SENTIMENT ANALYSIS

Batch 3

121710302025 - K. Sarika
121710302030 - K. Phani Chandana
121710302032 - M. Vishruthi
121710302059 - Y. Shreya
121710302062 - P. Meenakshi
121710302066 - Ch. Akhila

TABLE OF CONTENTS

INTRODUCTION

Sentiment analysis is the computational study of people's opinions, appraisals, and emotions toward entities, events and their attributes. In the past few years, it attracted a great deal of attentions from both academia and industry due to many challenging research problems and a wide range of applications. However, finding opinion sites and monitoring them on the Web can still be a formidable task because there are a large number of diverse sites, and each site may also have a huge volume of opinionated text.

The system we proposed covers analysis of any topic by parsing the tweets fetched from Twitter and the Text provided using Python. This computation process determines whether the information is positive, negative or neutral.

OBJECTIVE

Considering the sentiment analysis, we chose to work on the text data and twitter hashtags. Moreover, sentiments are defined based on semantic relations and the frequency of each word in an input sentence that allows getting a more precise output as a result. Text data it is used to predict the sentiment behind the text which is used in opinion analysis. For this analysis we approached the Textblob library, which is necessary task in NLP, depending on the obtained polarity ranges, the sentiment is reported. Twitter analysis is used to analyse people's pulse by giving a hashtag of a subject. For the analysis, Tweepy is the official python library for twitter API that enables Python to communicate with Twitter platform and get live tweets data. We have built a simple model, and trained the data on it. We also made predictions using the model.

