**Introduction:**

**What is Sentiment Analysis?**

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. However, analysis of social media streams is usually restricted to just basic sentiment analysis and count based metrics. This is akin to just scratching the surface and missing out on those high value insights that are waiting to be discovered.

**Relevance:**

Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people’s sentiments towards certain entities. From a user’s perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. From a researcher’s perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers. However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. he second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral.

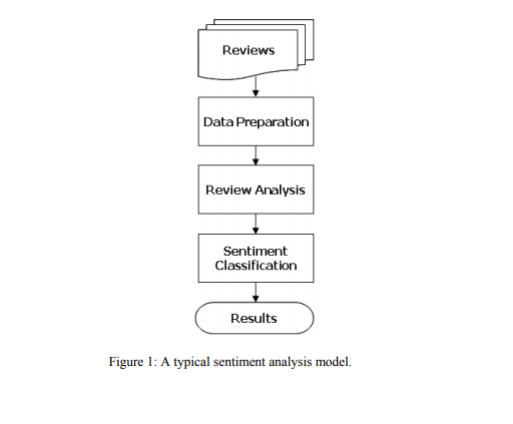
**Problem Definition:**

Sentiment Analysis also known as Opinion Mining refers to the use of natural language processing, text analysis to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

In this project, we aim to perform Sentiment Analysis of user reviews. Data used in this project are gathering from users (user defined data). We expect to do review if the statement is positive, negative or neutral. Also, this web application can display a percentage of positivity, negativity and neutral score as an output.

**Objective:**

* Taking and gathering all the user inputs(reviews) from the web portal.
* Analyze and Categories user inputs(reviews) using python nltk library.
* Analyze sentiments on dataset from document level (review level).
* Define and separate every word basis on the sentiment score.
* Categorization or classification of opinion sentiment into-
* Positive
* Negative
* Neutral



**Basic Concepts and Tools:**

**Introduction to Python:**

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.

Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

**Characteristics of Python:**

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation. We have listed below a few essential features.

**1) Easy to Learn and Use**

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

**2) Expressive Language**

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type **print("Hello World")**. It will take only one line to execute, while Java or C takes multiple lines.

**3) Interpreted Language**

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

**4) Cross-platform Language**

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

**5) Free and Open Source**

Python is freely available for everyone. It is freely available on its official website www.python.org

. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

**6) Object-Oriented Language**

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

**Python Flask:**

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects.

**WSGI:**

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

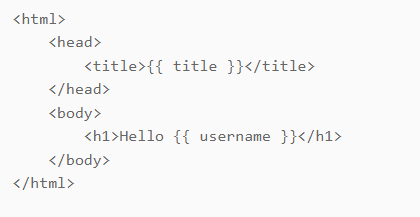
**Werkzeug:**

Werkzeug is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeg as one of its bases.

**jinja2:**

jinja2 is a popular template engine for Python.A web template system combines a template with a specific data source to render a dynamic web page.

This allows you to pass Python variables into HTML templates like this:



**Natural Language Toolkit (NLTK):**

The Natural Language Toolkit (NLTK) seems to be a Python programming environment for creating applications for statistical natural language processing (NLP).

For tokenization, parsing, classification, stemming, labeling, as well as semantic reasoning, it includes language processing libraries. Additionally, it comes with a curriculum and even a book to describes the usually presented various language processing jobs that NLTK offers, together with visual demos including experimental data repositories.

A collection of libraries as well as applications for statistics language comprehension can be found in the NLTK (Natural Language Toolkit) Library. One of the most potent NLP libraries, it includes tools that allow computers to comprehend natural language as well as respond appropriately whenever it is used.

**Python Regex:**

A regular expression is a set of characters with highly specialized syntax that we can use to find or match other characters or groups of characters. In short, regular expressions, or Regex, are widely used in the UNIX world.

The re-module in Python gives full support for regular expressions of Pearl style. The re module raises the re.error exception whenever an error occurs while implementing or using a regular expression.

**Python VaderSentiment:**

VADER (Valence Aware Dictionary and sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

**User Interface Model:**

**Introduction to HTML:**

HTML stands for Hypertext Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

**Features Of HTML:**

1) It is a very **easy and simple language**. It can be easily understood and modified.

2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.

3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add a **link** on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.

5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.

