Visvesvaraya Technological University

Belagavi, Karnataka- 590 018

A

PROJECT REPORT

ON

“SENTIMENTAL ANALYSIS WITH WEB SCRAPING”

Submitted in partial fulfilment of the requirements for the ***Project Work Phase II***

(17CSP85*)* *Course of the 8th semesters*

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY,

Dhruva V	Pruthviraj G	Purushothama N	Gagan M S
1JS18CS403	1JS18CS416	1JS18CS420	1JS17CS035

UNDER THE GUIDANCE OF

Mrs. SAVITA S
Assistant Professor, Dept of CSE,
JSSATE, BENGALURU

Mrs. RANJITHA S R
Assistant Professor, Dept of CSE,
JSSATE, BENGALURU

JSS Academy of Technical Education

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING 2020-2021

Kengeri-Uttarahalli Main Road, Bangalore-560060

JSS Mahavidyapeetha

JSS Academy of Technical Education

Kengeri-Uttarahalli Main Road, Bangalore-560060

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING 2020-2021

CERTIFICATE

Certified that the project work entitled **“SENTIMENTAL ANALYSIS WITH WEB SCRAPING”** carried out by **Mr. Dhruva V,** USN **1JS18CS403, Mr. Pruthviraj G,** USN **1JS18CS416**, **Mr. Purushothama N,** USN **1JS18CS420**, **Mr. Gagan M S,** USN **1JS17CS035** a bonafide student of **JSS Academy of Technical Education** in partial fulfillment for the award of Bachelor of Engineering / Bachelor of Technology in **Project Work Phase II (17CSP85)** of the Visveswaraiah Technological University, Belgaum during the year **2020-2021**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Mrs. SAVITA S	**Mrs. RANJITHA S R**	**Dr. NAVEEN N C**
Assistant Professor,	Assistant Professor,	HOD & Professor,
Dept of CSE,	Dept of CSE,	Dept of CSE,
JSSATEB, Bengaluru	JSSATEB, Bengaluru	JSSATEB, Bengaluru

External Viva

Name of the examiners Signature with date

1

2.

ABSTRACT:

E-commerce is getting used more and more these days to purchase products in an online store. A product review is usually used see if the product is worth buying or not.

In that sense, the present work proposes an innovative solution by combing a web-scraping unit which is used to read the reviews from a website, Vader sentimental analyzer which is used to analyze the reviews of a product and return if the review is positive or negative or neutral, and a wordcloud which is an image containing positive words the positive words are selected by textblob module.

This product used by selecting URL a product in amazon.in website opened in a browser, copy the URL first and open this product website and paste it in the input box and press enter, then this product processes the request and shows the score that the selected product has got and a wordcloud image

The result of this project has shown success by show the result.

ACKNOWLEDGMENTS

We express our humble pranams to His Holiness **Jagadguru Sri Sri Sri Shivarathreeshwara Mahaswamiji** for showering his blessings on us to receive good education and have a successful career.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of support from all ends during this project. So, we take this opportunity to express our gratitude to all whose guidance and encouragement helped us emerge successful.

We are thankful for the resourceful guidance, timely assistance and graceful gesture of our guide **Mrs. SAVITA S** and **Mrs. RANJITA S R,** Assistant professors of Department Computer Science and Engineering, who has helped us in every aspect of our project work.

We are also grateful to **Dr. Naveen N C,** Head of the Department, Computer Science and Engineering, for his unending support, guidance and encouragement in all our ventures.

We express our sincere thanks to our beloved principal, **Dr. Mrityunjaya V Latte** for having supported us in all academic endeavours.

Last but not the least, we would be immensely please to express our heartfelt thanks to all the teaching and non-teaching staff of the Department of CSE and our friend for their timely help, support and guidance.

Dhruva V

TABLE OF CONTENTS

TITLE	PAGE NO
CHAPTER 1. INTRODUCTION	
1.1 Introduction	5
1.2 Aims and Objectives	6
1.3 Problem Statement	6
1.4 Motivation	7
1.5 Literature Survey	7
CHAPTER 2. SOFTWARE REQUIREMENTS	
2.1 Existing System	9
2.2 Proposed System	9
2.3 System Requirements	9
CHAPTER 3. DESIGN	
3.1 System Design	10
CHAPTER 4. IMPLEMENTATION	
4.1 Implementation	12
4.2 Codes	14
CHAPTER 5. SNAPSHOTS	28
CHAPTER 6. CONCLUSION AND FUTURE WORK	30
CHAPTER 7. REFERENCES	31

TABLE OF FIGURES

TITLE	PAGE NO
1. Fig 3.1.1 The overview of our project	10
2. Fig 4.1.1 The Front End	12
3. Fig 4.1.2 The Web Scraping Unit	12
4. Fig 4.1.3 The Cleaning Unit	13
5. Fig 4.1.4 The VADER UNIT	13
6. Fig 4.1.1 The EDA Unit	14
7. Fig 5.1.1 The Amazon page	28
8. Fig 5.1.2 The Main Page	28
9. Fig 5.1.3 The Result Page	29

CHAPTER 1: INTRODUCTION

1.1 Introduction

Sentiment analysis also known as opinion mining in the field of natural language processing (NLP) that builds systems that tries to identify and extract the opinions within text. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall tonality of the document. In the recent years, the exponential increase in the internet usage and exchange of public opinions is driving the force behind the sentiment analysis today. The web is a huge repository of structured and unstructured data. The analysis of this data to extract latent public opinion and sentiment is a challenging task. Technology has turned into a fundamental piece of everybody's life. Social media technology is already used widely by the public to speak out once mind openly. This data can be leveraged to have a better understanding of the current state of decision making.

Machine learning approach is used for analyzing sentiments from the text .by the sentiment analysis in the specific domain, it is possible to identify the effect of domain info in sentiment classification.

Web scrapping (web harvesting, web data extraction) is a technique employed to extract the large amounts of data from the websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format. Web scrapping may access the world wide web directly using the hypertext transfer protocol, or through the web browser. Web scrapping, a web page involves fetching it and extracting from it. Fetching is the downloading of the page, once fetched then extraction can take place. The content of the page may be parsed, searched, reformatted, its data copied into a spreadsheet and so on. Web scraping is used for contact scraping, and as a component of applications used for web indexing, web mining and data mining, online price change monitoring and price comparison, product review scraping (to watch the competition), gathering real estate listings, weather data monitoring, website change detection, research, tracking online presence and reputation, web mashup and, web data integration. Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. However, most web pages are designed for human end-users and not for ease of automated use. As a result, specialized tools and software have been developed to facilitate the scraping of web pages.

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

1.2 Aims and Objectives

AIMS:

1) Our first aim is by using the web scrapping we generate the unstructured data from the amazon product pages.
2) To the scrapped data, we create a machine learning model and it is used for analyzing the sentiments of the cleaned data.
3) By doing so, with the help of the model we can classify the sentiment intensity (positive, negative or neutral) of the scrapped data.
4) At last the output will be integrated with the help of flask and bootstrap. (extra: bootstrap-->it is the most popular html, CSS and JS library in the world)

OBJECTIVES:

1) The main objective behind this project is to provide a platform which will enable users to check the credibility of a retailer/product by scanning reviews.
2) Instead of going through potentially hundreds of reviews, our platform offers a one-click result.

