COMPARATIVE SENTIMENT ANALYSIS BY SCRAPING PUBLIC OPINIONS FROM VARIOUS SOCIAL MEDIA

A Project Report

Submitted by

ALLA BADRINATH REDDY [CB.EN.U4CSE17305]
RAGUL S V [CB.EN.U4CSE17348]
RAJASEKAR M [CB.EN.U4CSE17349]

Under the guidance of

Dr. Anantha Narayanan
(Asst. Professor(SG), Department of Computer Sciene & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENE & ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

Amrita Nagar PO, Coimbatore - 641 112, Tamilnadu

May 2021

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE – 641 112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **COMPARATIVE SENTIMENT ANALYSIS BY SCRAPING PUBLIC OPINIONS FROM VARIOUS SOCIAL MEDIA** is submitted by Alla Badrinath Reddy (CB.EN.U4CSE17305), Ragul S V (CB.EN.U4CSE17348), Rajasekar M (CB.EN.U4CSE17349), in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in Computer Science and Engineering is a bonafide record of the work carried out under our guidance and supervision at Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore.

SIGNATURE

Dr. Anantha Narayanan
PROJECT GUIDE
Asst. Professor(SG)
Dept. of Computer Sciene & Engineering

Signature of the Internal Examiner

SIGNATURE

Dr. D. KINGSLY JEBA SINGH
CHAIRPERSON
Dept. of Computer Sciene & Engineering

Signature of the External Examiner

DECLARATION

we the undersigned solemnly declare that the project report **COMPARATIVE SENTIMENT ANALYSIS BY SCRAPING PUBLIC OPINIONS FROM VARIOUS SOCIAL MEDIA** are based on my own work carried out during the course of our study under the supervision of Dr. Anantha Narayanan, Asst. Professor(SG), Computer Sciene & Engineering, and have not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgement have been made wherever the findings of others have been cited.

AB

Alla Badrinath Reddy [CB.EN.U4CSE17305]

Ragul S V [CB.EN.U4CSE17348]

M. Raj.

Rajasekar M [CB.EN.U4CSE17349]

ABSTRACT

The process of understanding and demarcating the sentiment of user's content by using Natural Language Processing Techniques is called Sentiment Analysis. In today's hyper-connected world, we have the technology to accumulate massive amount of public opinion on any subject through various forms like blogs, comments, reviews and posts. Applying Sentiment Analysis on these content can potentially help us measure the collective cumulative sentiment of a user in the given subject. Since, the requirement and application for such technology is seemingly increasing in recent times, we've got a lot of packages and libraries that are being developed for the cause of Sentiment Analysis. Our idea with this project is to democratise that Technology and present this Technology in an User-friendly manner to the non-experts of the same. We had collected the public opinion from social media like reddit and twitter based on the user-given keyword and had done sentiment analysis on top of that to give the user, the collective feedback, in the form of numbers and graphs.

ACKNOWLEDGEMENTS

We would like to express our deep gratitude to our beloved Satguru **Sri Mata Amritanandamayi Devi** for providing the bright academic climate at this university, which has made this entire task appreciable. This acknowledgement is intended to be a thanksgiving measure to all those people involved directly or indirectly with our project. We would like to thank our Pro Chancellor **Bramachari Abhayamrita Chaitanya**, Vice Chancellor **Dr. Venkat Rangan. P** and **Dr. Sasangan Ramanathan**, Dean Engineering of Amrita Vishwa Vidyapeetham for providing us the necessary infrastructure required for the completion of the project. We express our thanks to **Dr. (Col) P.N. Kumar**, Chairperson of Department of Computer Science Engineering, **Dr. G. Jeyakumar and Dr. C. Shunmuga Velayutham,** Vice Chairpersons of the Department of Computer Science and Engineering for their valuable help and support during our study. we express our gratitude to our guide, **Dr. Anantha Narayanan** (Assistant Professor) for his guidance, support and supervision. we feel extremely grateful to **Ms. Prathilothamai M.**(Assistant Professor) **Dr. Prashant R. Nair** (Associate Professor) **Ms. Siva Ratna Kumari Narisetti** (Assistant Professor) for their feedback and encouragement which helped us to complete the project. we would also like to thank the entire staff of the Department of Computer Science and Engineering. we would like to extend our sincere thanks to our family and friends for helping and motivating me during the course of the project. Finally, we would like to thank all those who have helped, guided and encouraged me directly or indirectly during the project work. Last but not the least, we thank God for His blessings which made our project a success.

Alla Badrinath Reddy [CB.EN.U4CSE17305]

Ragul S V [CB.EN.U4CSE17348]

M. Raj.

Rajasekar M [CB.EN.U4CSE17349]

TABLE OF CONTENTS

ABBREVIATIONS

NLP Natural Language Processing

API Application Programming Interface

HTML Hyper Text Markup Language

CSS Cascading Style Sheets

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Problem Definition

To create a Web Application that which will do a comparative sentiment analysis by scraping public opinions from various Social Media

Sentiment Analysis is the understanding the emotion behind the text. This project emphasizes on classifying the natural language text based on how much of the text positive, negative and neutral. The availability of a lot of public content in social media, blogs, product reviews and e-commerce sites have encouraged a lot of people to express and publish their opinion online. Social media is the biggest source of them all with massive amount of text content that can be scraped and used for Sentiment Analysis. The Data used in this project were scraped from Social Media such as Reddit and Twitter by using their respective APIs.

Chapter 2

LITERATURE SURVEY

2.1 Sentiment Analysis on Social Media (Neri et al., 2012)

The enabling of research, the analysis, the classification of great volumes of a multitude of documents, assisting the analysts with information labyrinth, analysing to take account of complexity of public views, the automated assigning of the sentiment polarity and the accessibility to all the potential texts of interests was achieved by the linguistic and semantic approach implemented in this project.

2.2 Mobile Network Operators' Support on Social Media: Sentiment Analysis and Natural Language Processing (Ogudo and Nestor, 2019)

This paper explores and infers that the sentiment score calculated from the opinions of people from a particular age group or a particular area or from a particular profession can change the evolution of the product in the future and how that could be a vital information when it comes to the business world.

2.3 Polarity Sentiment Analysis in Product Reviews on Social Media (Nafees et al., 2018)

This research looks at a variety of methods for categorising natural language text evaluations based on expressed opinions in order to identify if the extensive behaviour is negative, positive, or neutral.

2.4 Depression-Related Posts on Reddit Social Media Forums Detection (Dey et al., 2020)

This paper uses various text classification algorithms to investigate the performance of both single feature and combination feature sets to measure the indications of depression. Our findings suggest that good feature selection and diverse feature combinations lead to better prediction performance.

2.5 Extractive Summarization using Wikipedia for Weakly Supervised Sentiment Analysis (Mukherjee and Bhattacharyya, 2012)

We saw a supervised sentiment analysis approach for movie reviews in this paper. The purpose is to categorise a movie review into one of two polarity classes, positive or negative, based purely on the lines that indicate an opinion on the film, with all other irrelevant text removed. Wikipedia's world knowledge of movie-specific elements is incorporated into the methodology that is used to produce an extracted summary of the review, which includes the reviewer's judgments about the specific components of the film. This gets rid of any ideas that aren't relevant or objective in the context of the movie. The suggested system, WikiSent, does not require labelled data to train. It outperforms or is on par with existing semi-supervised and unsupervised methods in the domain on the same dataset. WikiSent is also utilised to conduct a general trend analysis of movie reviews.

2.6 Software/Tools Requirements

Softwares: Jupyter Notebook, Visual Studio Code

Natural Language Toolkit, Pandas, Numpy, Matplotlib. Tweepy (Twitter Application Programming Interface (API)), TextBlob, PRAW (Reddit API).

Web development Tools: Flask, Hyper Text Markup Language (HTML), Bootstrap, Cascading Style Sheets (CSS), Heroku.

Chapter 3

PROPOSED SYSTEM

3.1 System Analysis

3.1.1 System Requirement Analysis

- To access the tweets the requirement is to use tweepy twitter API, by creating a twitter developer account to get access to this API. Similarly in Reddit, the requirement is to use PRAW reddit API.
- To do sentiment analysis, the requirements are
  1. NLTK toolkit package (nltk.sentiment.vader)
  2. The function used to find polarity score is SentimentIntensityAnalyzer()
  3. TextBlob (perform Natural Language Processing (NLP) tasks)
- matplotlib.pyplot (For Plotting)
- To create a Webapp the requirements are
  1. Python (version >3.0)
  2. Flask 1.1.2 (Back End)
  3. HTML and CSS (Front End)
  4. Github and Heroku (For deployment)

3.1.2 Module Details of the System

1. Social Media Sentiment Analysis
2. Visualize the Results
3. Comparative Study of Analysis
4. Creating a Web Application

1) Social Media Sentiment Analysis: Extract a given number of tweets related to a keywords from Twitter using Twitter's API, Content from Reddit using Reddit's API and also Wikipedia article and perform Sentiment Analysis on it.

For retrieving user comments from reddit we have used PRAW API. By specifying the keyword and number of comments to extract, the API can retrieve top comments from a reddit post related to that keyword. Similarly, for retrieving tweets from twitter, we have used tweepy API. This retrieved data is cleaned (removing punctuation and special symbols). We have used Snowball Stemmer to trim a word to its root(stem) form.

Then we apply the sentiment intensity analyzer function from the NLTK toolkit to the data and find out the polarity and subjectivity scores. Polarity Scores range from -1 to 1. If it is near to +1, then the sentence is positive. If it is near to -1, then the sentence is negative. If it is not near to either of them, then is it is classified as a neutral sentence.

This sentiment analyzer function decides the polarity scores by averaging the intensity of each word. This intensity can be found in the textblob word corpus. Finally, this result is stored in a Dataframe.

2) Visualize the Results: Visualize the results of the Analysis using pie-chart.

3) Comparative Study of Analysis: Compare the analysis results extracted from both the platforms and present the results.