**Introduction to CSS:**

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colors, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

While html uses tags, css uses rulesets. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

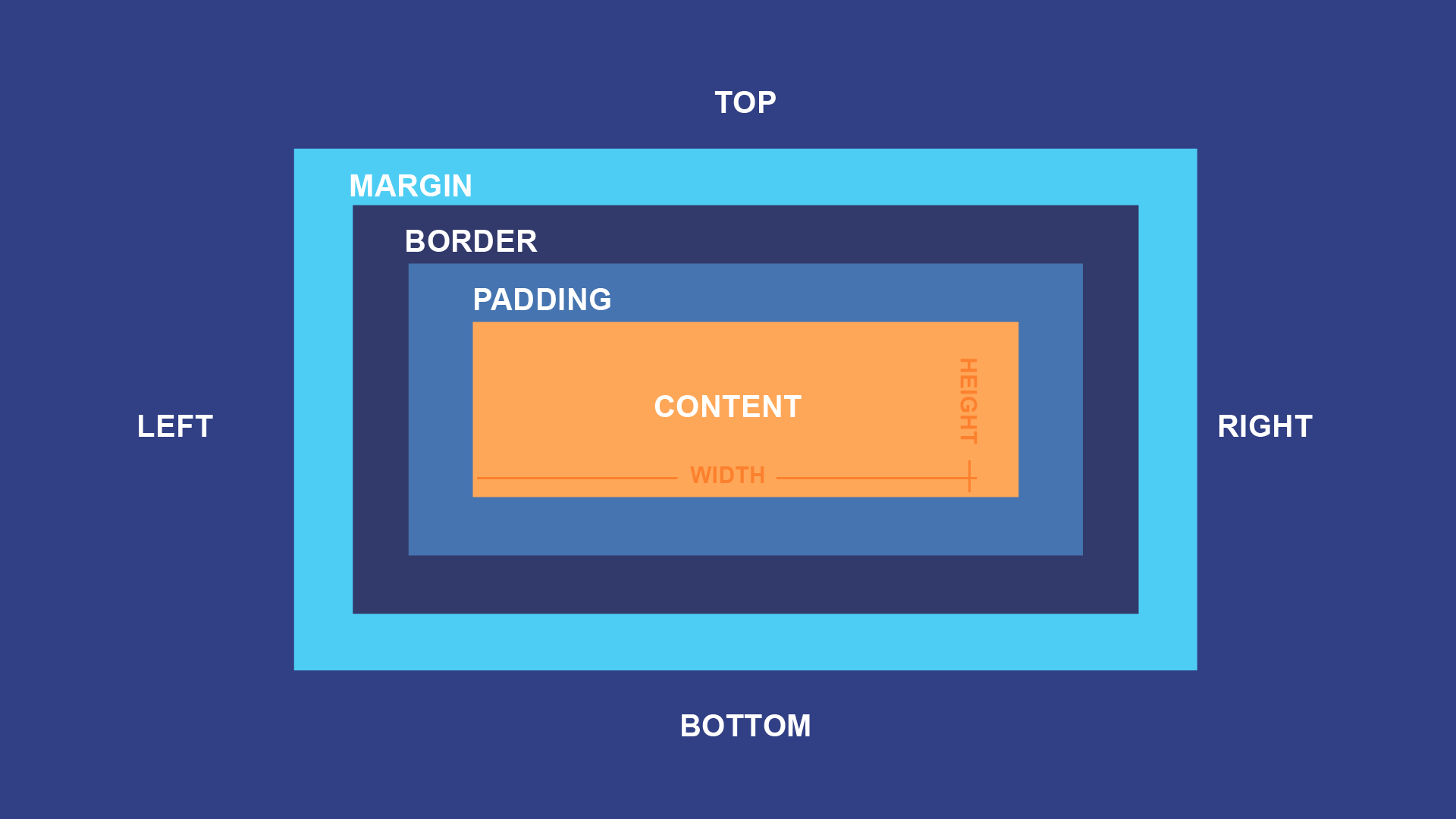
**Features Of CSS:**

* ***Opportunity in Web designing:***If anyone wants to begin a career in web designing professionally, it is essential to have knowledge of CSS and HTML.
* ***Website Design:***With the use of CSS, we can control various styles, such as the text color, the font style, the spacing among paragraphs, column size and layout, background color and images, design of the layout, display variations for distinct screens and device sizes, and many other effects as well**.**
* ***Web Control:***CSS has controlling power on the documents of HTML, so it is easy to learn. It is integrated with the HTML and the XHTML markup languages.
* ***Other Languages:***Once we have knowledge of some basics of CSS and HTML, other associated technologies like Angular, PHP, and JavaScript are become clearer to understand.