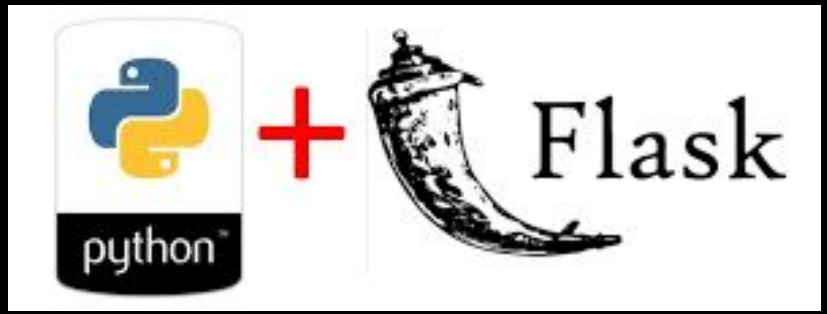
SOFTWARES USED

FRONTEND

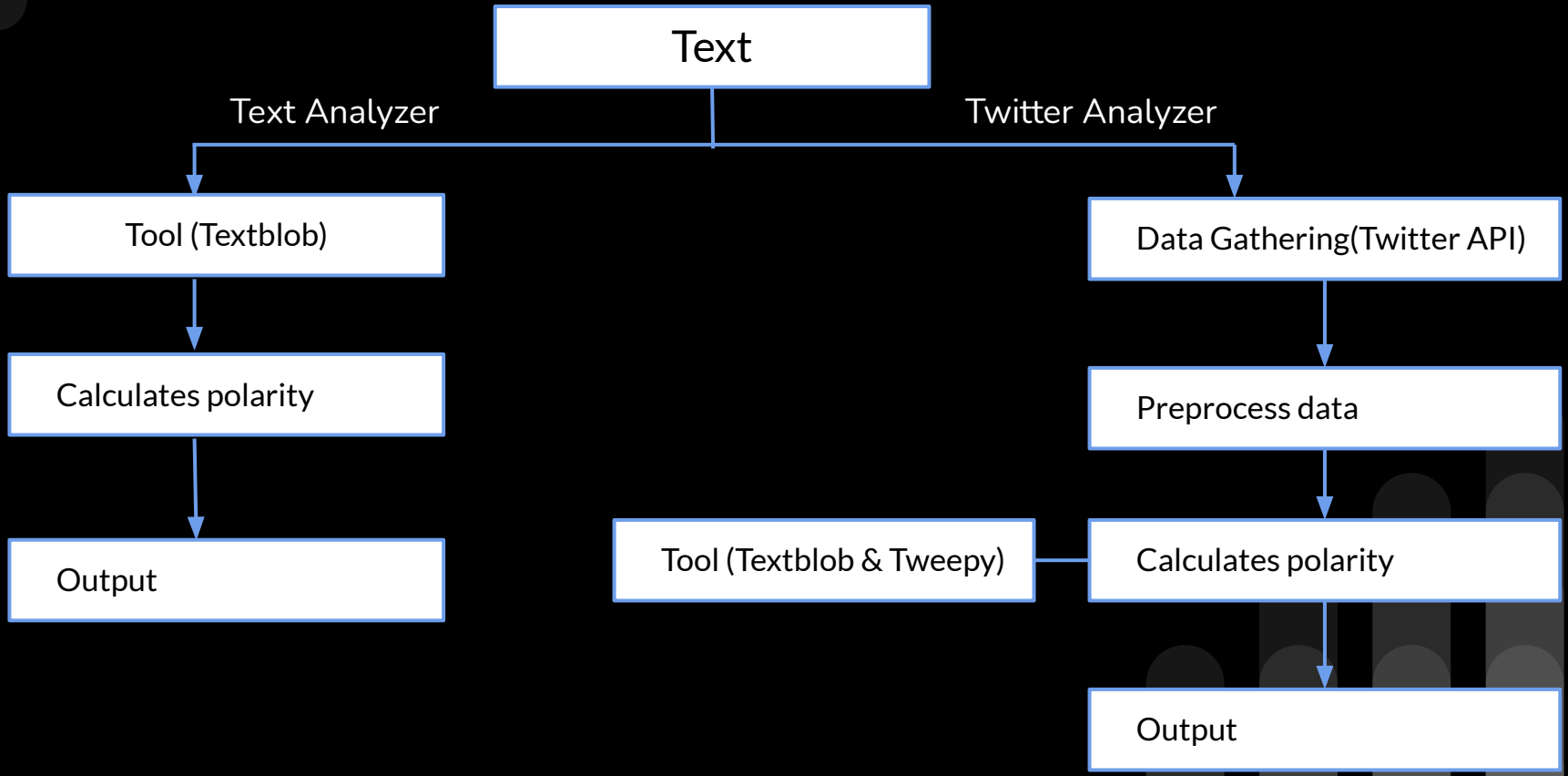
- HTML
- CSS
- JAVASCRIPT

BACKEND

- PYTHON
- FLASK - Python framework



FLOWCHART


Text
Text Analyzer
Twitter Analyzer
Tool (Textblob)
Calculates polarity
Output
Data Gathering(Twitter API)
Preprocess data
Tool (Textblob & Tweepy)
Calculates polarity
Output

METHODOLOGY

FLASK CODE:

```
@app.route("/")
def home():
    return render_template("sample1.html")
```

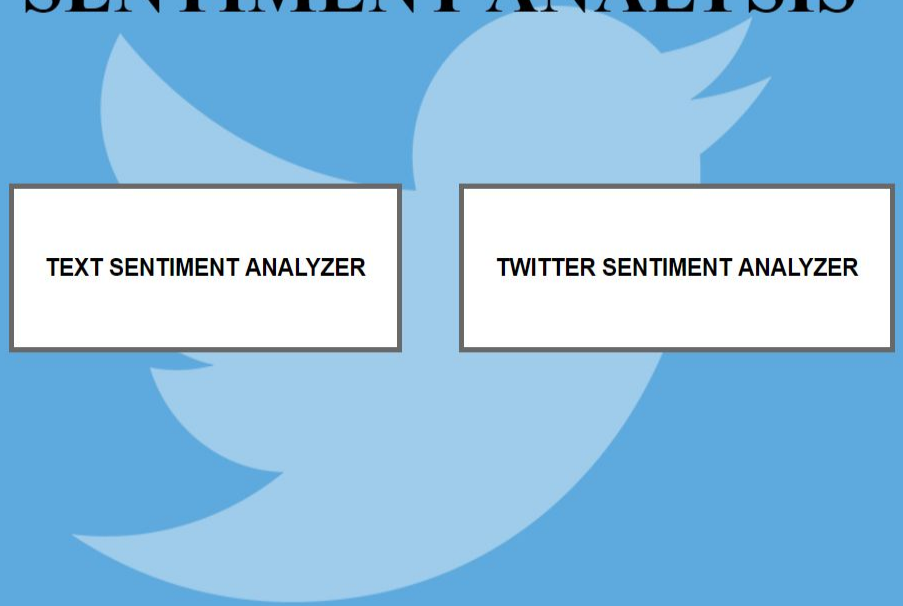
App routing is used to map the specific URL with the associated function that is intended to perform some task. It is used to access some particular page like Flask Tutorial in the web application.