4) Creating a Web Application: Wrap the whole project with a Web Application.
In the Home Page of the Web Application, two options are given (Reddit or Twitter), out of which one can be chosen which redirects to the dashboard. In the dashboard, two details are asked (the keyword and the number of comments to be extracted).

When these details are entered the backend will run the python code and the sentiment analysis is done. This final dataframe is then displayed as a table in the result page along with a pie chart for visualization.

3.2 System Design

3.2.1 Architecture Diagram of Twitter Sentiment Analysis

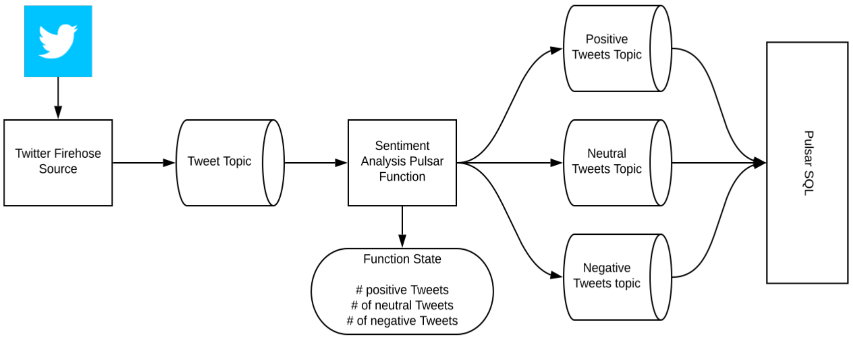


Figure 3.1: Architecture Diagram of Twitter Sentiment Analysis

3.2.2 Architecture Diagram of Reddit Sentiment Analysis

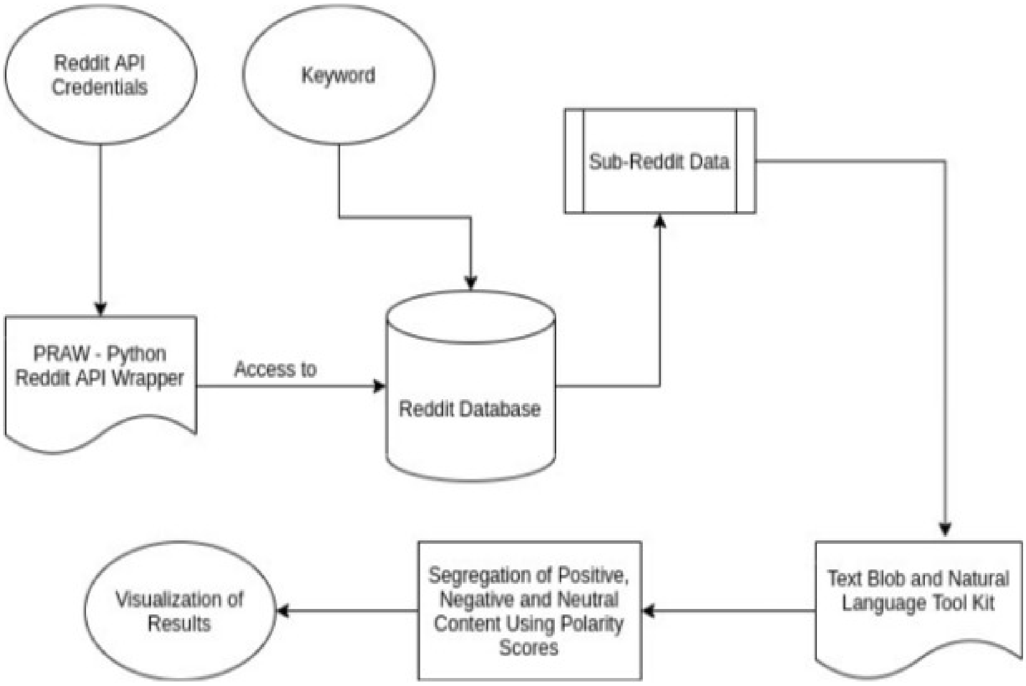

Reddit API Credentials
Keyword
Sub-Reddit Data
PRAW - Python Reddit API Wrapper
Access to
Reddit Database
Visualization of Results
Segregation of Positive, Negative and Neutral Content Using Polarity Scores
Text Blob and Natural Language Tool Kit

Figure 3.2: Architecture Diagram of Reddit Sentiment Analysis

3.2.3 Flow Diagram

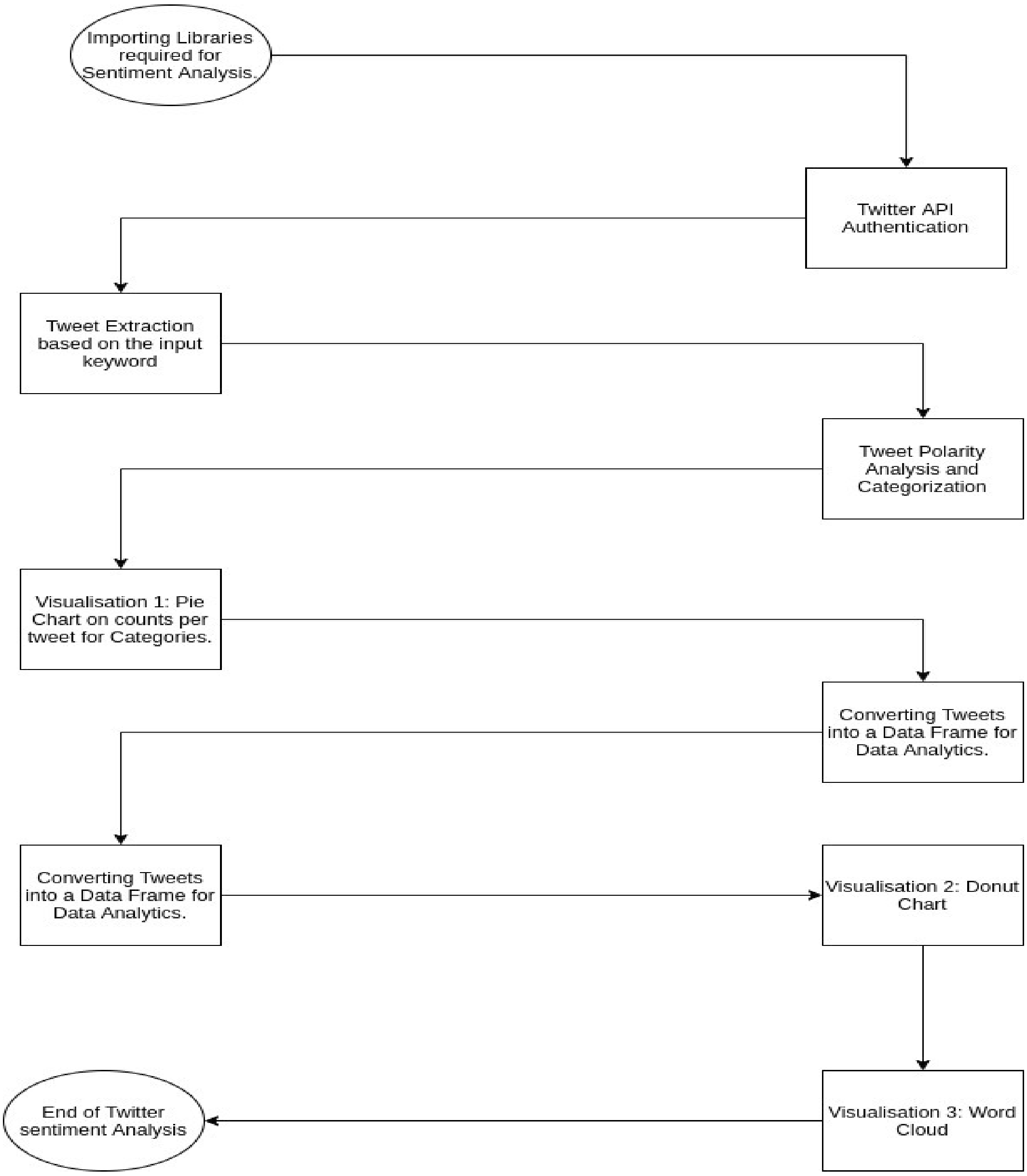

Importing Libraries required for Sentiment Analysis.
Twitter API Authentication
Tweet Extraction based on the input keyword
Tweet Polarity Analysis and Categorization
Visualisation 1: Pie Chart on counts per tweet for Categories.
Converting Tweets into a Data Frame for Data Analytics.
Converting Tweets into a Data Frame for Data Analytics.
Visualisation 2: Donut Chart
Visualisation 3: Word Cloud
End of Twitter sentiment Analysis

Figure 3.3: Flow Diagram

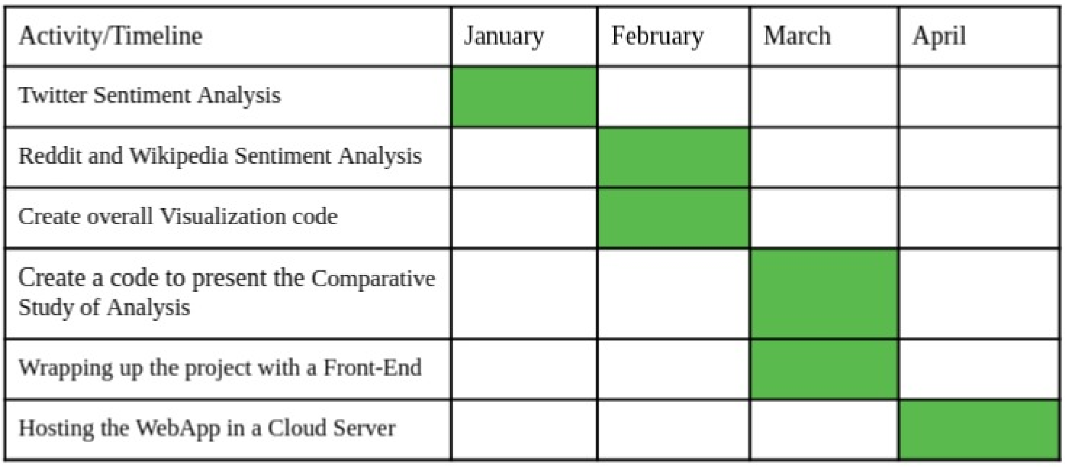
Chapter 4

PROJECT PLAN

4.1 Responsibilities of Team Members

1. ALLA BADRINATH REDDY [CB.EN.U4CSE17305]
    - Literature Survey and Documentation in Phase 1
    - Research and Ideation in Phase 2
    - User Interface Design in Phase 3
2. RAGUL S V [CB.EN.U4CSE17348]
    - Visualization and Analysis in Phase 1
    - Reddit Sentiment Analysis Algorithm in Phase 2
    - Routing and Integration with Backend in Phase 3
3. RAJASEKAR M [CB.EN.U4CSE17349]
    - Content Extraction and Analysis in Phase 1
    - Wikipedia Sentiment Analysis Algorithm in Phase 2
    - Hosting and Documentation in Phase 3