1.3 Problem Statement

The Primary focus of this project is to find the sentimental scores of a product review by the customers. By doing this we can find the sentimental scores of amazon product in amazon.in website, we can use this score to see if this product has good scores think about buying it. If the product has good positive reviews we can say that the product has satisfied the customers and if it has high negative reviews then we can say that it has not satisfied the customers. By using this project result we can think of buying the product or not. It shows a word cloud with words from the reviews. It shows the reviews with highest and lowest polarity.

1.4 Motivation

The main motivation behind the project is to understand the reviews submitted online by varying customers for a product. With knowing the product has positive or negative reviews we can have a clear understanding of the reviews submitted.

1.5 Literature Survey

Vidhi singrodia (2019) [1] proposed Web scraping is a recognizable phrase which has expanded significance owing to the requirement of “free” data accumulated in PDF documents or web pages. Numerous professionals and researchers require the data for processing, analysis and extraction of significant consequences. Alternatively, people dealing with B2B use cases require the admittance of data from several sources for its integration into innovative applications which will offer supplementary values and novelty. Throughout this paper we have reviewed the various aspects of Web Scrapper. Anirban mitra and subratapaul throughout this paper reviewed the various aspects of Web Scraper. Starting with the tools and software for web scraping, the operating principle, strength and drawbacks and finally the applications of web scraping systems. There are numerous features which can be reflected while the usage of a web scraper. This may be generalized on the solicitations of scraping. Likewise, the imprecise authority of construction of a project on the basis of scraped data creates it challenging to differentiate projects which are dependent on scraping machineries. We can, nevertheless provide an overview on the maximum characteristic arenas the technology is used in. Through some of these arenas will provide small instances on the means by which such a project could provide.

D. Deepa (2019) [2] proposed Natural Language Processing (NLP) is a study of computational treatment of human language in order to make it understandable for computers. It is used in the research fields like artificial intelligence, information engineering, statistics, sentiment analysis and linguistics. Sentiment analysis plays significant role in NLP which performs the series of operations computationally to identify and categorize the sentiment conveyed in segment of words. The proposed approach is to detect the polarity of words from twitter using feature extraction and dictionary-based methods. Raaji and Tamilarasj experimented for both feature engineering and dictionary-based techniques are trained and tested. The better results are obtained from the feature engineering. But the drawback of feature engineering is tedious and time consuming and all of the code wrote over for many hours cannot be applied to any other problem. These problems can be overcome in the dictionary-based approaches. In the future work, performance can be compared with more algorithms like Naïve Bayes, Linear SVM and Artificial Neural Network and optimization techniques can be performed for score adaptation to increase the accuracy in Dictionary based approach.

Harshavadantalpada (2019) [3] suggests that lexical and semantic-based methods for sentiment prediction offer better accuracy than Deep Learning methods. When a large enough and evenly distributed training dataset is not available. We observed that domain-specific knowledge affects the prediction accuracy of sentiment, mainly when the target text contains more domain-specific words. Malka N Halgamuge observed that domain-specific knowledge affects the prediction accuracy of sentiment, mainly when the target text contains more domain-specific words. Accuracy of Deep Learning methods is dependent on the quality of the training dataset and the distribution of the classes within the dataset. Nguyentranquocvinh says social media produces a large amount of data which can be utilized for data-driven or information-driven decision making. Among the lexical based methods, VADER shows the highest accuracy as it considers the semantic factor when making the prediction. In our case, we observed that telemedicine has a high number of positive sentiments. It is still in its infancy and has not spread to a broader demographic.

Shreya Upadhyay (2017) [4] proposes Massive volumes of data are generated by various users, entities, applications and disseminated online. This copious volume of big data is distributed across millions of websites and is available for various applications. Search engines do provide a simple mechanism to access this data. Accessing this data using search engines requires a user to spend time and resources to manually click and download. Clearly, such a manual approach is not scalable for a vast majority of real life applications at the enterprise and organization level.

CHAPTER 2: SOFTWARE REQUIREMENTS

2.1 EXISTING SYSTEM:

The existing system has many limitations:

1) Sentiment model is a separate unit that hasn’t been integrated with web scrapping
2) It is more complex to implement and its expensive. There is lack of credible user interface.

2.2 PROPOSED SYSTEM:

1) Integration of both modules (web scrapper and sentiment analyzer) into a single unit.
2) Developing a user interface using flask and bootstrap.
3) Providing better result on the scrapped Data by implementing VADER intensity analyzer.
4) VADER allows us to rate the reviews based on the emotions in the text.
5) We generate a word cloud, which is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance.

2.3 System Requirements: -

The Modules used in this project are:

Devices	: Local / Personal Computer.
OS	: Windows(Local Computer)
OS Distro	: Windows 10 (Local Computer)
Storage	: SD Card (8GB minimum)
Battery	: Power Bank / Li-ion(Optional if required)
Application	: python
Browser	: Google chrome or any other.