METHODOLOGY

HTML CODE:

```
<body>
  {% block content %}
  <div class = "center">
    <h1 style="color:black; fontsize:60px;">SENTIMENT ANALYSIS<h1>
  </div>
  <div class = "center1">
    <a href="/login"><button class="btn button1" style="margin-right: 22px;margin-left: 20px">
                    <b>TEXT SENTIMENT ANALYZER</b></button></a>
    <a href="/login1"><button class="btn button1" style="margin-right: 20px;margin-left: 20px">
                    <b>TWITTER SENTIMENT ANALYZER</b></button></a>
  </div>
  {% endblock %}
</body>
```

SENTIMENT ANALYSIS



TEXT SENTIMENT ANALYZER

TWITTER SENTIMENT ANALYZER

METHODOLOGY

FLASK APP CODE:

```
@app.route("/login", methods=["POST", "GET"])
def login():
    if request.method == "POST":
        session.permanent = True
        user = request.form["nm"]
        session["user"] = user
        return redirect(url_for("user"))
    else:
        if "user" in session:
            return redirect(url_for("user"))
        return render_template("login.html")
```

METHODOLOGY

HTML CODE:

```
{% extends "sample1.html" %}
{% block title %}Text Analysis{% endblock %}
{% block content %}
<h1 align="center" style="color:black;font-size:60px;">TEXT SENTIMENT ANALYZER<h1>
<br><br>
<form align = "center" action="#" method="post">
  <textarea rows="20" cols ="80" name="nm" placeholder="Write your text here!!!"></textarea>
  <br><br>
  <input type="submit" value="PREDICT" class="btn button1" style = "font-size:20px;font-weight:bold;
    position:absolute;left:33%;top:77.5%;">
</form>
<a href="/back"><button class="btn button1" style = "padding :17.5px 17.5px;font-weight:bold;
    position:absolute;left:60%;top:77.5%"><b>BACK</b></button></a>
{% endblock content %}
```

TEXT SENTIMENT ANALYZER

Write your text here!!!

PREDICT

BACK

METHODOLOGY

FLASK APP CODE:

```
@app.route("/user")
def user():
  if "user" in session:
    user = session["user"]
    y = user
    from textblob import TextBlob
    edu = TextBlob(y)
    x=edu.sentiment.polarity
    z = x*100
    z = str(z)
    if x<0:
      b = "Negative"
      return render_template('login1.html', value1=z, value2=b,value3=y)
    elif x == 0:
      d = "Neutral"
      return render_template('login1.html', value1=z, value2=d,value3=y)
    elif x>0 and x<=1:
      f = "Positive"
      return render_template('login1.html', value1=z, value2=f,value3=y)
  else:
    return redirect(url_for("login"))
```

METHODOLOGY

HTML CODE:

```
<body>
  {% block content %}
  <h1 align="center" style="color:black;font-size:60px;">PREDICTED RESULT<h1>
  <div class ="form1" style="width:700px;height:100px;">Your text is {{value1}} % {{value2}}</div>
  <div class = "ha3" style="width:400px;height:500px;">
  {% if value2 == "Positive" %}
    <img src="{{url_for('static', filename='h.png')}}"/>
  {% elif value2 == "Neutral" %}
    <img src="{{url_for('static', filename='neu.png')}}"/>
  {% elif value2 == "Negative" %}
    <img src="{{url_for('static', filename='n.png')}}"/>
  {% endif %}
  </div>
  <a href="/back" class = "form2"><button class="btn button1" style = "padding : 17px 17px"><b>BACK</b></button></a>
  {% endblock content %}
</body>
```

PREDICTED RESULT

Your text is 70.0 % Positive

BACK

METHODOLOGY

FLASK APP CODE:

```
@app.route("/login1", methods=["POST", "GET"])
def login1():
    if request.method == "POST":
        session.permanent = True
        user1 = request.form["op"]
        session["user"] = user1
        return redirect(url_for("user1"))
    else:
        if "user" in session:
            return redirect(url_for("user1"))
        return render_template("login2.html")
```

METHODOLOGY

HTML CODE:

