

SENTIMENT ANALYSIS SYSTEM

1. Introduction

This project focuses on developing a Natural Language Processing (NLP) application capable of analyzing sentiment in text data such as customer reviews, social media comments and survey responses. It predicts the sentiment into 3 categories positive, negative or neutral and visualizes the results based on different factors such as age, gender, language and city

Objectives:

The main objectives of this project are:

- To implement an NLP application for sentiment analysis.
- To integrate Google Sheets API for data retrieval.
- To use NLTK and VADER Sentiment for sentiment analysis.
- To visualize sentiment analysis results using Plotly.
- To create an interactive interface using Streamlit.

2. Data Retrieval

Data collection is done through google sheets api. It involves creating a project in Google cloud console, configuring OAuth credentials and obtaining a client Id and client secrets

Data is loaded into the local machine by URL.

The modules and methods used for data retrieval are `google_auth_oauthlib`, `InstalledAppFlow`, `googleapiclient.discovery` and `build`

3. Methodology

The sentiment analysis is performed using NLTK and VADER Sentiment. NLTK is employed for natural language processing tasks, while VADER Sentiment provides a pre-trained model for sentiment analysis.

4. Implementation

Backend: Data Collection: Google sheets with py

Data Organization: Pandas

Data Analysis: nltk, vaderSentiment

Data Visualization: plotly

```
C:\Windows\System32\cmd.e x + v
(sentiment) C:\myprojects\sentiment\Scripts>backend.py
Please visit this URL to authorize this application: https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=541900502920-4u087270vkts5fpm2dc03h4am11r9v.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A52500%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&state=YTOSMiv1BOVLAutqzhsjFpzCbZhpq&access_type=offline
Traceback (most recent call last):
  File "C:\myprojects\sentiment\Scripts\backend.py", line 9, in <module>
    df=pd.DataFrame(data=d[1:],columns=d[0])
    ^^
NameError: name 'pd' is not defined. Did you mean: 'd'?

(sentiment) C:\myprojects\sentiment\Scripts>backend.py
Please visit this URL to authorize this application: https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=541900502920-4u087270vkts5fpm2dc03h4am11r9v.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A52564%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&state=kvGBdR8cQEXv2DIBoUvshDCRUSPhVh&access_type=offline
Timestamp      Name Age  Gender Language Opinion
0 11/16/2023 18:00:04  Sirisha  20  Female  English  Good

(sentiment) C:\myprojects\sentiment\Scripts>backend.py
Please visit this URL to authorize this application: https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=541900502920-4u087270vkts5fpm2dc03h4am11r9v.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A52776%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&state=mBZsem3JLl0C0sUwq6oVjqYc0na1&access_type=offline
Timestamp      Name Age  Gender Language Opinion
0 11/16/2023 18:00:04  Sirisha  20  Female  English  Good
Good

(sentiment) C:\myprojects\sentiment\Scripts>
```

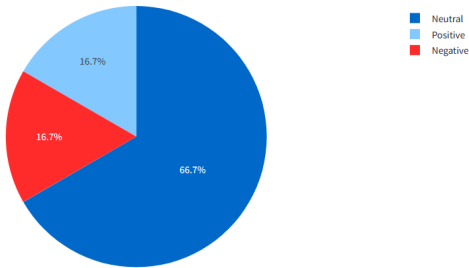
Frontend:Google form

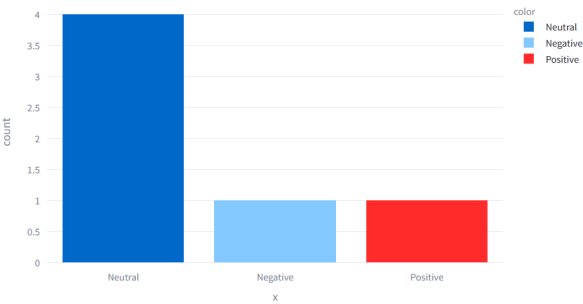
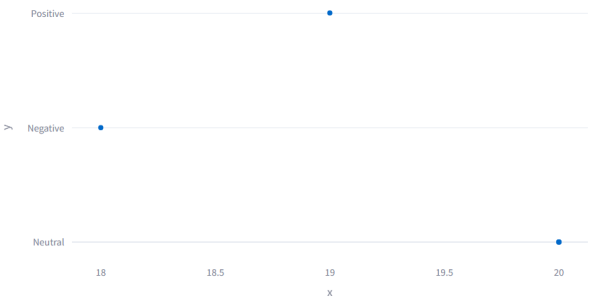
Web application:Streamlit

5. Data Analysis and Visualization

The sentiment analysis results are visualized using Plotly to create informative and interactive plots. These visualizations help in understanding the distribution of sentiments in the analyzed text data.

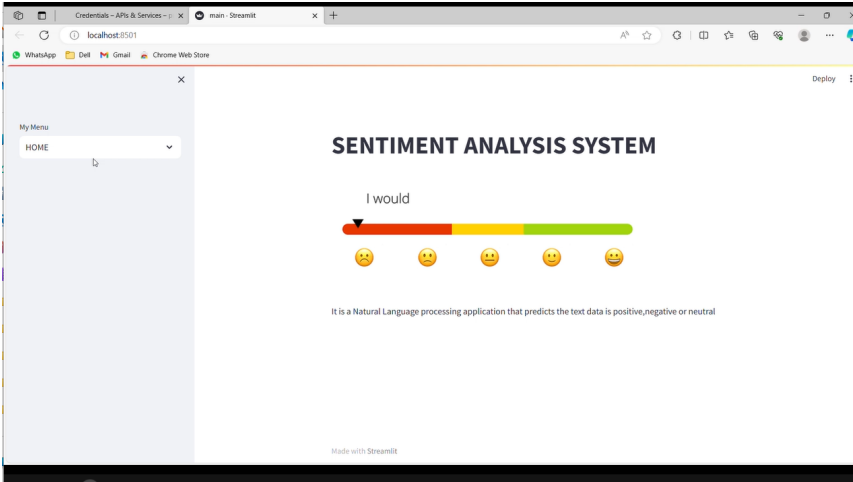
- Histogram for Sentiment Distribution
- Pie Chart for Overall Sentiment Composition
- Scatter Plot for Text Data and Sentiment Scores





6. User Interface

The user interface is developed using Streamlit, allowing users to interact with the application seamlessly. Users can input text data, and the application provides sentiment predictions along with visualizations.