CHAPTER 3: DESIGN

3.1 System Design:

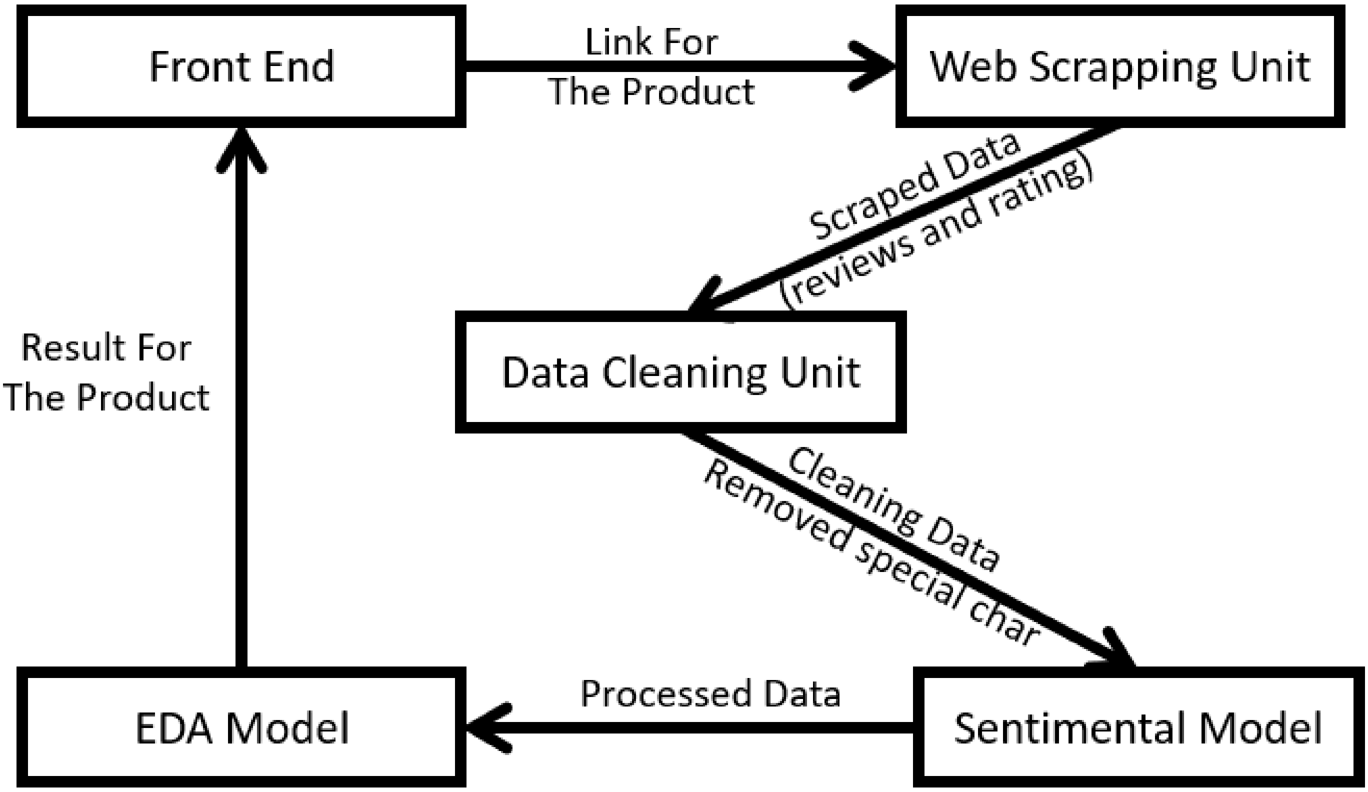


Fig 3.1.1 The overview of our project

The user can interact with our product through the web page. It also contains a tutorial and a short explanation of how our product works. It was built using Flask & Bootstrap 4.

Generally, text data contains a lot of noise either in the form of symbols or in the form of punctuations and stopwords (Stopwords are the words in any language which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as “The Who” or “Take That”.). Therefore, it becomes necessary to clean the text, not just for making it more understandable but also for getting better insights

Sentiment Model accepts the cleaned data and determines whether a piece of writing is positive, negative or neutral.

The EDA (Exploratory Data Analysis) unit is a crucial part of any project because that’s where you get to know more about the data. In this phase, you can reveal hidden patterns in the data and generate insights from it.

CHAPTER 4: IMPLEMENTATION

4.1 Implementation:

Front End: When it comes to the front-end deployment, we are using flask and bootstrap templates. Flask is a micro-framework that helps in building reliable, scalable, and maintainable web applications. It manages the requests and responses to the flask server. We are using bootstrap templates to dynamically create new views for our users.

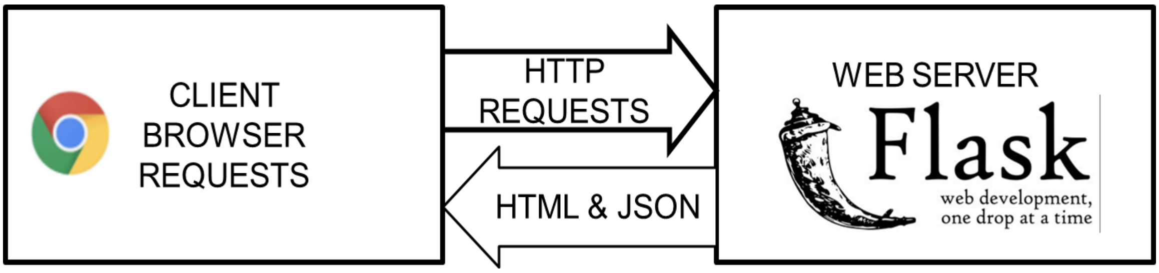


Fig 4.1.1 The Front End

Web Scraping Unit: The Web scraping unit is responsible for extracting the required data from the webpages. We are mainly using 1 library; Beautiful soup is a python library for pulling data out of html and xml pages. The extracted data is then stored in a csv file. After the user enters the URL, it is sent to the web scraping unit, where the required information is extracted and saved in a csv file.

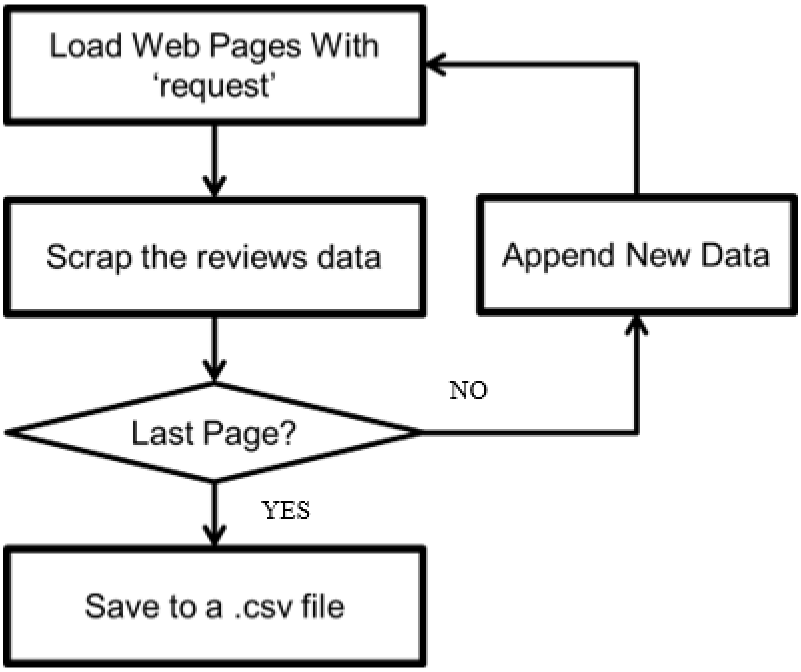


Fig 4.1.2 The Web Scraping Unit

Cleaning unit:

In the cleaning unit we remove the unwanted data from the text (i.e. scrapped text data) then we convert the text (review column) to lowercase and remove integers or numerical in text. Punctuations, null values and extra (spaces as they don't make any sense) and all these things are done by importing regular expressions (import re) Stopwords are this is in, which does not add much value. so, we remove them to decrease the size of dataset this process is called normalizing text (with spacy) spacy is most versatile and widely used library in NLP Lemmatization is nothing but normalization of words which means reducing a word into its root form.