**CSS Box Model:**

A CSS box model is a compartment that includes numerous assets, such as edge, border, padding and material. It is used to develop the design and structure of a web page. It can be used as a set of tools to personalize the layout of different components. According to the CSS box model, the web browser supplies each element as a square prism.

The following diagram illustrates how the CSS properties of width, height, padding, border and margin dictate that how much space an attribute will occupy on a web page.

****

**Introduction to JavaScript:**

JavaScript is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for webpages. It is well-known for the development of web pages; many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript is both imperative and declarative type of language. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

**Features of JavaScript:**

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.

**System Analysis:**

**Feasibility Study:**

The feasibility study is an evaluation and analysis of the potential of a proposed project which is based on extensive investigation and research to support the process of decision making. Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study.

|  |  |
| --- | --- |
| **Feasibility Study** | |
| System: Sentiment Analysis | Date: 20-01-2023 |
| Author: Sangram Gupta | Page: 1 |
| **Product** | |
| The product is a Sentiment analysis web application which will help the e-commerce companies to predict the customer satisfactions, based on their reviews. | |
| **Technical Feasibility** | |
| The Sentiment Analysis web app is created using Visual Studio Code and python flask. | |
| **Social Feasibility** | |
| Some Sentiment statements of end users are required. All the users should be e-commerce user and also provide proper feedbacks. | |
| **Market Research** | |
| Market research says that this web app would be useful for the users and business owners as it could seamlessly help them to improve their lifestyle. | |
| **Economic Feasibility** | |
| The application can be developed within budget. | |
| **Alternative Solution** | |
| Could be use in android applications. | |

**Existing System:**

* The welcome page will appear in the index page named “Sentiment Analysis Vader Sentiment”
* One message box will be displayed in the welcome page and it will take all the user inputs(comments/reviews).
* After providing the review statement in the message box the user has to click on submit.
* The model will predict the sentiment score and also predict the percentage of positivity in the statement.
* Based on this web application the e-commerce business analysts can improve their products and also rank their products based on positive reviews.

All these endings demand a new web application which will reduce the manual works and do everything automatically. Also, there are some drawbacks in the system which motivates to upgrade the system.

Those drawbacks are follows:

* There is no database available for holding multiple users’ reactions and reviews in a queue.
* There is no database available for holding the sentiment scores and positiveness of a statement.
* There are possibilities to hanging down the web application.

**Proposed System:**

* As this is a web application of a sentiment analysis tool, it needs an ecommerce website to attach the tool in the backend of the user’s reviews. So firstly, to deploy the app it needs an ecommerce website for future use.
* One database should be implemented in the backend to gather the user statements and to save all the sentiment scores in the database for future use.
* An admin panel should be implemented to check and maintain the databases(add/remove/update) depends on future use.
* Accuracy score (Sentiment Score) of the model can be improve in future

**Table of Comparison:**

Comparison of Existing System & Proposed System:

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Existing System** | **Proposed System** |
| **Admin Panel** | Not Required | Required |
| **Time** | More Time Consuming | Less Time Consuming |
| **Database** | Not Required | Required |
| **Reliability** | Less | More |

**Hardware Requirements:**

* Core i5/i7 processor
* At least 8 GB Ram
* At least 60 GB of Usable Hard Disk Space

**Software Requirements:**

* Python 3.x
* Visual Studio Code Editor
* NLTK Toolkit
* Flask
* Unix/Linux/Windows Operating System

**Diagram:**

**Data Flow Diagram:**

A data low diagram (DFD) is a graphical representation of the "low" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system. DFDs can also be used for the visualization of data processing. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

**Level 0 DFD:**

Reviews/Comments

Sentiment Score

Sentiment App

User

**Use Case Diagram:**

Reviews

Sentiment Score

Submit

Sentiment Analysis App

User

Application

**Entity Relationship Diagram:**

negative

positive

Sentiment

Score

neutral

no

yes

Repeat?