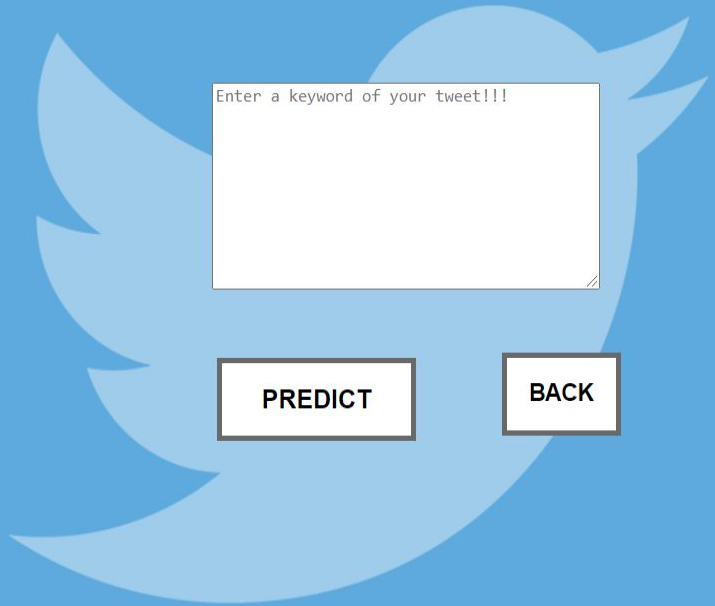
```
<body>
  {% block content %}
  <h1 align="center" style="color:black;font-size:60px;">TWITTER SENTIMENT ANALYZER<h1>
  <form align = "center" action="#" method="post">
    <br><br>
    <textarea rows="10" cols ="40" name="op" placeholder="Enter a keyword of your tweet!!!">
    </textarea>
    <br><br>
    <input type="submit" value="PREDICT" class="btn button1" style = "font-size:20px;
      font-weight:bold;position:absolute;left:40%;top:57%">
  </form>
  <a href="/back"><button class="btn button1" style = "padding :17.5px 17.5px;
    font-weight:bold;position:absolute;left:55%;top:57%"><b>BACK</b></button></a>
  {% endblock content %}
</body>
```

Template Inheritance — Flask Do

Twitter Sentimental Analysis

127.0.0.1:5000/login1

TWITTER SENTIMENT ANALYZER



METHODOLOGY

FLASK APP CODE:

```
@app.route("/user1")
def user1():
   if "user" in session:
      user = session["user"]
      import tweepy
      from textblob import TextBlob
      consumer_key = ‘your twitter consumer key'
      consumer_key_secret = ‘your twitter consumer secret key'
      access_token = ‘your twitter access token'
      access_token_secret = ‘your twitter access token scret key'
      auth = tweepy.OAuthHandler(consumer_key, consumer_key_secret)
      auth.set_access_token(access_token, access_token_secret)
      api = tweepy.API(auth)
      public_tweets = api.search(user)
      p=0
      n=0
```

METHODOLOGY

Continue...

```
for tweet in public_tweets:
        edu = TextBlob(tweet.text)
        x=edu.sentiment.polarity
        if x>0:
          p +=1
        else:
          n +=1
    a = ((p)/(p+n))*100
    b = 100-a
    return render_template('login3.html', value1=a, value2=b,value3=user)
else:
    return redirect(url_for("login"))
```

METHODOLOGY

HTML CODE:

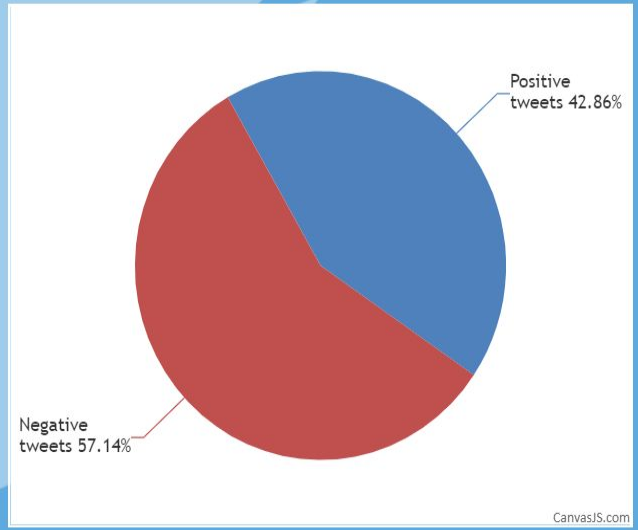
```
<div class ="ha3" id="chartContainer"></div>
<script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
<h3 align="center" style="color:black;font-size:60px;">PREDICTED RESULT<h3>
<div class ="form1" style="width:500px;height:400px;border:1px solid #000;">{{value3}}</div>
<a class = "form2" href="/back"><button class="btn button1" style = "padding : 18px 18px"><b>BACK</b></button></a>
<script>
window.onload = function()
{  var chart = new CanvasJS.Chart("chartContainer", {
   animationEnabled: true,
   data:
   [{
      type: "pie",
      startAngle: 240,
      yValueFormatString: "##0.00\"%\"",
      indexLabel: "{label} {y}",
      dataPoints:
      [
         {y: {{value1}}, label: "Positive tweets"},
         {y: {{value2}}, label: "Negative tweets"}
      ] }]  });
   chart.render(); }
```