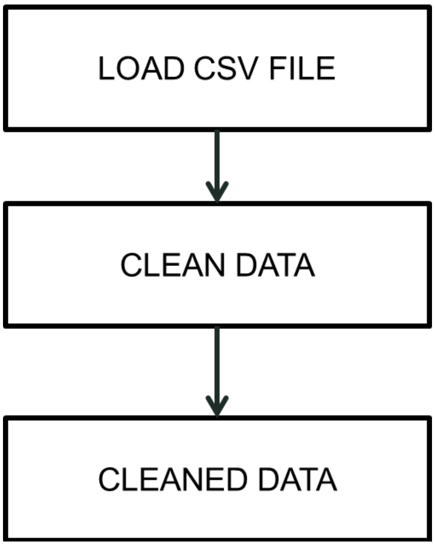


Fig 4.1.3 The Cleaning Unit

Sentiment analysis using VADER:

- VADER (Valence Aware Dictionary for sentiment Reasoning) is an Unsupervised model used that is adaptive to interpret emotional (positive/negative) and emotional (strength) text feelings.
- VADER is focused on the lexicons of words related to sentiment. Each of the words in the lexicon is rated as to whether it is positive or negative and assigns scores to them.
- It uses the polarity scores () method to get the sentiment metrics for a piece of text.
- Vader is an easy-to-use and powerful, this package that is based on lexicons of sentiment-related words.

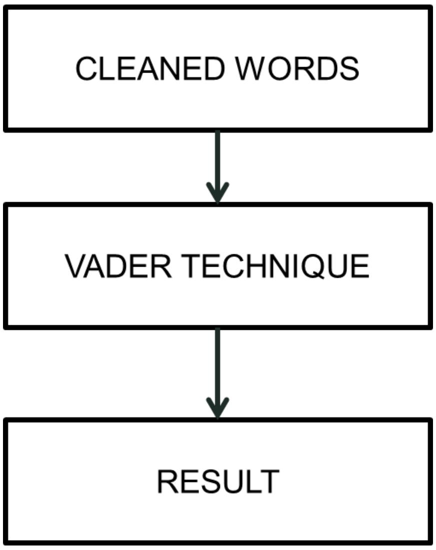


Fig 4.1.4 The VADER UNIT

Exploratory Data Analysis (EDA):

- It is the process of exploring data, generating insights, testing hypotheses, and revealing underlying hidden patterns in the data.
- In the large resource of data, we pick the required and clean it for Exploratory Data Analysis.
- we'll create a Document Term Matrix that we'll later use in our analysis to get the insights in data and with the help of wordcloud we display it in our project.

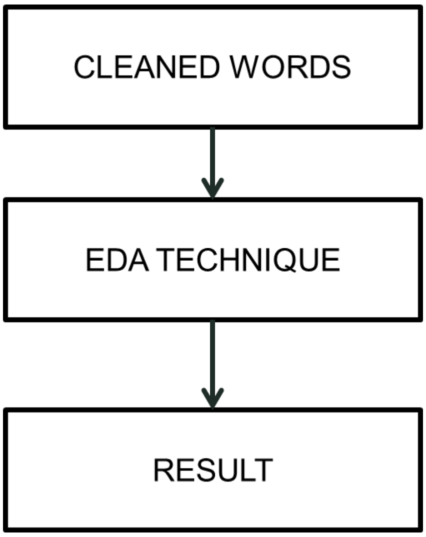


Fig 4.1.1 The EDA Unit

- From textblob sentiment polarity we pick the top most emotions contained words in the data.

4.2 Codes:

- app.py

```
from flask import Flask, render_template, request

import requests

import seaborn as sns

import numpy as np

import os

import os.path

from os import path

from bs4 import BeautifulSoup

import pandas as pd

import matplotlib.pyplot as plt

import re
import string

from amazon_product_review_scraper import amazon_product_review_scraper

import spacy

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

from sklearn.feature_extraction.text import CountVectorizer

from wordcloud import WordCloud

from textwrap import wrap

from textblob import TextBlob

import wordcloud_gen

app = Flask(__name__)

@app.route('/')

def index():

    if path.exists('scraped_data.csv'):

        os.remove('scraped_data.csv')

    return render_template('index.html', product='Product')

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

def process():

    url = request.form.get('url')

    if 'amazon.in' in url:

        product_asin = url[url.find('dp/')+3: url.find('dp/')+13]

        try:

            review_scraper = amazon_product_review_scraper( amazon_site="amazon.in",
product_asin=product_asin)

        except:
            return '<script>alert("Only works with
amazon.in");window.history.back();</script>'
        reviews_df, p_title, p_image = review_scraper.scrape()
        reviews_df['rating'] = reviews_df['rating'].str[:1].astype(int)
        with open('scraped_data.csv', 'w') as csv_file:
            reviews_df.to_csv('scraped_data.csv', index=False)
    else:
        return '<script>alert("Error in Input");window.history.back();</script>'
    print(p_title, p_image)
    df = pd.read_csv('scraped_data.csv')
    no_of_reviews = len(df)
    df = df[['content', 'rating']]
    df['cleaned'] = df['content'].apply(lambda x: re.sub('\w*\d\w*', '', x))
    df['cleaned'] = df['cleaned'].apply(lambda x: re.sub(
        '[%s]' % re.escape(string.punctuation), '', x))  # Remove Punctuations
    df['cleaned'] = df['cleaned'].apply(lambda x: re.sub(' +', ' ', x))
    # python -m spacy download en_core_web_sm
    nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner'])
    df['lemmatized'] = df['cleaned'].apply(lambda x: ' '.join(
        [token.lemma_ for token in list(nlp(x)) if (token.is_stop == False)]))
    df = df[['rating', 'lemmatized']]
    df_new = df.rename(columns={'lemmatized': 'content'})
    df = df_new
    sent = SentimentIntensityAnalyzer()
    sentiment_dict = []


    for i in range(0, len(df)):
        sentiment_dict.append(sent.polarity_scores(df.iloc[i, 1]))
    positive = []
    neutral = []
    negative = []
    compound = []
    for item in sentiment_dict:
        positive.append(item['pos'])
        neutral.append(item['neu'])
        negative.append(item['neg'])
        compound.append(item['compound'])
    sentiment_df = pd.DataFrame(list(zip(positive, neutral, negative, compound)),
columns=[
                                'Positive', 'Neutral', 'Negative', 'Compound'])
    df['Positive'] = sentiment_df['Positive']
    df['Negative'] = sentiment_df['Negative']
    df['Neutral'] = sentiment_df['Neutral']
    df['Compound'] = sentiment_df['Compound']
    print(df.columns)
    df_temp = df[['rating', 'content']]
    df_temp = df_temp.assign(new="1")
    df_grouped = df_temp[['new', 'content']].groupby(by='new').agg(lambda x: ' '.join(x))
```

```
cv = CountVectorizer(analyzer='word')

data = cv.fit_transform(df_grouped['content'])

df_dtm = pd.DataFrame(data.toarray(), columns=cv.get_feature_names())

df_dtm.index = df_grouped.index

def generate_wordcloud(data, title):

    wc = WordCloud(width=400, height=330, max_words=150,

                   background_color='white'). generate_from_frequencies(data)

    plt.figure(figsize=(10, 8))

    plt.imshow(wc, interpolation='bilinear')

    plt.axis("off")

    plt.title('\n'.join(wrap(title, 60)), fontsize=13)

    # plt.show()

    if path.exists('Project/static/wordcloud.png'):

        os.remove('Project/static/wordcloud.png')

    # wc.to_file('wordcloud.png')

    # return plt,wc

    plt.savefig('Project/static/wordcloud.png', format='png', dpi=500)

df_dtm = df_dtm.transpose()

for index, product in enumerate(df_dtm.columns):

    generate_wordcloud(df_dtm[product], product)

   # wordcloud_gen.generate_wordcloud(df_dtm[product], product)

highest_polarity = pd.DataFrame(columns=['content'])

lowest_polarity = pd.DataFrame(columns=['content'])

df['polarity'] = df['content'].apply(
        lambda x: TextBlob(x).sentiment.polarity)

    for index, Review in
enumerate(df.iloc[df['polarity'].sort_values(ascending=False)[:3].index]['content']):

        highest_polarity = highest_polarity.append(

            {'content': str(Review)}, ignore_index=True)

    for index, Review in
enumerate(df.iloc[df['polarity'].sort_values(ascending=True)[:3].index]['content']):

        lowest_polarity = lowest_polarity.append(

            {'content': str(Review)}, ignore_index=True)

    if float(df['Positive'].mean() * 10) > 6:

        verdict = 'This product is highly recommended!!!'

    elif float(df['Negative'].mean() * 10) < 0:

        verdict = 'This product is not recommended!'

    else:

        verdict = 'This product is recommended'

    return render_template('process.html', asin_id=product_asin, p_image=p_image,
p_title=p_title, len_r=no_of_reviews, row_data=list(highest_polarity.values.tolist()),
row2_data=list(lowest_polarity.values.tolist()), titles=highest_polarity.columns.values,
total_reviews=len(df), pos=str(df['Positive'].mean() * 10)[0:3],
neg=str(df['Neutral'].mean() * 10)[0:3], neutral=str(df['Negative'].mean() * 10)[0:3],
verdict=verdict)

if __name__ == '__main__':

    app.run(debug=True)
```