Stop

Finish

Close

Submit

Feedbacks

review

User

**Form Design:**

**Components:**

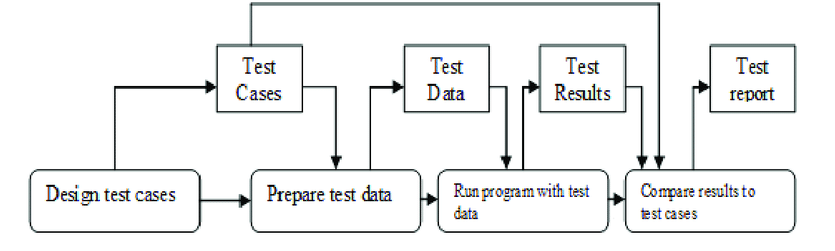
* Textarea
* Button
* Label
* Input

**Testing:**

* **Objective:**

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. We will also test the user friendliness of our app. The application will be used as an important tool, but we would like to ensure that it could be run on a variety of platforms with little impact on performance or usability.

* **Process Overview:**
* **Step-1: Assess Development Plan and Status**  
  This initiative may be prerequisite to putting together Verification, Validation, and Testing Plan wont to evaluate implemented software solution. During this step, testers challenge completeness and correctness of event plan. Based on extensiveness and completeness of Project Plan testers can estimate quantity of resources they’re going to get to test implemented software solution.
* **Step-2: Develop the Test Plan**   
  Forming plan for testing will follow an equivalent pattern as any software planning process. The structure of all plans should be an equivalent, but content will vary supported degree of risk testers perceive as related to software being developed.
* **Step-3: Test Software Requirements**   
  Incomplete, inaccurate, or inconsistent requirements cause most software failures. The inability to get requirement right during requirements gathering phase can also increase cost of implementation significantly. Testers, through verification, must determine that requirements are accurate, complete, and they do not conflict with another.
* **Step-4: Test Software Design**   
  This step tests both external and internal design primarily through verification techniques. The testers are concerned that planning will achieve objectives of wants, also because design being effective and efficient on designated hardware.
* **Step-5: Build Phase Testing**   
  The method chosen to build software from internal design document will determine type and extensiveness of testers needed. As the construction becomes more automated, less testing are going to be required during this phase. However, if software is made using waterfall process, it’s subject to error and will be verified. Experience has shown that it’s significantly cheaper to spot defects during development phase, than through dynamic testing during test execution step.
* **Step-6: Execute and Record Result**   
  This involves testing of code during dynamic state. The approach, methods, and tools laid out in test plan are going to be wont to validate that executable code actually meets stated software requirements, and therefore the structural specifications of design.
* **Step-7: Acceptance Test**   
  Acceptance testing enables users to gauge applicability and usefulness of software in performing their day-to-day job functions. This tests what user believes software should perform, as against what documented requirements state software should perform.
* **Step-8: Report Test Results**   
  Test reporting is continuous process. It may be both oral and written. It is important that defects and concerns be reported to the appropriate parties as early as possible, so that corrections can be made at the lowest possible cost.
* **Testing Process:**



* **Requirement Analysis:**

The first step of the manual testing procedure is requirement analysis. In this phase, tester analyses requirement document of SDLC (Software Development Life Cycle) to examine requirements stated by the client. After examining the requirements, the tester makes a test plan to check whether the software is meeting the requirements or not.

* **Test Plan Creation:**

Test plan creation is the crucial phase of STLC where all the testing strategies are defined. Tester determines the estimated effort and cost of the entire project. This phase takes place after the successful completion of the Requirement Analysis Phase. Testing strategy and effort estimation documents provided by this phase. Test case execution can be started after the successful completion of Test Plan Creation.

* **Environment Setup:**

Setup of the test environment is an independent activity and can be started along with Test Case Development. This is an essential part of the manual testing procedure as without environment testing is not possible. Environment setup requires a group of essential software and hardware to create a test environment. The testing team is not involved in setting up the testing environment, its senior developers who create it.

* **Test Case Execution:**

Test case Execution takes place after the successful completion of test planning. In this phase, the testing team starts case development and execution activity. The testing team writes down the detailed test cases, also prepares the test data if required. The prepared test cases are reviewed by peer members of the team or Quality Assurance leader.

RTM (Requirement Traceability Matrix) is also prepared in this phase. Requirement Traceability Matrix is industry level format, used for tracking requirements. Each test case is mapped with the requirement specification. Backward & forward traceability can be done via RTM.