4.2 Project time line



Activity/Timeline	January	February	March	April
Twitter Sentiment Analysis	🟩			
Reddit and Wikipedia Sentiment Analysis		🟩		
Create overall Visualization code		🟩		
Create a code to present the Comparative Study of Analysis			🟩	
Wrapping up the project with a Front-End			🟩	
Hosting the WebApp in a Cloud Server				🟩

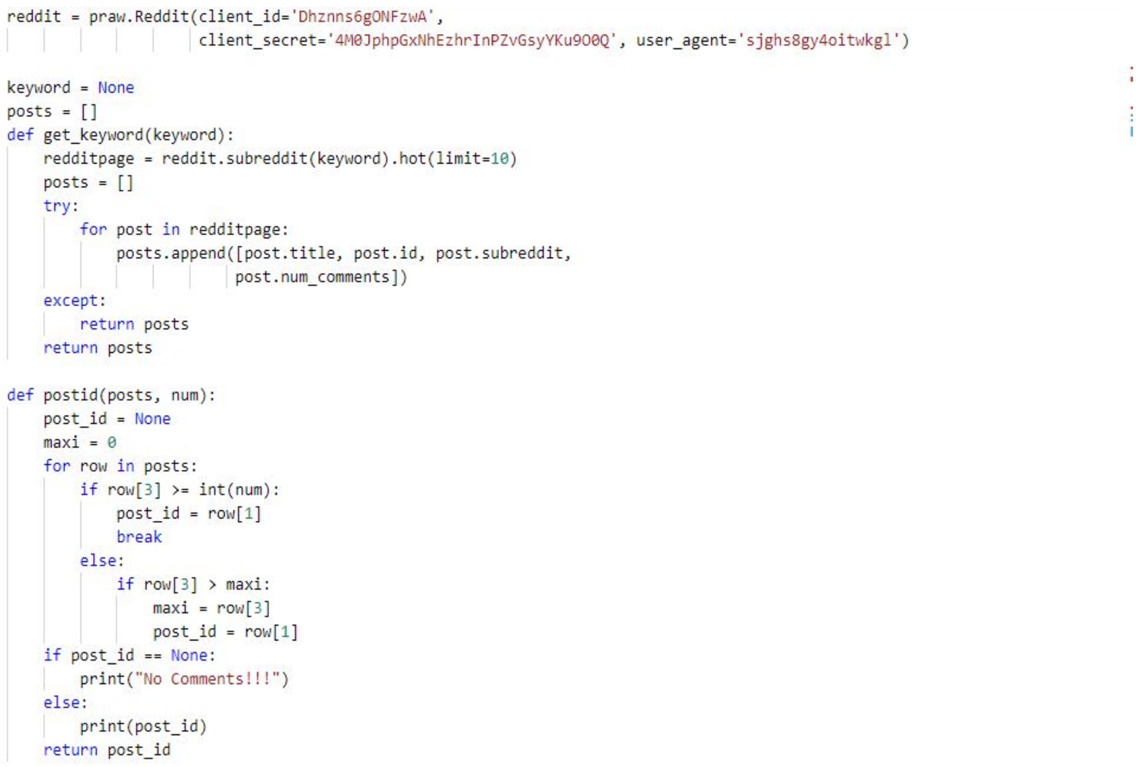
Figure 4.1: Gantt Chart

Chapter 5

IMPLEMENTATION

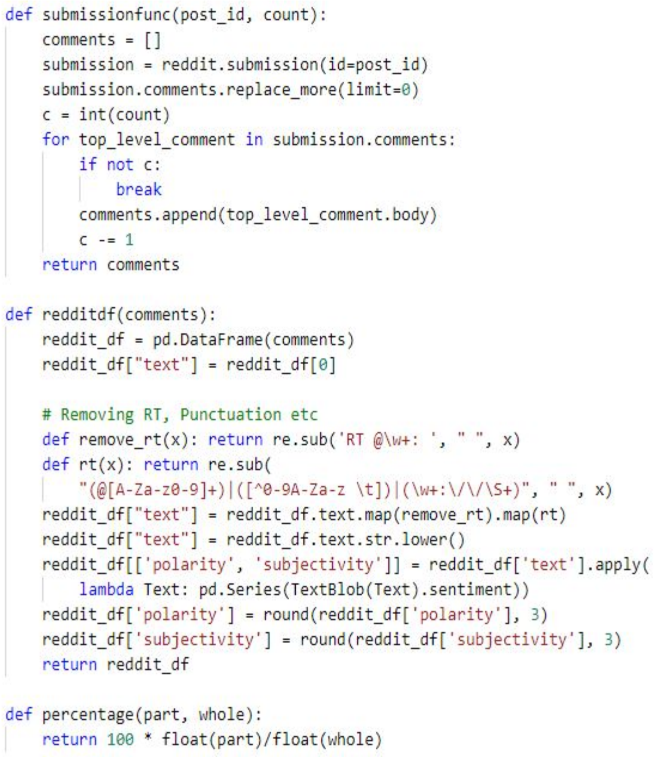
5.1 Tools Used

1. Twitter API to fetch the given number of tweets related to the given keyword as a POST Request.
2. Reddit API to fetch the given number of commands related to the given keyword as a POST Request.
3. NLTK or Natural Language Toolkit as our NLP Engine to identify the polarity of the content.
4. Flask, a python based front-end framework to integrate our code with the front-end web design.
5. Bootstrap as our Cascaded Style Sheet framework that helped us with the web design.
6. GitHub as our version control system to help us collaborate as a team for the project.
7. Heroku as our Cloud Server Hosting platform that helped us publish our project as a WebApp in the domain name https://group29sentimentanalyser.herokuapp.com.



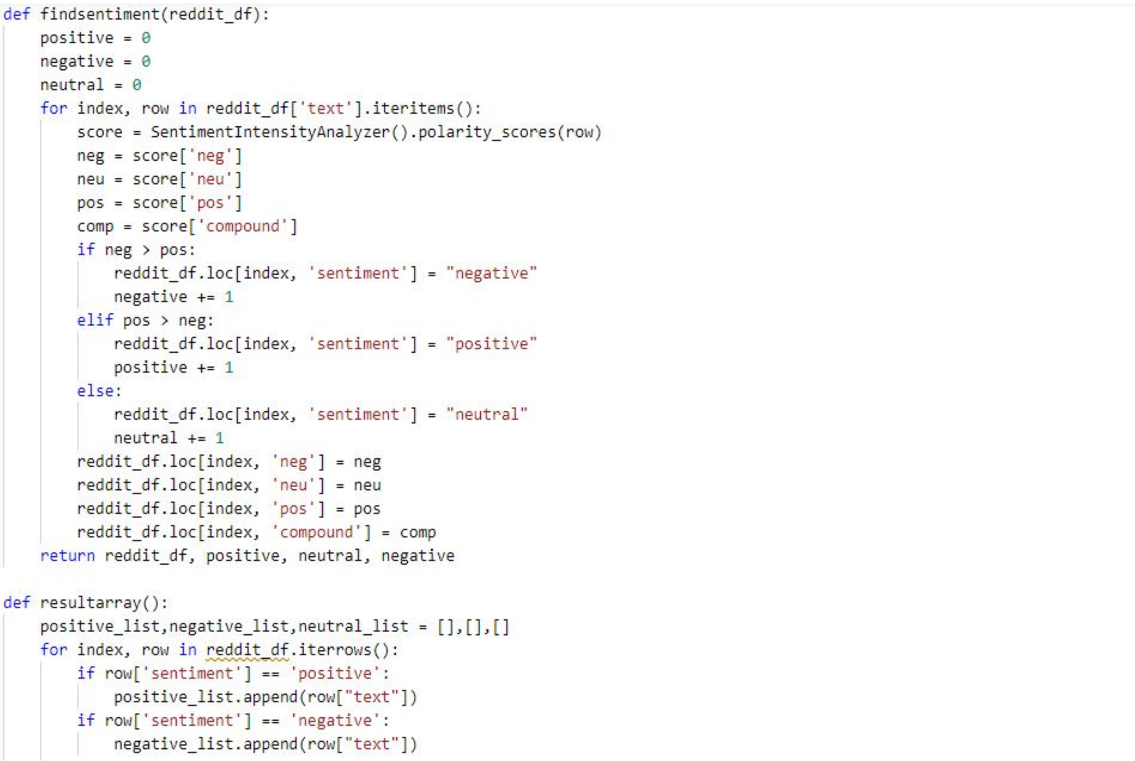
```
reddit = praw.Reddit(client_id='Dhznns6gONFzwA',
                     client_secret='4M0JphpGxNhEzhrInPZvGsyYKu9O0Q', user_agent='sjghs8gy4oitwkgl')

keyword = None
posts = []
def get_keyword(keyword):
    redditpage = reddit.subreddit(keyword).hot(limit=10)
    posts = []
    try:
        for post in redditpage:
            posts.append([post.title, post.id, post.subreddit,
                          post.num_comments])
    except:
        return posts
    return posts

def postid(posts, num):
    post_id = None
    maxi = 0
    for row in posts:
        if row[3] >= int(num):
            post_id = row[1]
            break
        else:
            if row[3] > maxi:
                maxi = row[3]
                post_id = row[1]
    if post_id == None:
        print("No Comments!!!")
    else:
        print(post_id)
    return post_id
```