- index.html

```
<!DOCTYPE html>

<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> REVIEW ANALYZER </title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
rel="stylesheet"integrity="sha384+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPpr
VMTNDbiYZCxYbOOl7+AMvyTG2x" crossorigin="anonymous">
</head>
<body style="background-color: #CDF0EA;">
    <nav class="navbar navbar-light fixed-top" style="background-color: #C490E4;">
        <div class="container-fluid">
            <a class="navbar-brand" href="http://127.0.0.1:5000/">
                <img class="rounded" src="{{ url_for('static',filename='logo.jpg') }}"
alt="REVIEW ANALYZER" width="30"
                    height="24">
                REVIEW ANALYZER
            </a>
            <form class="d-flex" action="/process" method="POST">
                <input class="form-control me-2" type="search" placeholder="Amazon product
link" aria-label="text"
                    name="url">
                <button class="btn btn-outline-success" type="submit">Search</button>
            </form>
        </div>
    </nav>
    <div class="row text-center" style="background-color: #F6C6EA;margin: 1% 11%;padding: 2%;margin-top: 4%;">
        <div class="col">
            <h2 style="font-size: 4vw;">Do you want to buy a product but you dont trust the seller?</h2>
            <div style="padding: 0px 25%;">
                <p style="font-size: 1.5vw;justify-content: center;">This REVIEW ANALYZER, analyzes all the reviews left
                    behind by previous buyers and helps you make an educated decision. By using machine learning
                    technique Vader Sentimental analyzer, Reviewaholic can help you find the right reasons for buying
                    the right products.</p>
            </div>
        </div>
    </div>
    <div class="row text-center" style="margin-top: 35px;margin: 1% 11%;background-color: #F6C6EA;padding: 2%;">
        <div class="row">
            <h2 class=" text-center" style="font-size: 3vw;">Tutorial</h2>
        </div>
        <div class="col">
            <p class="category" style="font-size: 1.5vw;">1) Select the desired product</p>
            <img alt="Rounded Image" height="200px" width="170px" class="rounded-circle" src="/static/1.png">
        </div>
        <div class="col">
            <p class="category" style="font-size: 1.5vw;">2) Select and copy the url</p>
            <img alt="Rounded Image" height="200px" width="170px" class="rounded-circle" src="/static/2.png">
        </div>
        <div class="col">
            <p class="category" style="font-size: 1.5vw;">3) Enter the url and click 'Submit'</p>
            <img alt="Rounded Image" height="200px" width="170px" class="rounded-circle" src="/static/3.png">
        </div>
    </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-gtEjrD/SeCtmISkJkNUaaKMoLD0//ElJ19smozuHV6z3Iehds+3Ulb9Bn9Plx0x4"
        crossorigin="anonymous"></script>
</body>
</html>
```

- **process.html**

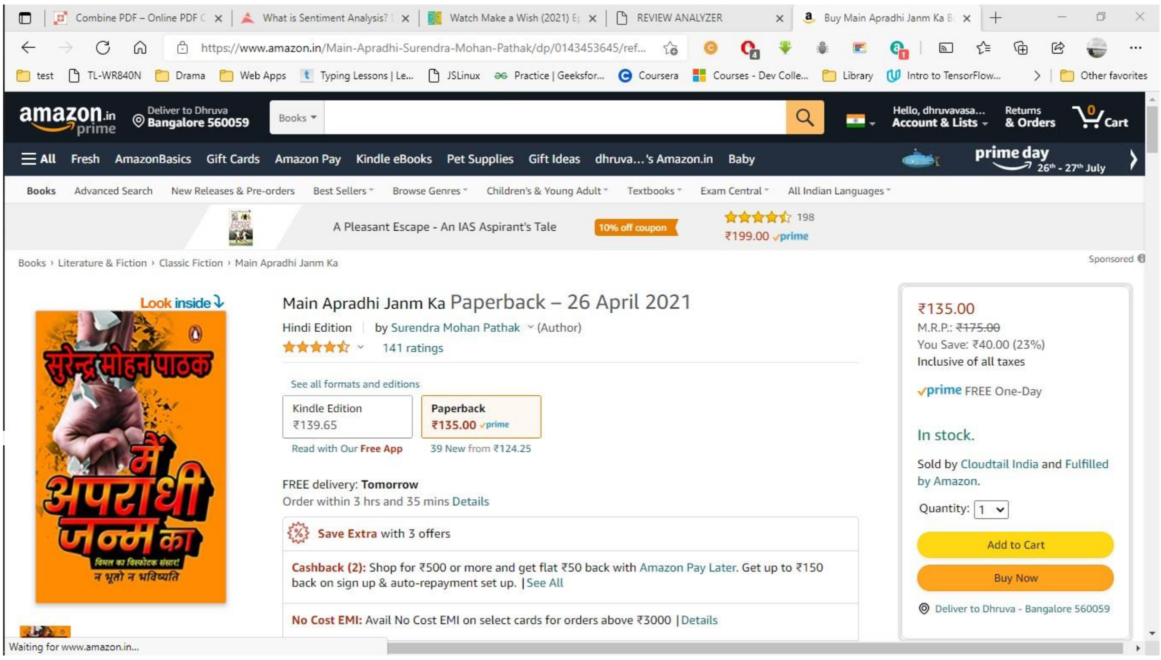
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title> {{ p_title }} </title>