* **Defect Logging:**

Testers and developers evaluate the completion criteria of the software based on test coverage, quality, time consumption, cost, and critical business objectives. This phase determines the characteristics and drawbacks of the software. Test cases and bug reports are analyzed in depth to detect the type of defect and its severity.

Defect logging analysis mainly works to find out defect distribution depending upon severity and types. If any defect is detected, then the software is returned to the development team to fix the defect, then the software is re-tested on all aspects of the testing.

* **Test Cycle Closure:**

The test cycle closure report includes all the documentation related to software design, development, testing results, and defect reports.

**Source Code:**

**Form.html :**

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

}

.c11{

border: 3px solid blue;

}

</style>

<title>{{ title }} Sentiment Analysis</title>

<!--link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='style.css') }}"-->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body><div class="container c11" >

<div class="container-fluid p-5 bg-success text-white text-center">

<h1>Sentiment Analysis VADER Sentiment</h1>

<p>A machine learning end to end flask web app for <b>"Sentiment Analysis" </b>model created using NLTK &amp; VADER Sentiment.</p>

</div>

<!--h1 align="center">A machine learning end to end flask web app for <b>"Sentiment Analysis" </b>model created using Scikit-learn &amp; VADER Sentiment.</h1-->

</br></br><div align="center">

<form method="POST">

<textarea name="text1" placeholder="Say Something: ...." rows="10" cols="109"></textarea><br><br>

<input class="btn btn-success" type="submit">

</form>

</div>

{% if final %}

<!--result start -->

</br></br> <div align="center">

</br> </br>

<h2 class="bg-success text-white">The Sentiment of</h2>

'{{ text1 }}'

<h2>is {{ final \*100}}% positive !</h2>

<h2>Score table</h2>

<div class="container table-responsive-sm">

<table class="table table-bordered">

<tr>

<th>SENTIMENT METRIC</th>

<th>SCORE</th>

</tr>

<tr>

<td>Positive</td>

<td>{{text2}}</td>

</tr>

<tr>

<td>Neutral</td>

<td>{{text3}}</td>

</tr>

<tr>

<td>Negative</td>

<td>{{text5}}</td>

</tr>

<tr>

<td>Compound</td>

<td>{{text4}}</td>

</tr>

</table>

</div>

{% else %}

<p></p>

{% endif %}

</div></br></br>

<!--result end -->

</div> </body>

</html>

**Form.html.bak:**

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

}

.c11{

border: 3px solid blue;

}

</style>

<title>{{ title }} Sentiment Analysis</title>

<!--link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='style.css') }}"-->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body><div class="container c11" >

<div class="container-fluid p-5 bg-success text-white text-center">

<h1>Sentiment Analysis VADER Sentiment</h1>

<p>A machine learning end to end flask web app for <b>"Sentiment Analysis" </b>model created using NLTK &amp; VADER Sentiment.</p>

</div>

<!--h1 align="center">A machine learning end to end flask web app for <b>"Sentiment Analysis" </b>model created using Scikit-learn &amp; VADER Sentiment.</h1-->

</br></br><div align="center">

<form method="POST">

<textarea name="text1" placeholder="Say Something: ...." rows="10" cols="109"></textarea><br><br>

<input class="btn btn-success" type="submit">

</form>

</div>

{% if final %}

<!--result start -->

</br></br> <div align="center">

</br> </br>

<h2 class="bg-success text-white">The Sentiment of</h2>

'{{ text1 }}'

<h2>is {{ final \*100}}% positive !</h2>

<h2>Score table</h2>

<div class="container table-responsive-sm">

<table class="table table-bordered">

<tr>

<th>SENTIMENT METRIC</th>

<th>SCORE</th>

</tr>

<tr>

<td>Positive</td>

<td>{{text2}}</td>

</tr>

<tr>

<td>Neutral</td>

<td>{{text3}}</td>

</tr>

<tr>

<td>Negative</td>

<td>{{text5}}</td>

</tr>

<tr>

<td>Compound</td>

<td>{{text4}}</td>

</tr>

</table>

</div>

{% else %}

<p></p>

{% endif %}

</div></br></br>

<!--result end -->

</div> </body>

</html>

**Style.css:**

html {

background: #404550;

}

body {

font: 100% Arial, Helvetica, sans-serif;

line-height: 1.5;

position: relative;

background: #fff;

color: rgb(76, 67, 65);

font-weight:normal;

font-style:normal;

width: 1280px;

margin: 20 auto;

padding: 20px;