Figure 5.1: Reddit Retrieving Data using PRAW API



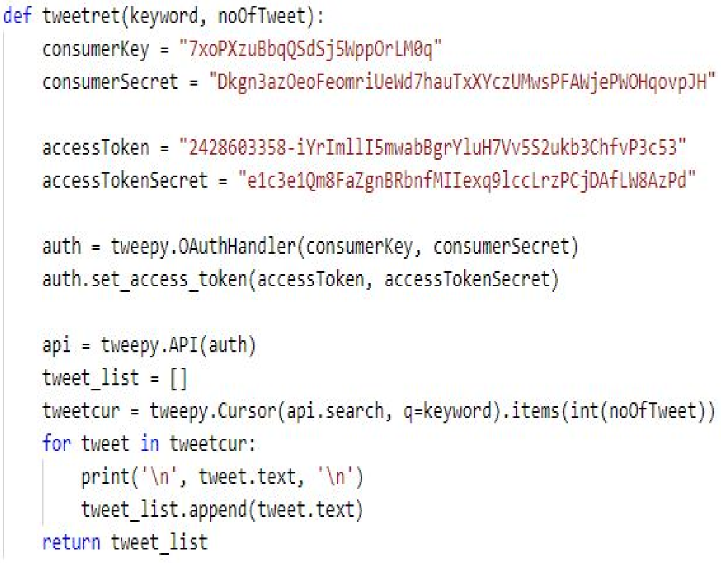
```
def submissionfunc(post_id, count):
    comments = []
    submission = reddit.submission(id=post_id)
    submission.comments.replace_more(limit=0)
    c = int(count)
    for top_level_comment in submission.comments:
        if not c:
            break
        comments.append(top_level_comment.body)
        c -= 1
    return comments

def redditdf(comments):
    reddit_df = pd.DataFrame(comments)
    reddit_df["text"] = reddit_df[0]

    # Removing RT, Punctuation etc
    def remove_rt(x): return re.sub('RT @\w+: ', " ", x)
    def rt(x): return re.sub(
        "(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", x)
    reddit_df["text"] = reddit_df.text.map(remove_rt).map(rt)
    reddit_df["text"] = reddit_df.text.str.lower()
    reddit_df[['polarity', 'subjectivity']] = reddit_df['text'].apply(
        lambda Text: pd.Series(TextBlob(Text).sentiment))
    reddit_df['polarity'] = round(reddit_df['polarity'], 3)
    reddit_df['subjectivity'] = round(reddit_df['subjectivity'], 3)
    return reddit_df

def percentage(part, whole):
    return 100 * float(part)/float(whole)
```

Figure 5.2: Reddit Data Formatting



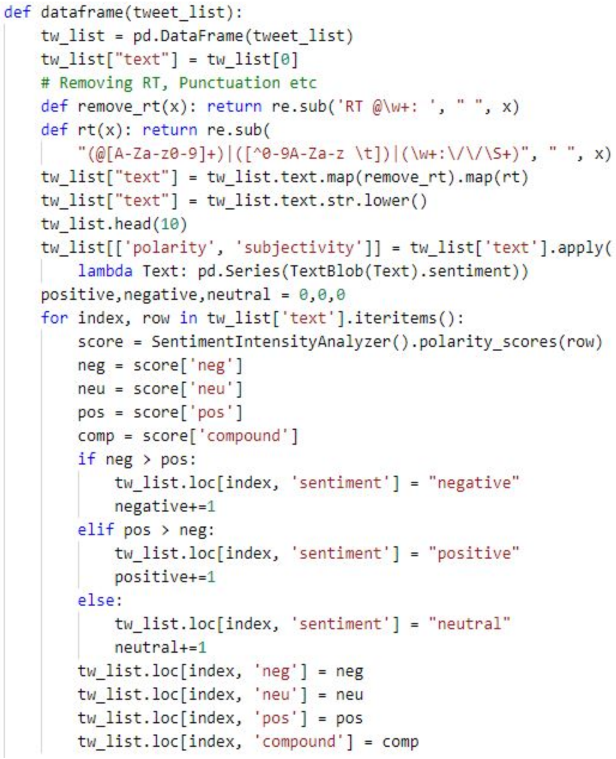
```
def findsentiment(reddit_df):
    positive = 0
    negative = 0
    neutral = 0
    for index, row in reddit_df['text'].iteritems():
        score = SentimentIntensityAnalyzer().polarity_scores(row)
        neg = score['neg']
        neu = score['neu']
        pos = score['pos']
        comp = score['compound']
        if neg > pos:
            reddit_df.loc[index, 'sentiment'] = "negative"
            negative += 1
        elif pos > neg:
            reddit_df.loc[index, 'sentiment'] = "positive"
            positive += 1
        else:
            reddit_df.loc[index, 'sentiment'] = "neutral"
            neutral += 1
        reddit_df.loc[index, 'neg'] = neg
        reddit_df.loc[index, 'neu'] = neu
        reddit_df.loc[index, 'pos'] = pos
        reddit_df.loc[index, 'compound'] = comp
    return reddit_df, positive, neutral, negative

def resultarray():
    positive_list,negative_list,neutral_list = [],[],[]
    for index, row in reddit_df.iterrows():
        if row['sentiment'] == 'positive':
            positive_list.append(row["text"])
        if row['sentiment'] == 'negative':
            negative_list.append(row["text"])
```

Figure 5.3: Sentiment Analysing Function



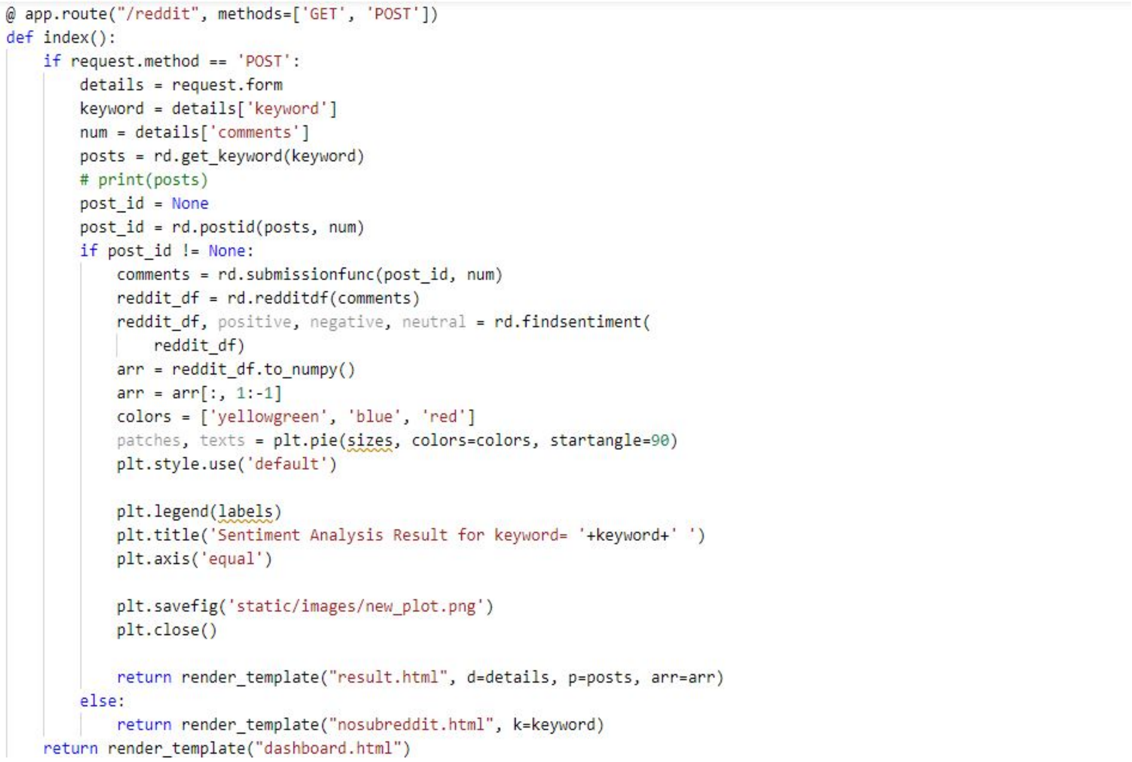
```
def tweetret(keyword, noOfTweet):
    consumerKey = "7xoPXzuBbqQSdSj5WppOrLM0q"
    consumerSecret = "Dkgn3azOeoFeomriUeWd7hauTxXYczUMwsPFAWjePWOHqovpJH"

    accessToken = "2428603358-iYrImllI5mwabBgrYluH7Vv5S2ukb3ChfvP3c53"
    accessTokenSecret = "e1c3e1Qm8FaZgnBRbnfMIIexq9lccLrzPCjDAfLW8AzPd"

    auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
    auth.set_access_token(accessToken, accessTokenSecret)

    api = tweepy.API(auth)
    tweet_list = []
    tweetcur = tweepy.Cursor(api.search, q=keyword).items(int(noOfTweet))
    for tweet in tweetcur:
        print('\n', tweet.text, '\n')
        tweet_list.append(tweet.text)
    return tweet_list
```