  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x" crossorigin="anonymous">

</head>

<body style="background-color: #CDF0EA;">

    <nav class="navbar navbar-light fixed-top" style="background-color: #C490E4;">

        <div class="container-fluid">

            <a class="navbar-brand" href="http://127.0.0.1:5000/">

                <img class="rounded" src="{{ url_for('static',filename='logo.jpg') }}" alt="REVIEW ANALYZER" width="30"

                    height="24">

                REVIEW ANALYZER

            </a>

            <form class="d-flex" action="/process" method="POST">

                <input class="form-control me-2" type="search" placeholder="Amazon product link" aria-label="text" name="url">

                <button class="btn btn-outline-success" type="submit">Search</button>

            </form>

        </div>

    </nav>

    <div class="container">
```

```
<div class="row" style="background-color: #F6C6EA;margin: 1% 11%;padding: 2%;margin-top: 5%;">
    <div class="col">
        <div class="photo-container">
            <a href="https://www.amazon.in/dp/{{ asin_id }}">
                <img src="{{ p_image }}" alt="{{ p_title }}">
                <span>{{ p_title }}</span> </a>
            <p>{{ len_r }} Reviews & {{ verdict }}</p>
        </div>
    </div>
    <div class="col text-center">
        <div class="row">
            <div class="col">
                <h2>{{pos}}</h2>
                <p>Positive</p>
            </div>
            <div class="col text-center">
                <h2>{{neg}}</h2>
                <p>Negative</p>
            </div>
            <div class="col text-center">
                <h2>{{neutral}}</h2>
                <p>Neutral</p>
            </div>
        </div>
      </div>
    </div>
    <div class="row" style="background-color: #F6C6EA;margin: 1% 11%;padding:
2%;">
      <div class="col align-middle" style="margin: auto 0;">
        <h3 class="title">Thats a WordClud --></h3>
        <h5>Word Cloud is a data visualization technique used for representing text
data in which the
          size of each word indicates
          its frequency or importance. Significant textual data points can be
highlighted using a word
          cloud.
          Word clouds are widely used for analyzing data from social network
websites</h5>
      </div>
      <div class="col text-center">
        <img src="{{ url_for('static',filename='wordcloud.png') }}" height="300px"
width="300px"
          alt="Raised Image" class="rounded" />
      </div>
    </div>

    <div class="row text-center" style="background-color: #F6C6EA;margin: 1%
11%;padding: 2%;">
      <div class="col">
```

```
            <p style="font-size: large;"><u> 3 Random Reviews with Highest Polarity:</u></p>

            <div>

                <!-- Tab panes -->

                <table>

                    <thead>

                        <tr>

                            <th>Review</th>

                        </tr>

                    </thead>

                    <tbody>

                        {% for row in row_data %}

                        <tr>

                            <td>{{ row[0] }}</td>

                        </tr>

                        {% endfor %}

                    </tbody>

                </table>

            </div>

        </div>

        <div class="col">

            <p style="font-size: large;">3 Random Reviews with Lowest Polarity:</p>

            <div>

                <!-- Tab panes -->
```

```
                    <table>
                        <thead>
                            <tr>
                                <th>Review</th>
                            </tr>
                        </thead>
                        <tbody>
                            {% for row in row2_data %}
                            <tr>
                                <td>{{ row[0] }}</td>
                            </tr>
                            {% endfor %}
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

CHAPTER 5: SNAPSHOTS



A Pleasant Escape - An IAS Aspirant's Tale 10% off coupon 198 ₹199.00 prime

Sponsored

Books › Literature & Fiction › Classic Fiction › Main Apradhi Janm Ka

Main Apradhi Janm Ka Paperback – 26 April 2021

Hindi Edition | by Surendra Mohan Pathak (Author)

141 ratings

See all formats and editions

Kindle Edition ₹139.65 | Paperback ₹135.00 prime

Read with Our Free App | 39 New from ₹124.25

FREE delivery: Tomorrow
Order within 3 hrs and 35 mins Details

Save Extra with 3 offers

Cashback (2): Shop for ₹500 or more and get flat ₹50 back with Amazon Pay Later. Get up to ₹150 back on sign up & auto-repayment set up. | See All

No Cost EMI: Avail No Cost EMI on select cards for orders above ₹3000 | Details

₹135.00
M.R.P.: ~~₹175.00~~
You Save: ₹40.00 (23%)
Inclusive of all taxes

prime FREE One-Day

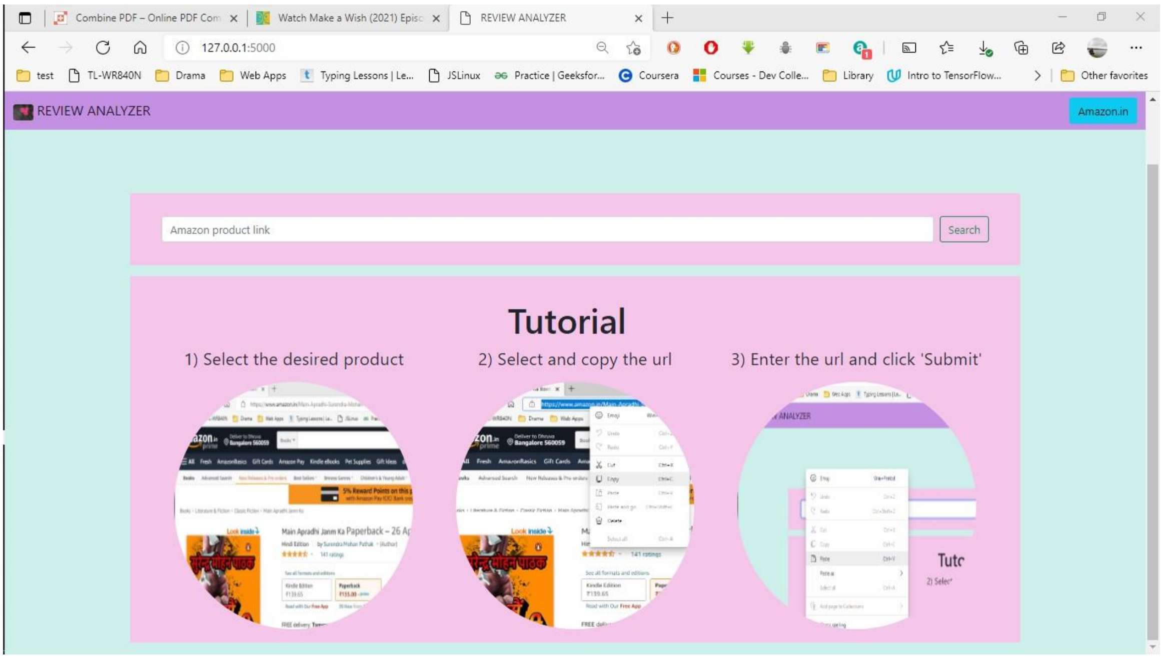
In stock.