7. Applications

Business Insights: Gain valuable insights into customer satisfaction, market reception, and brand perception by analyzing sentiments in customer feedback and reviews.

Social Media Management: Monitor and manage brand reputation on social media platforms by tracking sentiments in real-time, evaluating marketing campaign effectiveness, and addressing issues promptly.

Customer Service Enhancement: Improve automated customer service systems by integrating sentiment analysis into chatbots, enabling tailored responses and efficient issue resolution.

Market Research Optimization: Enhance market research efforts by evaluating sentiments around product launches, conducting competitor analysis, and understanding market dynamics.

Political and Public Opinion Analysis: Utilize sentiment analysis to gauge public opinion during elections, assess reactions to policy changes, and monitor sentiments in news articles and social media.

Healthcare Improvement: Improve patient experience and healthcare quality by analyzing sentiments in patient feedback, and contribute to mental health research through social media sentiment analysis.

Financial Decision Support: Inform financial decisions by analyzing sentiments in financial news and social media, predicting stock market trends, and detecting fraudulent activities in banking.

8. Conclusion

In conclusion, the NLP sentiment analysis application achieves the set objectives. It provides a robust solution for analyzing sentiments in text data and visualizing the results.

9. Acknowledgments

I would like to express my gratitude to my Trainer Mr.Abhinav whose inputs and suggestions provided valuable perspectives throughout the project's development.

10.Code

```
import streamlit as st

from google_auth_oauthlib.flow import InstalledAppFlow

from googleapiclient.discovery import build

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

import pandas as pd

import plotly.express as px
```

```

st.title("SENTIMENT ANALYSIS SYSTEM")

choice=st.sidebar.selectbox("My Menu",("HOME","ANALYSIS","RESULTS"))

if(choice=="HOME"):

    st.image("https://miro.medium.com/proxy/1*_JW1JaMpK_fVGld8pd1_JQ.gif")

    st.write("It is a Natural Language processing application that predicts the text data is
positive,negative or neutral")

elif(choice=="ANALYSIS"):

    sid=st.text_input("Enter your google sheet ID")

    r=st.text_input("Enter range between first column and last column")

    c=st.text_input("Enter column name that is to be analyzed")

    btn=st.button("Analyze")

    if btn:

        if 'cred' not in st.session_state:

            f=InstalledAppFlow.from_client_secrets_file("C:\myprojects\sentiment\Scripts\key.json.json",
["https://www.googleapis.com/auth/spreadsheets"])

            st.session_state['cred']=f.run_local_server(port=0)

        mymodel=SentimentIntensityAnalyzer()

        service=build("Sheets","v4",credentials=st.session_state['cred']).spreadsheets().values()

        k=service.get(spreadsheetId=sid,range=r).execute()

        d=k['values']

        df=pd.DataFrame(data=d[1:],columns=d[0])

        l=[]

        for i in range(0,len(df)):

            t=df._get_value(i,c)

            pred=mymodel.polarity_scores(t)

            if(pred['compound']>0.5):

                l.append("Positive")

            elif(pred['compound']<-0.5):

                l.append("Negative")

```

```

else:

    l.append("Neutral")

df["Sentiment"]=l

df.to_csv("results.csv",index=False)

st.subheader("The analysis results are saved by results.csv file")

elif(choice=="RESULTS"):

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

    choice2=st.selectbox("Choose visualization",("NONE","PIE CHART","HISTOGRAM","SCATTER
PLOT"))

    st.dataframe(df)

    if(choice2=="PIE CHART"):

        posper = (len(df[df['Sentiment'] == 'Positive']) / len(df)) * 100

        negper=(len(df[df['Sentiment']== 'Negative']) / len(df)) * 100

        neuper=(len(df[df['Sentiment']=='Neutral'])/len(df))*100

        fig = px.pie(values=[posper, negper, neuper], names=['Positive', 'Negative', 'Neutral'])

        st.plotly_chart(fig)

    elif(choice2=="HISTOGRAM"):

        k=st.selectbox("Choose column",df.columns)

        if k:

            fig=px.histogram(x=df[k],color=df['Sentiment'])

            st.plotly_chart(fig)

    elif(choice2=="SCATTER PLOT"):

        k=st.text_input("Enter the continuous column name")

        if k:

            fig=px.scatter(x=df[k],y=df['Sentiment'])

            st.plotly_chart(fig)

```

SENTIMENT ANALYSIS SYSTEM

Choose visualization

SCATTER PLOT					
<div><div></div><div></div><div></div></div>					
	Age	Gender	Language	Opinion	Sentiment
0	20	Female	English	Good	Neutral
1	20	Female	English	Nothing to say	Neutral
2	20	Male	English	Good	Neutral
3	20	Female	English	It is good and it	Neutral
4	18	Female	English	Worst	Negative
5	19	Female	English	Best	Positive