Figure 5.4: Twitter Retrieving Data using Tweepy API



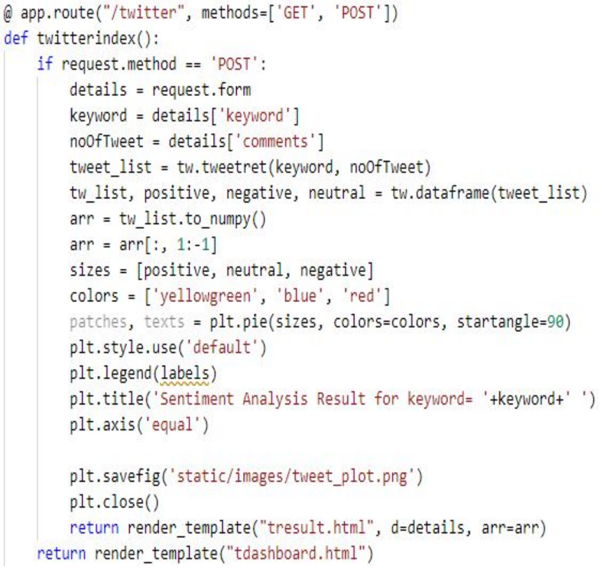
```python
def dataframe(tweet_list):
    tw_list = pd.DataFrame(tweet_list)
    tw_list["text"] = tw_list[0]
    # Removing RT, Punctuation etc
    def remove_rt(x): return re.sub('RT @\w+: ', " ", x)
    def rt(x): return re.sub(
        "(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", x)
    tw_list["text"] = tw_list.text.map(remove_rt).map(rt)
    tw_list["text"] = tw_list.text.str.lower()
    tw_list.head(10)
    tw_list[['polarity', 'subjectivity']] = tw_list['text'].apply(
        lambda Text: pd.Series(TextBlob(Text).sentiment))
    positive,negative,neutral = 0,0,0
    for index, row in tw_list['text'].iteritems():
        score = SentimentIntensityAnalyzer().polarity_scores(row)
        neg = score['neg']
        neu = score['neu']
        pos = score['pos']
        comp = score['compound']
        if neg > pos:
            tw_list.loc[index, 'sentiment'] = "negative"
            negative+=1
        elif pos > neg:
            tw_list.loc[index, 'sentiment'] = "positive"
            positive+=1
        else:
            tw_list.loc[index, 'sentiment'] = "neutral"
            neutral+=1
        tw_list.loc[index, 'neg'] = neg
        tw_list.loc[index, 'neu'] = neu
        tw_list.loc[index, 'pos'] = pos
        tw_list.loc[index, 'compound'] = comp
```

Figure 5.5: Tweet Curating Function



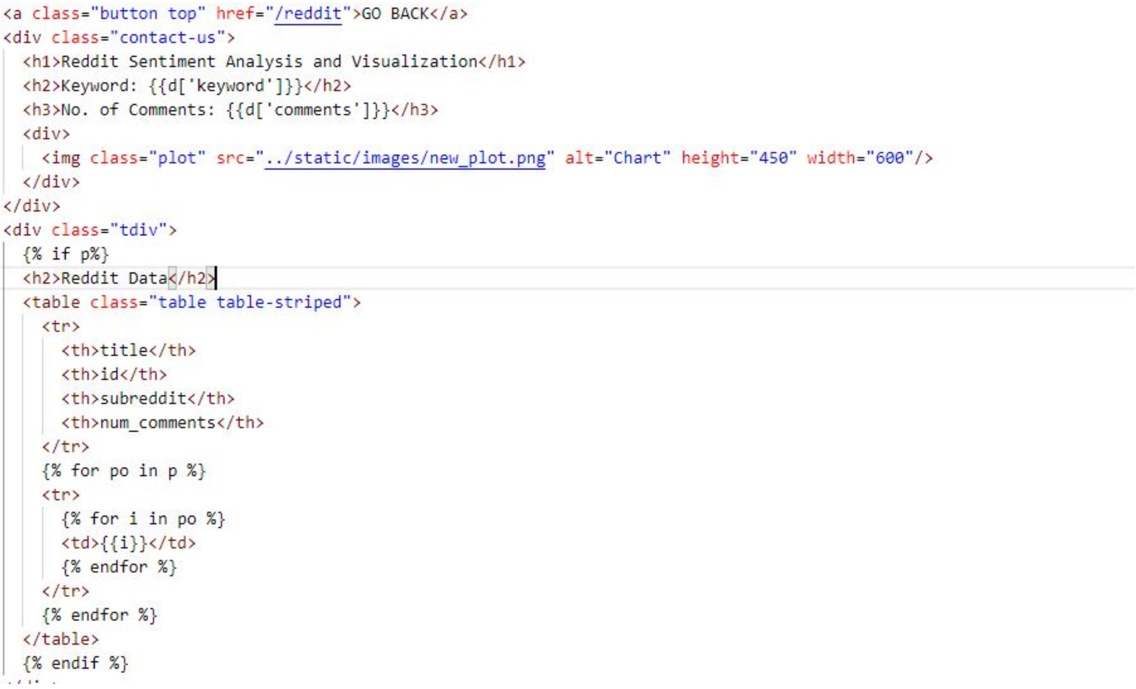
```python
@ app.route("/reddit", methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        details = request.form
        keyword = details['keyword']
        num = details['comments']
        posts = rd.get_keyword(keyword)
        # print(posts)
        post_id = None
        post_id = rd.postid(posts, num)
        if post_id != None:
            comments = rd.submissionfunc(post_id, num)
            reddit_df = rd.redditdf(comments)
            reddit_df, positive, negative, neutral = rd.findsentiment(
                reddit_df)
            arr = reddit_df.to_numpy()
            arr = arr[:, 1:-1]
            colors = ['yellowgreen', 'blue', 'red']
            patches, texts = plt.pie(sizes, colors=colors, startangle=90)
            plt.style.use('default')

            plt.legend(labels)
            plt.title('Sentiment Analysis Result for keyword= '+keyword+' ')
            plt.axis('equal')

            plt.savefig('static/images/new_plot.png')
            plt.close()

            return render_template("result.html", d=details, p=posts, arr=arr)
        else:
            return render_template("nosubreddit.html", k=keyword)
    return render_template("dashboard.html")
```

Figure 5.6: Index Routing Function - Flask (Reddit)



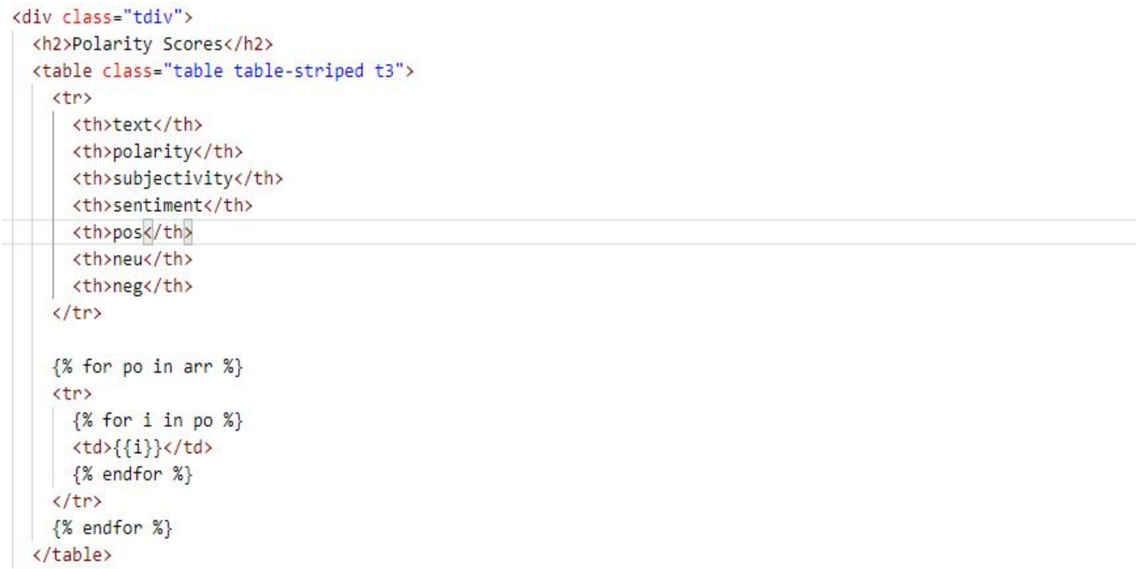
```python
@ app.route("/twitter", methods=['GET', 'POST'])
def twitterindex():
    if request.method == 'POST':
        details = request.form
        keyword = details['keyword']
        noOfTweet = details['comments']
        tweet_list = tw.tweetret(keyword, noOfTweet)
        tw_list, positive, negative, neutral = tw.dataframe(tweet_list)
        arr = tw_list.to_numpy()
        arr = arr[:, 1:-1]
        sizes = [positive, neutral, negative]
        colors = ['yellowgreen', 'blue', 'red']
        patches, texts = plt.pie(sizes, colors=colors, startangle=90)
        plt.style.use('default')
        plt.legend(labels)
        plt.title('Sentiment Analysis Result for keyword= '+keyword+' ')
        plt.axis('equal')

        plt.savefig('static/images/tweet_plot.png')
        plt.close()
        return render_template("tresult.html", d=details, arr=arr)
    return render_template("tdashboard.html")
```

Figure 5.7: Index Routing Function - Flask (Twitter)