Sold by Cloudtail India and Fulfilled by Amazon.

Quantity: 1

Deliver to Dhruva - Bangalore 560059

Fig 5.1.1 The Amazon page

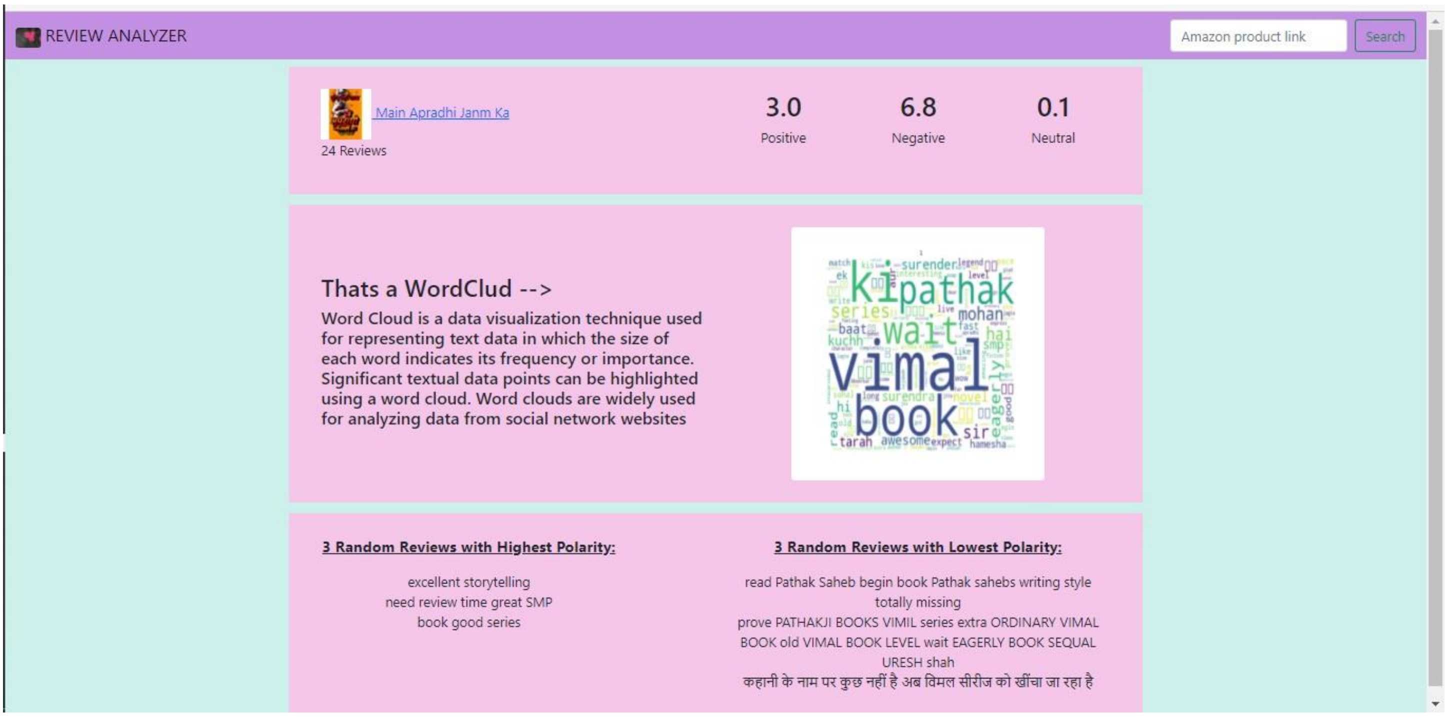


Amazon product link

Tutorial

1) Select the desired product
2) Select and copy the url
3) Enter the url and click 'Submit'

Fig 5.1.2 The Main Page



Main Apradhi Janm Ka

24 Reviews

3.0 Positive

6.8 Negative

0.1 Neutral

Thats a WordClud -->

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites

3 Random Reviews with Highest Polarity:

excellent storytelling

need review time great SMP

book good series

3 Random Reviews with Lowest Polarity:

read Pathak Saheb begin book Pathak sahebs writing style totally missing

prove PATHAKJI BOOKS VIMIL series extra ORDINARY VIMAL BOOK old VIMAL BOOK LEVEL wait EAGERLY BOOK SEQUAL URESH shah

कहानी के नाम पर कुछ नहीं है अब विमल सीरीज को खींचा जा रहा है

Fig 5.1.3 The Result Page

CHAPTER 6: CONCLUSION AND FUTURE WORK

Conclusion:

The task of sentiment analysis especially in the domain of web scrapping is still in developing stage and far from complete. We propose a model which we feel are worth exploring in the future and may result in the further improved performance. In this way, the effects of human confidence can be visualized in sentiment analysis

Future Work:

- If partnered with amazon.in you do multiple requests without using any round about method.
- If Vader dataset improved it can access more words
- If Vader algorithm is improved we get better result
- Implement continuous learning
- Improve loading time

CHAPTER 7: REFERENCES

- [1] https://librarycarpentry.org/lcwebscraping/reference
- [2] https://ieeexplore.ieee.org/document/7724353 (IEEE Paper)
- [3] List of Web Harvester, Data Scrapper, Web Scraping Software and Tools, and WebData Scraping. URL http://webdatascraping.com/webscraping-software/
- [4] S.C.M. de S Sirisuriya,2015, A Comparative Study on Web Scraping. Proceedings of 8th International Research Conference, KDU.
- [5] Web Data Extraction, Applications and Techniques: A SurveyEmilio Ferraraa, Pasquale DeMeob, Giacomo Fiumarac, Robert Baumgartnerd
- https://www.happiestminds.com/whitepapers/website-scraping.pdf,
- www.IJARIIT.com.
- https://arxiv.org/abs/1304.4520
- https://www.researchgate.net/publication/236203597_Sentiment_Analysis_A_Literature_Survey
- https://ieeexplore.ieee.org/document/8821809
- https://ieeexplore.ieee.org/document/9032456
- https://ieeexplore.ieee.org/document/8919363
- https://ieeexplore.ieee.org/document/8117827