Template Inheritance — Flask Do | Twitter Analysis

127.0.0.1:5000/user1#

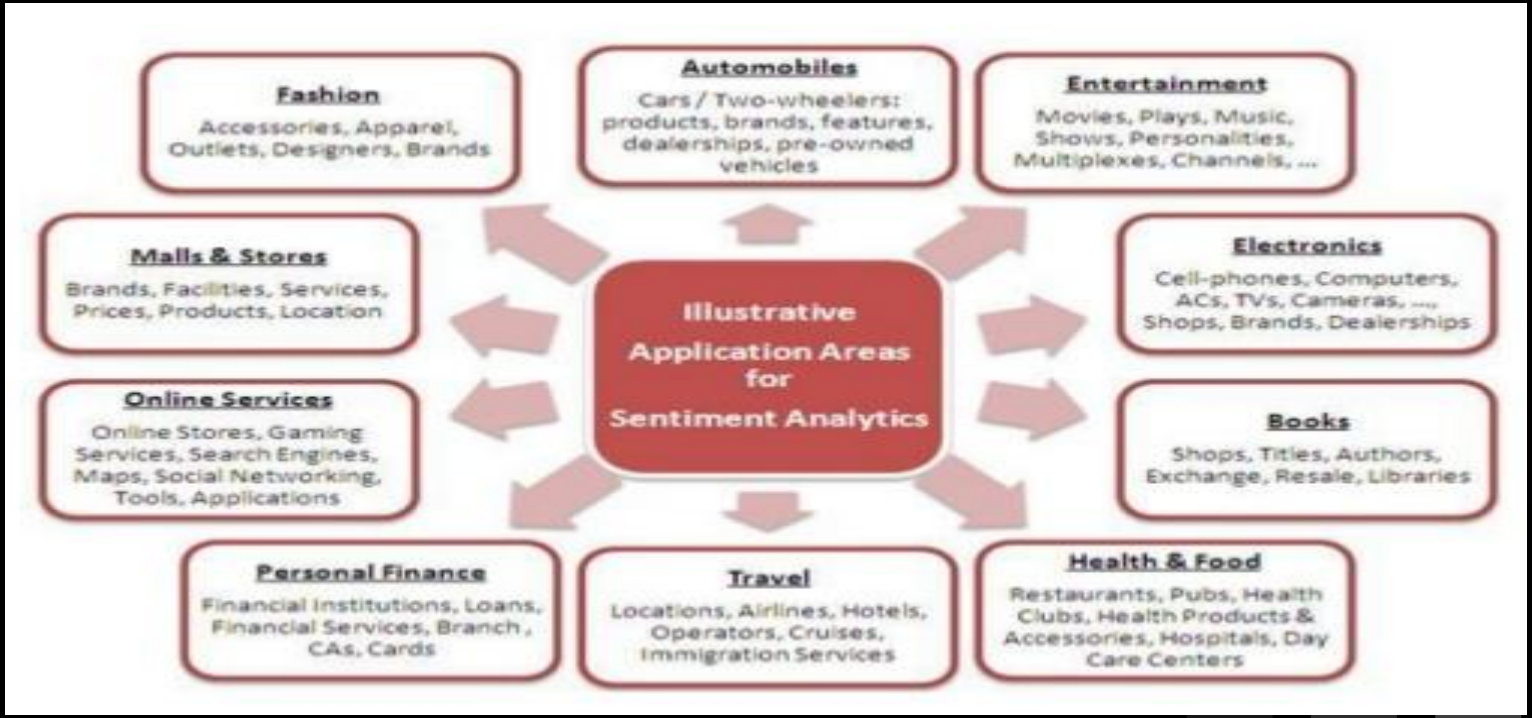
PREDICTED RESULT

trump



BACK

APPLICATIONS



CONCLUSION

We presented results for sentiment analysis on Twitter and Text data using python. Applying sentimental analysis to extract the sentiment became an important work for many organizations and even individuals. The model created presents approach for sentiment analysis to uncover the sentiment, we extracted the opinion words by combining the adjectives in the text and the hashtags from twitter. This model can be improved further with more semantic knowledge and using fields like machine learning and NLP.

FUTURE SCOPE

- Data Pre-Processing using more parameters to get best sentiments
- Updating Dictionary for new Synonym and Antonyms of already existing words
- Web-Application can be converted to Mobile Application
- Context Sentimental Analysis may be implemented in future for accuracy purposes.

THANK YOU ..