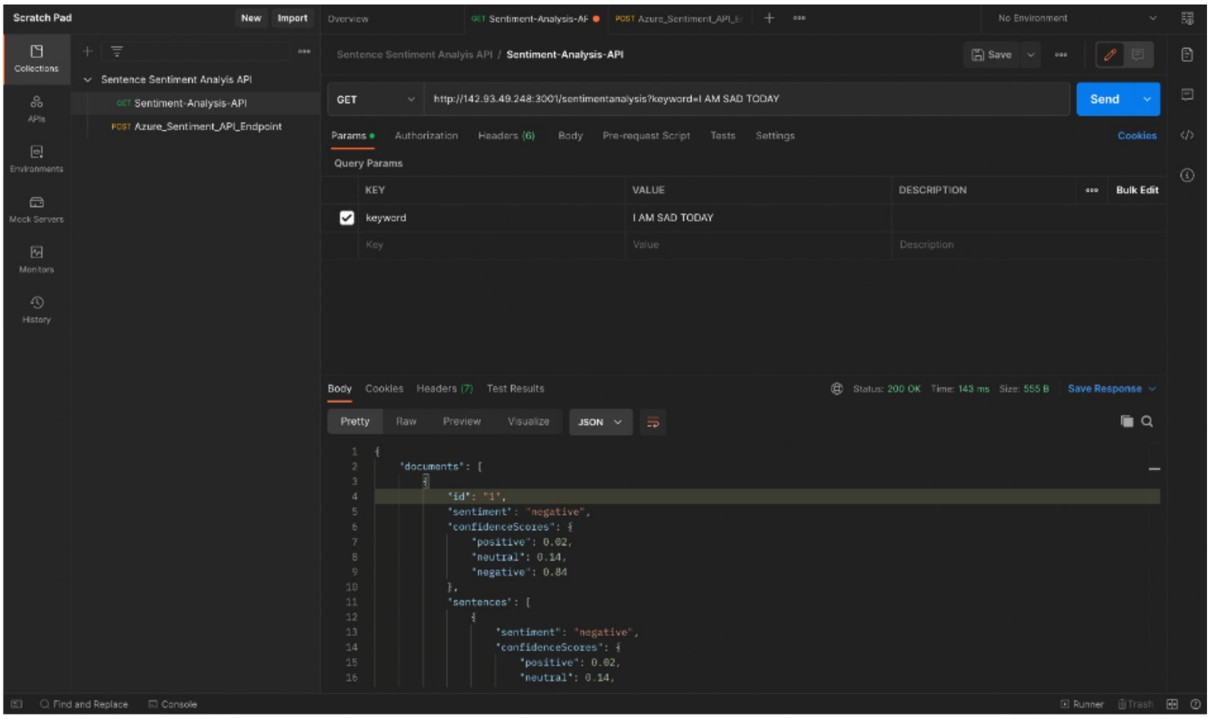
	KEY	VALUE	DESCRIPTION
☑	keyword	I AM SAD TODAY	
	Key	Value	Description

Status: 200 OK Time: 143 ms Size: 555 B

```
{"documents":[{"id":"1","sentiment":"negative","confidenceScores":{"positive":0.02,"neutral":0.14,"negative":0.84},"sentences":[{"sentiment":"negative","confidenceScores":{"positive":0.02,"neutral":0.14,"negative":0.84},"offset":0,"length":14,"text":"I AM SAD TODAY"}],"warnings":[]}],"errors":[],"modelVersion":"2022-11-01"}
```

Just mentioned above in the previous example of POSITIVE, following **output** for the same is as below:

```
{
  "documents": [
    {
      "id": "1",
      "sentiment": "negative",
      "confidenceScores": {
        "positive": 0.02,
        "neutral": 0.14,
        "negative": 0.84
      },
      "sentences": [
        {
          "sentiment": "negative",
          "confidenceScores": {
            "positive": 0.02,
            "neutral": 0.14,
                    "negative": 0.84
                },
                "offset": 0,
                "length": 14,
                "text": "I AM SAD TODAY"
            }
        ],
        "warnings": []
    }
],
"errors": [],
"modelVersion": "2022-11-01"
}
```



GET http://142.93.49.248:3001/sentimentanalysis?keyword=I AM SAD TODAY

	KEY	VALUE	DESCRIPTION
☑	keyword	I AM SAD TODAY	
	Key	Value	Description

Status: 200 OK Time: 143 ms Size: 555 B

```
{
    "documents": [
        {
            "id": "1",
            "sentiment": "negative",
            "confidenceScores": {
                "positive": 0.02,
                "neutral": 0.14,
                "negative": 0.84
            },
            "sentences": [
                {
                    "sentiment": "negative",
                    "confidenceScores": {
                        "positive": 0.02,
                        "neutral": 0.14,
```

Conclusion:

What I discovered and learned specifically was:

Sentiment Analysis engines are widely accessible on the market; it is hard to know all of them or which ones work best. The multi-cloud strategy is the greatest technique we have to incorporate sentiment analysis technology since it ensures you'll get the finest results and costs based on our data and project. In this project, I utilized contextual services in Azure that displayed the text's emotions, whether they were positive or bad. These services also analyze user input and produce a sentiment score that ranges from 0 (negative) to 1 (positive) (positive). This aids in the analysis of customer reviews for any service.