}

form, div{

border-bottom: 2px solid rgb(76, 67, 65);

margin-bottom: 5em;

margin-left: auto;

margin-right: auto;

width: 50em

}

h1 {

font-family: Georgia, Times, "Times New Roman", serif;

font-size: 1.8em;

border-bottom: 2px solid rgb(76, 67, 65);

margin-bottom: 1.5em;

background: url(../\_images/icon\_sprites\_50.png) no-repeat

}

.example\_a {

border: none;

background: #404040;

color: #ffffff !important;

font-weight: 100;

padding: 20px;

text-transform: uppercase;

border-radius: 6px;

display: inline-block;

transition: all 0.3s ease 0s;

float: right;

margin-top: 2em;

}

.example\_a:hover {

color: #404040 !important;

font-weight: 700 !important;

letter-spacing: 3px;

background: none;

-webkit-box-shadow: 0px 5px 40px -10px rgba(0,0,0,0.57);

-moz-box-shadow: 0px 5px 40px -10px rgba(0,0,0,0.57);

transition: all 0.3s ease 0s;

}

**Style.css.bak:**

html {

background: #404550;

}

body {

font: 100% Arial, Helvetica, sans-serif;

line-height: 1.5;

position: relative;

background: #00004d;

color: rgb(76, 67, 65);

font-weight:normal;

font-style:normal;

width: 1280px;

margin: 20 auto;

padding: 20px;

}

form, div{

border-bottom: 2px solid rgb(76, 67, 65);

margin-bottom: 5em;

margin-left: auto;

margin-right: auto;

width: 50em

}

h1 {

font-family: Georgia, Times, "Times New Roman", serif;

font-size: 1.8em;

border-bottom: 2px solid rgb(76, 67, 65);

margin-bottom: 1.5em;

background: url(../\_images/icon\_sprites\_50.png) no-repeat

}

.example\_a {

border: none;

background: #404040;

color: #ffffff !important;

font-weight: 100;

padding: 20px;

text-transform: uppercase;

border-radius: 6px;

display: inline-block;

transition: all 0.3s ease 0s;

float: right;

margin-top: 2em;

}

.example\_a:hover {

color: #404040 !important;

font-weight: 700 !important;

letter-spacing: 3px;

background: none;

-webkit-box-shadow: 0px 5px 40px -10px rgba(0,0,0,0.57);

-moz-box-shadow: 0px 5px 40px -10px rgba(0,0,0,0.57);

transition: all 0.3s ease 0s;

}

**app.py:**

from flask import Flask, request, render\_template

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

import nltk

from string import punctuation

import re

from nltk.corpus import stopwords

nltk.download('stopwords')

set(stopwords.words('english'))

app = Flask(\_\_name\_\_)

@app.route('/')

def my\_form():

return render\_template('form.html')

@app.route('/', methods=['POST'])

def my\_form\_post():

stop\_words = stopwords.words('english')

#convert to lowercase

text1 = request.form['text1'].lower()

text\_final = ''.join(c for c in text1 if not c.isdigit())

#remove punctuations

#text3 = ''.join(c for c in text2 if c not in punctuation)

#remove stopwords

processed\_doc1 = ' '.join([word for word in text\_final.split() if word not in stop\_words])

sa = SentimentIntensityAnalyzer()

dd = sa.polarity\_scores(text=processed\_doc1)

compound = round((1 + dd['compound'])/2, 2)

return render\_template('form.html', final=compound, text1=text\_final,text2=dd['pos'],text5=dd['neg'],text4=compound,text3=dd['neu'])

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True, host="127.0.0.1", port=5002, threaded=True)

**launch.json:**

{

// Use IntelliSense to learn about possible attributes.

// Hover to view descriptions of existing attributes.

// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387

"version": "0.2.0",

"configurations": [

{

"name": "Python: Current File",

"type": "python",

"request": "launch",

"program": "${file}",

"console": "integratedTerminal"

},

{

"type": "pwa-chrome",

"request": "launch",

"name": "Launch Chrome against localhost",

"url": "http://localhost:8080",

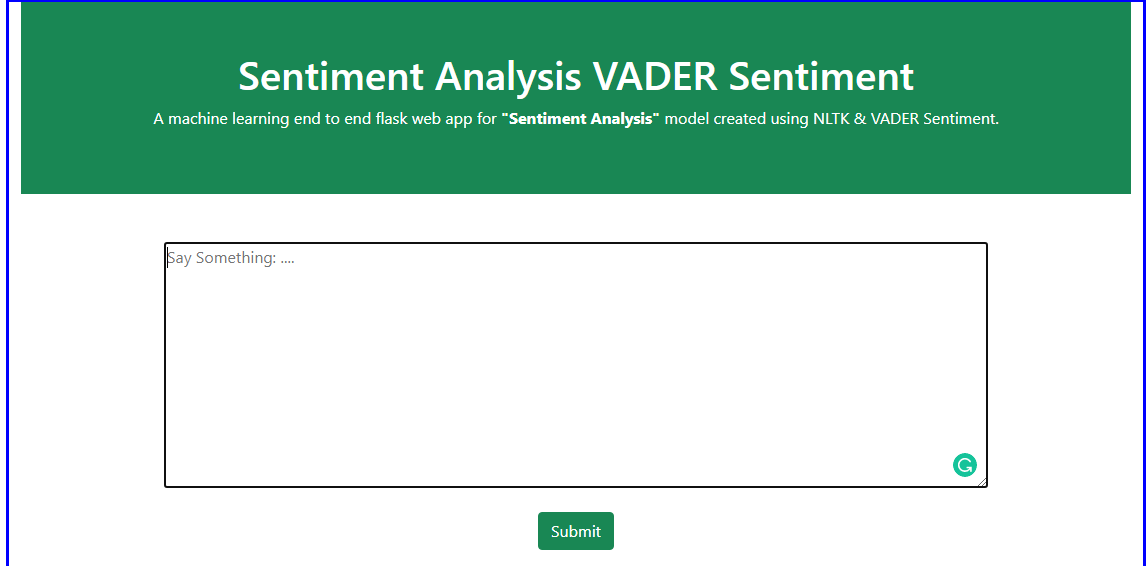
"webRoot": "${workspaceFolder}"

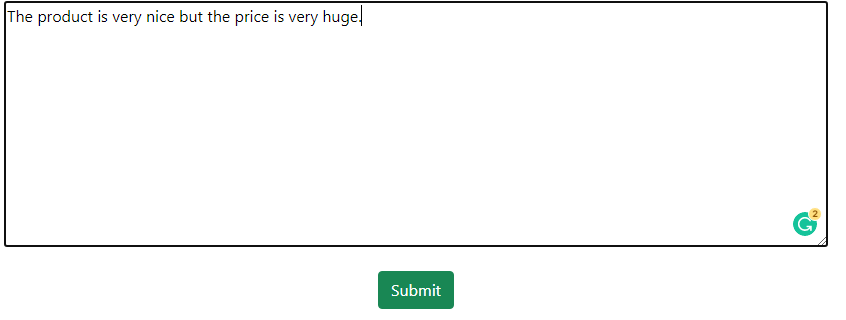
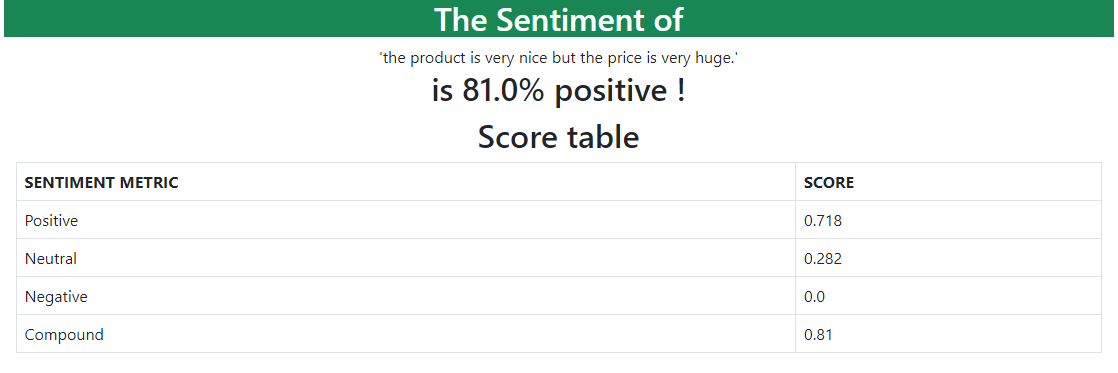
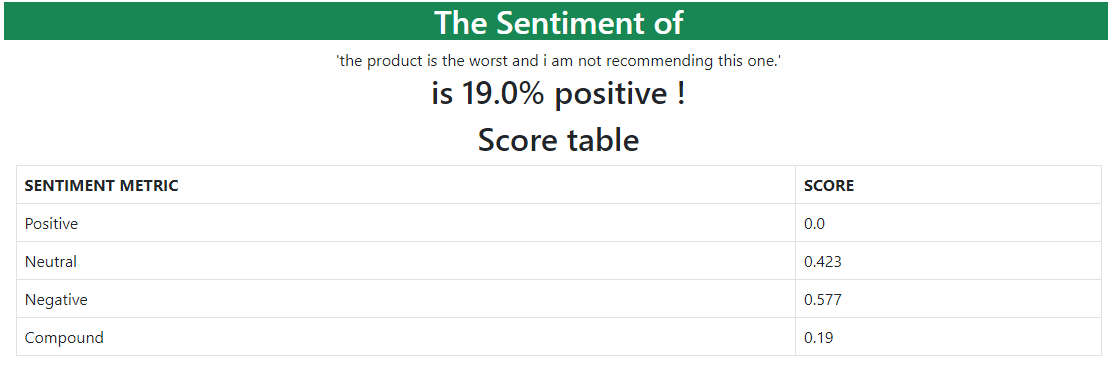
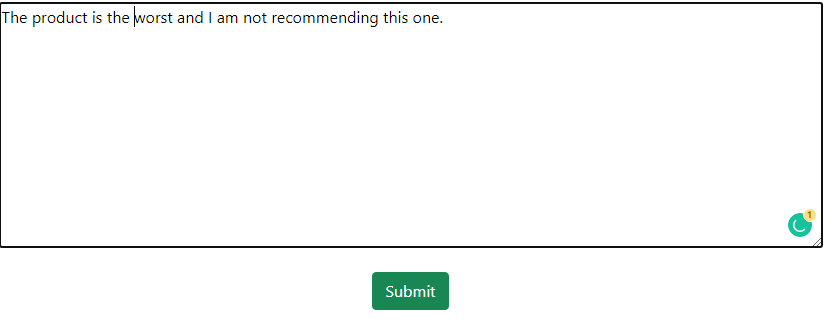
}