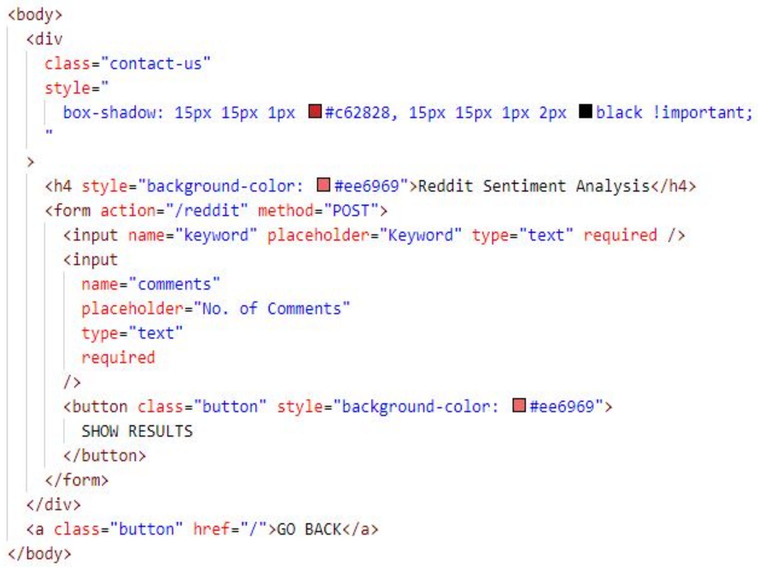
```html
<a class="button top" href="/reddit">GO BACK</a>
<div class="contact-us">
  <h1>Reddit Sentiment Analysis and Visualization</h1>
  <h2>Keyword: {{d['keyword']}}</h2>
  <h3>No. of Comments: {{d['comments']}}</h3>
  <div>
    <img class="plot" src="../static/images/new_plot.png" alt="Chart" height="450" width="600"/>
  </div>
</div>
<div class="tdiv">
  {% if p%}
  <h2>Reddit Data</h2>
  <table class="table table-striped">
    <tr>
      <th>title</th>
      <th>id</th>
      <th>subreddit</th>
      <th>num_comments</th>
    </tr>
    {% for po in p %}
    <tr>
      {% for i in po %}
      <td>{{i}}</td>
      {% endfor %}
    </tr>
    {% endfor %}
  </table>
  {% endif %}
```

Figure 5.8: Reddit Sentiment Result HTML file



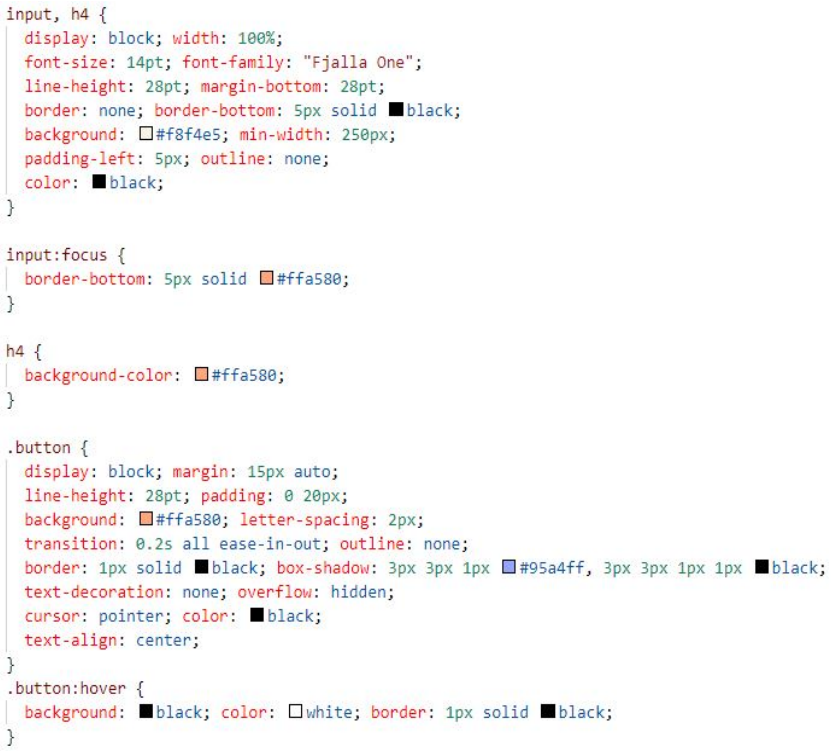
```
<div class="tdiv">
  <h2>Polarity Scores</h2>
  <table class="table table-striped t3">
    <tr>
      <th>text</th>
      <th>polarity</th>
      <th>subjectivity</th>
      <th>sentiment</th>
      <th>pos</th>
      <th>neu</th>
      <th>neg</th>
    </tr>

    {% for po in arr %}
    <tr>
      {% for i in po %}
      <td>{{i}}</td>
      {% endfor %}
    </tr>
    {% endfor %}
  </table>
```

Figure 5.9: Twitter Sentiment Result HTML file



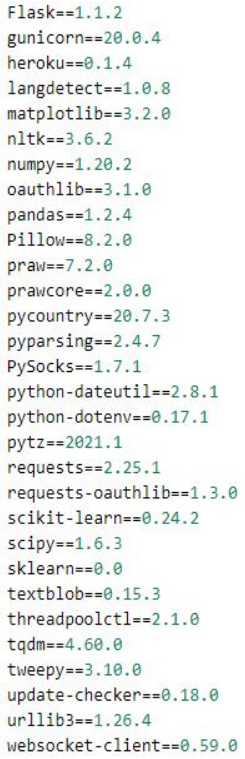
```
<body>
  <div
    class="contact-us"
    style="
      box-shadow: 15px 15px 1px ■#c62828, 15px 15px 1px 2px ■black !important;
    "
  >
    <h4 style="background-color: ■#ee6969">Reddit Sentiment Analysis</h4>
    <form action="/reddit" method="POST">
      <input name="keyword" placeholder="Keyword" type="text" required />
      <input
        name="comments"
        placeholder="No. of Comments"
        type="text"
        required
      />
      <button class="button" style="background-color: ■#ee6969">
        SHOW RESULTS
      </button>
    </form>
  </div>
  <a class="button" href="/">GO BACK</a>
</body>
```

Figure 5.10: Reddit Dashboard



```
input, h4 {
  display: block; width: 100%;
  font-size: 14pt; font-family: "Fjalla One";
  line-height: 28pt; margin-bottom: 28pt;
  border: none; border-bottom: 5px solid ■black;
  background: □#f8f4e5; min-width: 250px;
  padding-left: 5px; outline: none;
  color: ■black;
}

input:focus {
  border-bottom: 5px solid ■#ffa580;
}

h4 {
  background-color: ■#ffa580;
}

.button {
  display: block; margin: 15px auto;
  line-height: 28pt; padding: 0 20px;
  background: ■#ffa580; letter-spacing: 2px;
  transition: 0.2s all ease-in-out; outline: none;
  border: 1px solid ■black; box-shadow: 3px 3px 1px ■#95a4ff, 3px 3px 1px 1px ■black;
  text-decoration: none; overflow: hidden;
  cursor: pointer; color: ■black;
  text-align: center;
}
.button:hover {
  background: ■black; color: □white; border: 1px solid ■black;
}
```

Figure 5.11: CSS file



```
Flask==1.1.2
gunicorn==20.0.4
heroku==0.1.4
langdetect==1.0.8
matplotlib==3.2.0
nltk==3.6.2
numpy==1.20.2
oauthlib==3.1.0
pandas==1.2.4
Pillow==8.2.0
praw==7.2.0
prawcore==2.0.0
pycountry==20.7.3
pyparsing==2.4.7
PySocks==1.7.1
python-dateutil==2.8.1
python-dotenv==0.17.1
pytz==2021.1
requests==2.25.1
requests-oauthlib==1.3.0
scikit-learn==0.24.2
scipy==1.6.3
sklearn==0.0
textblob==0.15.3
threadpoolctl==2.1.0
tqdm==4.60.0
tweepy==3.10.0
update-checker==0.18.0
urllib3==1.26.4
websocket-client==0.59.0
```

Figure 5.12: Libraries Requirements File

Chapter 6

RESULTS AND DISCUSSIONS

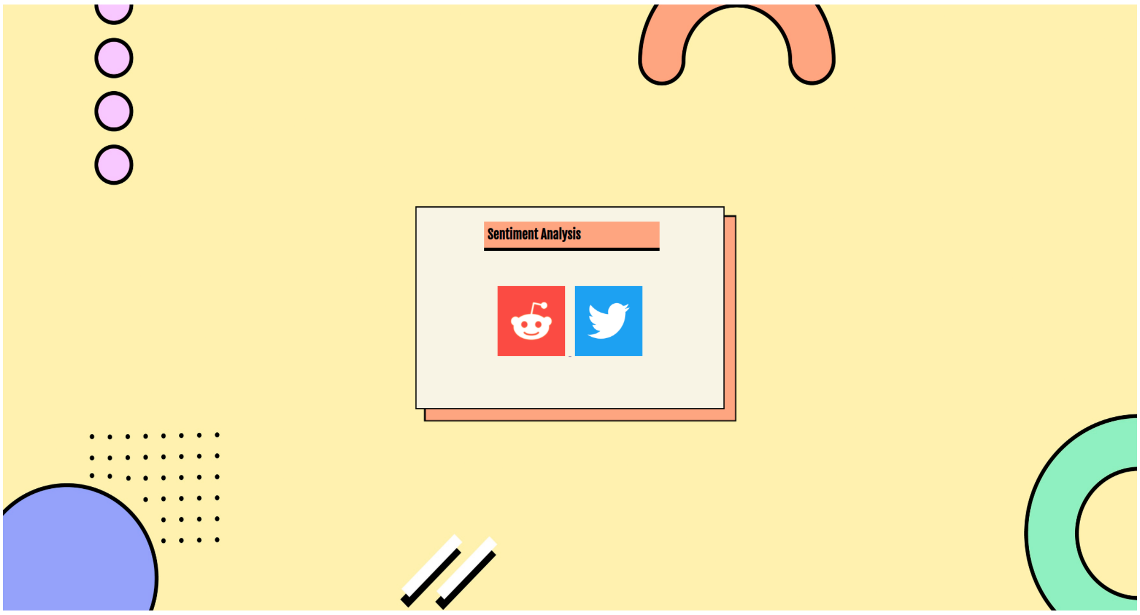
6.1 Our Intention

Our main objective with the project was that, we wanted to create a user-friendly web application that does sentiment analysis and publishes the result in a well-curated manner and hence we focused more on User Experience.