]

}

**Snapshots of all the web pages:**

****

****

**Conclusion:**

Sentiment analysis deals with the classification of texts based on the sentiments they contain. This article focuses on a typical sentiment analysis model consisting of three core steps, namely data preparation, review analysis and sentiment classification, and describes representative techniques involved in those steps. Sentiment analysis is an emerging research area in text mining and computational linguistics, and has attracted considerable research attention in the past few years. Future research shall explore sophisticated methods for opinion and product feature extraction, as well as new classification models that can address the ordered labels property in rating inference. Applications that utilize results from sentiment analysis is also expected to emerge in the near future.

**References:**

* S. ChandraKala1 and C. Sindhu2, “OPINION MINING AND SENTIMENT CLASSIFICATION: A SURVEY,”.Vol .3(1),Oct 2012,420-427
* G.Angulakshmi , Dr.R.ManickaChezian ,”An Analysis on Opinion Mining: Techniques and Tools”. Vol 3(7), 2014 [www.iarcce.com](http://www.iarcce.com).
* Callen Rain,”Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning” Swarthmore College, Department of Computer Science
* Kavya Suppala, Narasinga Rao “Sentiment Analysis Using Naïve Bayes Classifier”International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-8 June, 2019.
* Bhagyashri Wagh,J.V. Shinde ,N.R.Wankhade”Sentiment Analysis on Twitter Data using Naive Bayes” International Journal of Advanced Research in Computer and Communication Engineering ISO 3297: 2007 Certified
* Machine Learning Tom M. Mitchell
* Akshi kumar, Prakhar Dogra and Vikrant Dabas “Emotion Analysis of Twitter using Opinion Mining” Dept. of Computer Engineering , Delhi Technology University,New Delhi India
* Shiv Dhar, Suyog Pednekar, Kishan Borad- Methods for Sentiment Analysis Computer Engineering, VIVA Institute of Technology, University of Mumbai, India.

**Bibliography:**

**•** [**www.stackoverlow.com**](http://www.stackoverlow.com)

**•** [**www.w3schools.com**](http://www.w3schools.com)

**•** [**www.wikipedia.org**](http://www.wikipedia.org)