6.2 Our Outcome

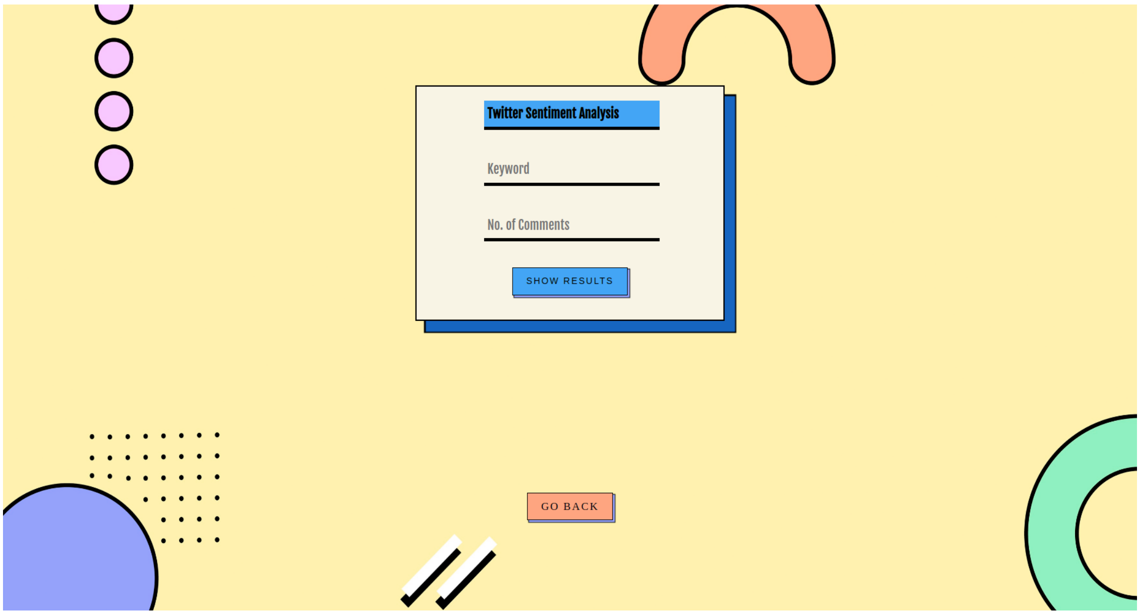
As a Result, we had created the same and had hosted it in a cloud server for anyone around the world to access under the domain name group29sentimentanalyser.herokuapp.com

6.3 Demo Screenshots



Sentiment Analysis

Figure 6.1: Landing Page of our Project

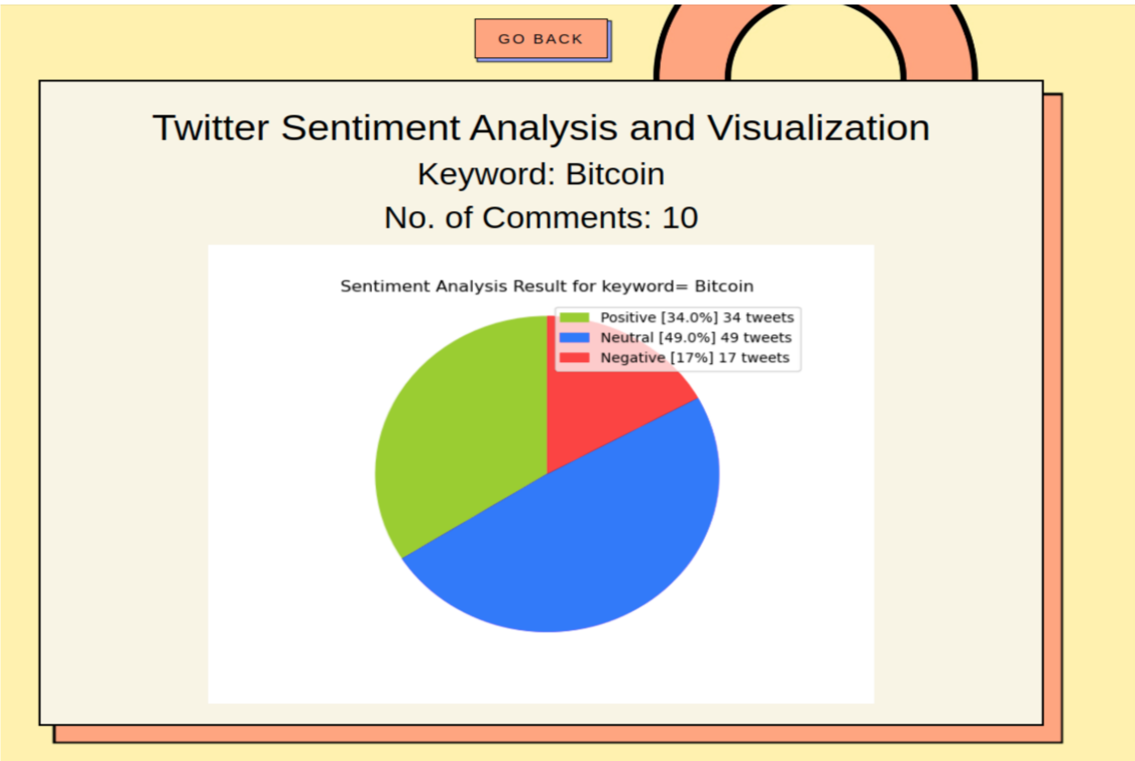


Twitter Sentiment Analysis

Keyword

No. of Comments

Figure 6.2: Input Page of Twitter Sentiment Analysis



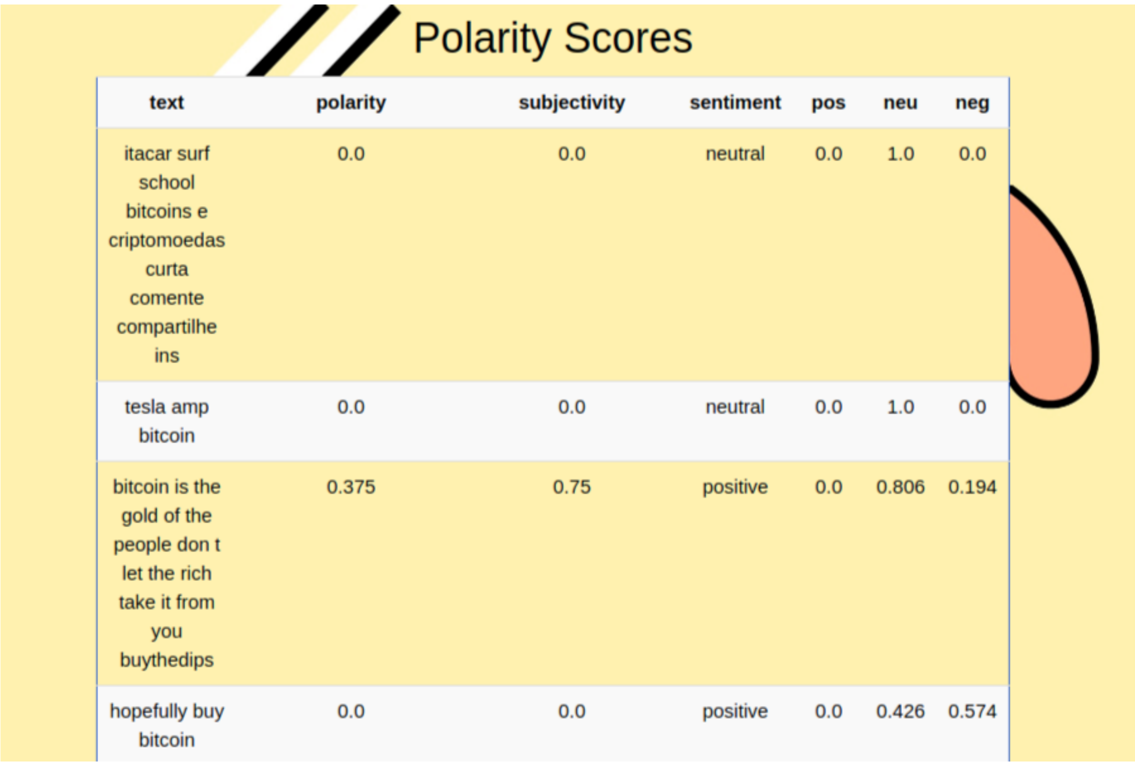
Twitter Sentiment Analysis and Visualization

Keyword: Bitcoin

No. of Comments: 10

Sentiment Analysis Result for keyword= Bitcoin
🟩 Positive [34.0%] 34 tweets
🟦 Neutral [49.0%] 49 tweets
🟥 Negative [17%] 17 tweets

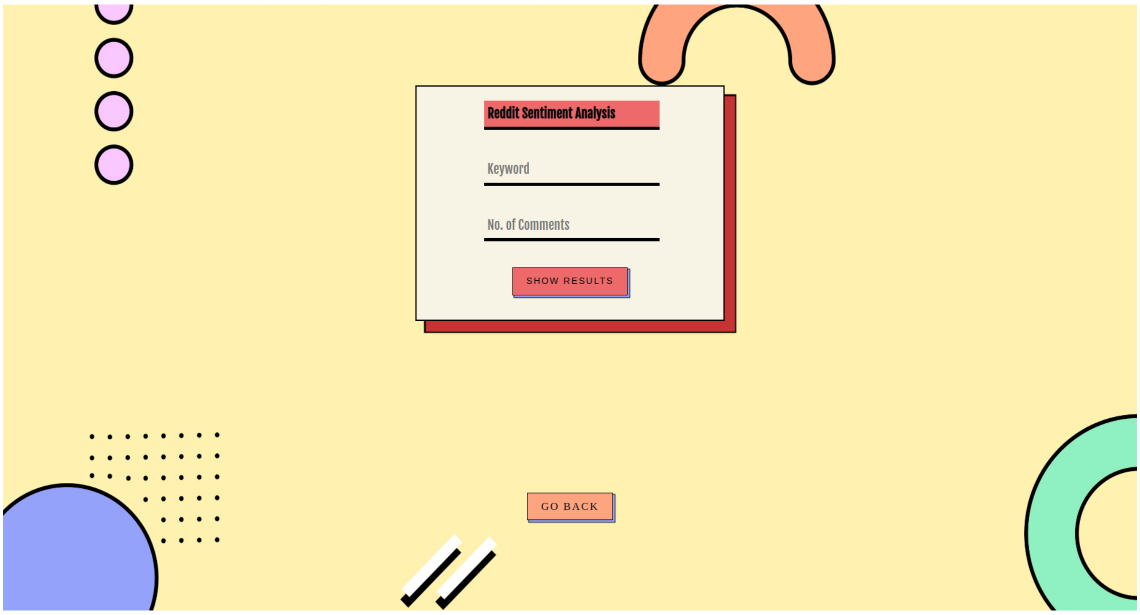
Figure 6.3: Result card of the Analysis



Polarity Scores

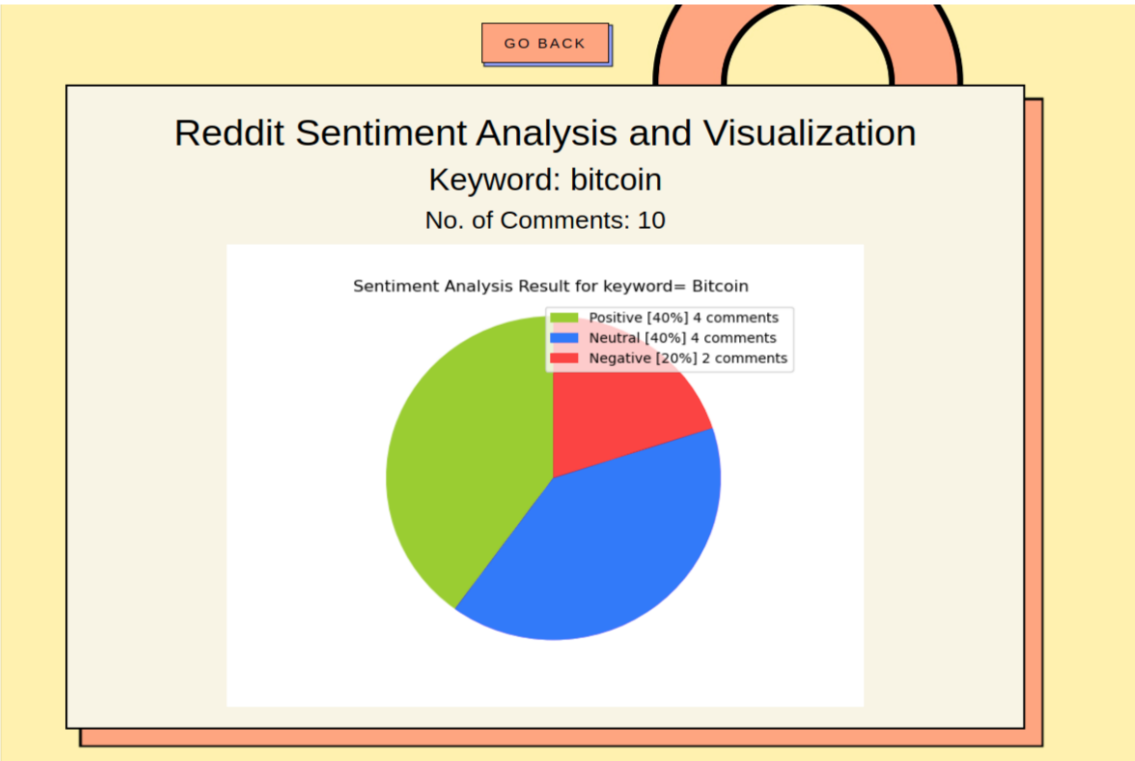
text	polarity	subjectivity	sentiment	pos	neu	neg
itacar surf school bitcoins e criptomoedas curta comente compartilhe ins	0.0	0.0	neutral	0.0	1.0	0.0
tesla amp bitcoin	0.0	0.0	neutral	0.0	1.0	0.0
bitcoin is the gold of the people don t let the rich take it from you buythedips	0.375	0.75	positive	0.0	0.806	0.194
hopefully buy bitcoin	0.0	0.0	positive	0.0	0.426	0.574

Figure 6.4: Curated list of tweets and their sentiment



Reddit Sentiment Analysis

Figure 6.5: Input page of Reddit Sentiment Analysis



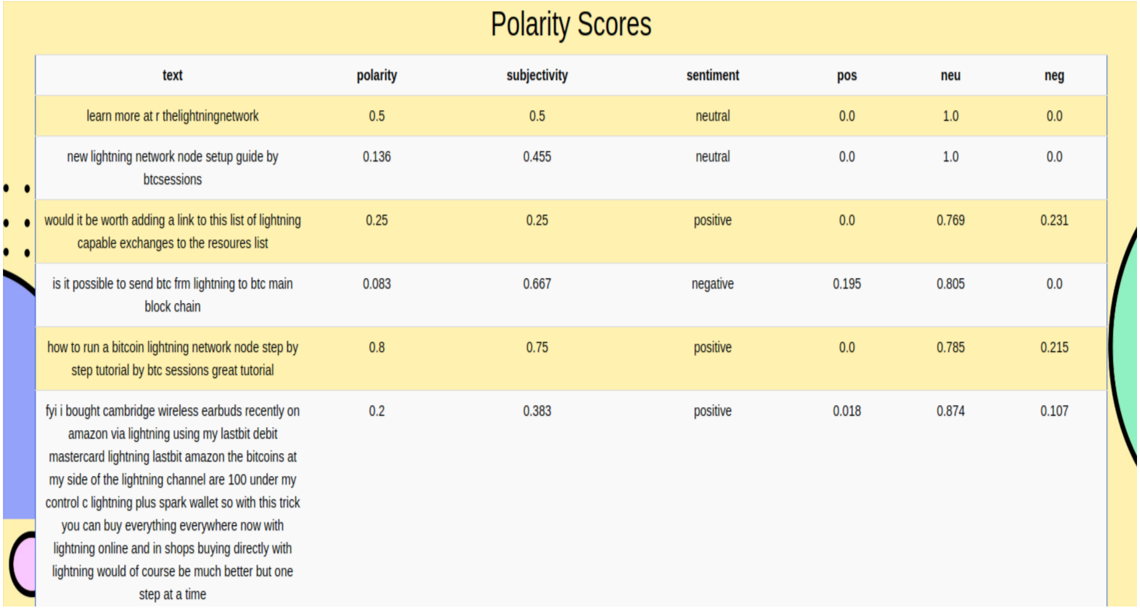
Reddit Sentiment Analysis and Visualization

Keyword: bitcoin

No. of Comments: 10

Sentiment Analysis Result for keyword= Bitcoin
🟩 Positive [40%] 4 comments
🟦 Neutral [40%] 4 comments
🟥 Negative [20%] 2 comments

Figure 6.6: Result card of the Analysis



Polarity Scores

text	polarity	subjectivity	sentiment	pos	neu	neg
learn more at r thelightningnetwork	0.5	0.5	neutral	0.0	1.0	0.0
new lightning network node setup guide by btcsessions	0.136	0.455	neutral	0.0	1.0	0.0
would it be worth adding a link to this list of lightning capable exchanges to the resoures list	0.25	0.25	positive	0.0	0.769	0.231
is it possible to send btc frm lightning to btc main block chain	0.083	0.667	negative	0.195	0.805	0.0
how to run a bitcoin lightning network node step by step tutorial by btc sessions great tutorial	0.8	0.75	positive	0.0	0.785	0.215
fyi i bought cambridge wireless earbuds recently on amazon via lightning using my lastbit debit mastercard lightning lastbit amazon the bitcoins at my side of the lightning channel are 100 under my control c lightning plus spark wallet so with this trick you can buy everything everywhere now with lightning online and in shops buying directly with lightning would of course be much better but one step at a time	0.2	0.383	positive	0.018	0.874	0.107

Figure 6.7: Curated list of Reddit Comments and their sentiment

Chapter 7

CONCLUSION

As the world becomes more and more democratic and hyper-connected, there is room for insurmountable amount of public opinions that is to be generated on potentially every debatable topic and with Natural Language Processing as a Technology, we can start understanding the voices of people more deeply to derive more meaningful solutions for them. This project of ours is a step towards democratising the technology of Sentiment Analysis for the non-experts in the same. It focused on bringing out an Ubiquitous solution that which is easy and quick to use. In the end, we were able to achieve what we set out to accomplish.

Chapter 8

FUTURE ENHANCEMENTS

Some of the Future Enhancements for the project would be:

1. Adding other social media like YouTube and Facebook for a wider perspective.
2. Adding more graphical representations of Result for better understanding.
3. A feature to import and export data in different formats for reference.
4. A feature to make a query based tweet extraction to narrow down the search funnel.
5. Improving the Data Cleaning algorithm for cleaner prediction and sharper result.

Since our project is uploaded in GitHub, it is open source. Every Open Source Developer across the globe is invited to contribute to the project.

REFERENCES

1. Dey, S., Sarkar, I., Chakraborty, S., and Roy, S. (2020). "Depression detection using intelligent algorithms from social media context - state of the art, trends and future roadmap. 45–62.

2. Mukherjee, S. and Bhattacharyya, P. (2012). "Wikisent: Weakly supervised sentiment analysis through extractive summarization with wikipedia." *Machine Learning and Knowledge Discovery in Databases*, P. A. Flach, T. De Bie, and N. Cristianini, eds., Berlin, Heidelberg. Springer Berlin Heidelberg, 774–793.

3. Nafees, M., Dar, H., Lali, I. U., and Tiwana, S. (2018). "Sentiment analysis of polarity in product reviews in social media." *2018 14th International Conference on Emerging Technologies (ICET)*. 1–6.

4. Neri, F., Aliprandi, C., Capeci, F., Cuadros, M., and By, T. (2012). "Sentiment analysis on social media.

5. Ogudo, K. A. and Nestor, D. M. J. (2019). "Sentiment analysis application and natural language processing for mobile network operators' support on social media." *2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*. 1–10.