Sentence

SENTIMENTAL ANALYSIS WITH WEB- SCRAPING ABSTRACT: E-commerce is getting used more and more these days to purchase products in an online store. A product review is usually used see if the product is worth buying or not. In that sense, the present work proposes an innovative solution by combing a web-scraping unit which is used to read the reviews from a website, Vader sentimental analyzer which is used to analyze the reviews of a product and return if the review is positive or negative or neutral, and a wordcloud which is an image containing positive words the positive words are selected by textblob module. This product used by selecting URL a product in amazon.in website opened in a browser, copy the URL first and open this product website and paste it in the input box and press enter, then this product processes the request and shows the score that the selected product has got and a wordcloud image The result of this project has shown success by show the result. ACKNOWLEDGMENTS We express our humble pranams to His Holiness Jagadguru Sri Sri Sri Shivarathreeshwara Mahaswamiji for showering his blessings on us to receive good education and have a successful career. The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of support from all ends during this project. So, we take this opportunity to express our gratitude to all whose guidance and encouragement helped us emerge successful. We are thankful for the resourceful guidance, timely assistance and graceful gesture of our guide Mrs. SAVITA S and Mrs. RANJITA S R, Assistant professors of Department Computer Science and Engineering, who has helped us in every aspect of our project work. We are also grateful to Dr. Naveen N C, Head of the Department, Computer Science and Engineering, for his unending support, guidance and encouragement in all our ventures. We express our sincere thanks to our beloved principal, Dr. Mrityunjaya V Latte for having supported us in all academic endeavours. Last but not the least, we would be immensely please to express our heartfelt thanks to all the teaching and non-teaching staff of the Department of CSE and our friend for their timely help, support and guidance. Dhruva V TABLE OF CONTENTS TITLE PAGE NO 1. INTRODUCTION 1.1 Introduction 6 1.2 Aims and Objectives 7 1.3 Problem Statement 7 1.4 Motivation 7 1.5 Literature Survey 8 2. SOFTWARE REQUIREMENTS 2.1 Existing System 10 2.2 Proposed System 10 2.3 System Requirements 10 3. DESIGN 3.1 System Design 11 4. IMPLEMENTATION 4.1 Implementation 12 4.2 Codes 14 5. SNAPSHOTS 27 6. CONCLUSION AND FUTURE WORK 28 7. REFERENCES 29 TABLE OF FIGURES TITLE PAGE NO 1. Fig 3.1.1 The overview of our project 11 2. Fig 4.1.1 The Front End 12 3. Fig 4.1.2 The Web Scraping Unit 12 4. Fig 4.1.3 The Cleaning Unit 13 5. Fig 4.1.4 The VADER UNIT 13 6. Fig 4.1.1 The EDA Unit 14 7. Fig 5.1.1 The Amazon page 27 8. Fig 5.1.2 The Main Page 27 9. Fig 5.1.3 The Result Page 28 CHAPTER 1: INTRODUCTION 1.1 Introduction Sentiment analysis also known as opinion mining in the field of natural language processing (NLP) that builds systems that tries to identify and extract the opinions within text. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall tonality of the document. In the recent years, the exponential increase in the internet usage and exchange of public opinions is driving the force behind the sentiment analysis today. The web is a huge repository of structured and unstructured data. The analysis of this data to extract latent public opinion and sentiment is a challenging task. Technology has turned into a fundamental piece of everybody's life. Social media technology is already used widely by the public to speak out once mind openly. This data can be leveraged to have a better understanding of the current state of decision making. Machine learning approach is used for analyzing sentiments from the text .by the sentiment analysis in the specific domain, it is possible to identify the effect of domain info in sentiment classification. Web scrapping (web harvesting, web data extraction) is a technique employed to extract the large amounts of data from the websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format. Web scrapping may access the world wide web directly using the hypertext transfer protocol, or through the web browser. Web scrapping, a web page involves fetching it and extracting from it. Fetching is the downloading of the page, once fetched then extraction can take place. The content of the page may be parsed, searched, reformatted, its data copied into a spreadsheet and so on. Web scraping is used for contact scraping, and as a component of applications used for web indexing, web mining and data mining, online price change monitoring and price comparison, product review scraping (to watch the competition), gathering real estate listings, weather data monitoring, website change detection, research, tracking online presence and reputation, web mashup and, web data integration. Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. However, most web pages are designed for human end-users and not for ease of automated use. As a result, specialized tools and software have been developed to facilitate the scraping of web pages. Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. 1.2 Aims and Objectives AIMS: 1) Our first aim is by using the web scrapping we generate the unstructured data from the amazon product pages. 2) To the scrapped data, we create a machine learning model and it is used for analyzing the sentiments of the cleaned data. 3) By doing so, with the help of the model we can classify the sentiment intensity (positive, negative or neutral) of the scrapped data. 4) At last the output will be integrated with the help of flask and bootstrap. (extra: bootstrap--&gt;it is the most popular html, CSS and JS library in the world) OBJECTIVES: 1) The main objective behind this project is to provide a platform which will enable users to check the credibility of a retailer/product by scanning reviews. 2) Instead of going through potentially hundreds of reviews, our platform offers a one-click result. 1.3 Problem Statement The Primary focus of this project is to find the sentimental scores of a product review by the customers. By doing this we can find the sentimental scores of amazon product in amazon.in website, we can use this score to see if this product has good scores think about buying it. If the product has good positive reviews we can say that the product has satisfied the customers and if it has high negative reviews then we can say that it has not satisfied the customers. By using this project result we can think of buying the product or not. It shows a word cloud with words from the reviews. It shows the reviews with highest and lowest polarity. 1.4 Motivation The main motivation behind the project is to understand the reviews submitted online by varying customers for a product. With knowing the product has positive or negative reviews we can have a clear understanding of the reviews submitted. 1.5 Literature Survey Vidhi singrodia (2019) [1 proposed Web scraping is a recognizable phrase which has expanded significance owing to the requirement of “free” data accumulated in PDF documents or web pages. Numerous professionals and researchers require the data for processing, analysis and extraction of significant consequences. Alternatively, people dealing with B2B use cases require the admittance of data from several sources for its integration into innovative applications which will offer supplementary values and novelty. Throughout this paper we have reviewed the various aspects of Web Scrapper. Anirban mitra and subratapaul throughout this paper reviewed the various aspects of Web Scraper. Starting with the tools and software for web scraping, the operating principle, strength and drawbacks and finally the applications of web scraping systems. There are numerous features which can be reflected while the usage of a web scraper. This may be generalized on the solicitations of scraping. Likewise, the imprecise authority of construction of a project on the basis of scraped data creates it challenging to differentiate projects which are dependent on scraping machineries. We can, nevertheless provide an overview on the maximum characteristic arenas the technology is used in. Through some of these arenas will provide small instances on the means by which such a project could provide. D. Deepa (2019) [2 proposed Natural Language Processing (NLP) is a study of computational treatment of human language in order to make it understandable for computers. It is used in the research fields like artificial intelligence, information engineering, statistics, sentiment analysis and linguistics. Sentiment analysis plays significant role in NLP which performs the series of operations computationally to identify and categorize the sentiment conveyed in segment of words. The proposed approach is to detect the polarity of words from twitter using feature extraction and dictionary-based methods. Raaji and Tamilarasj experimented for both feature engineering and dictionary-based techniques are trained and tested. The better results are obtained from the feature engineering. But the drawback of feature engineering is tedious and time consuming and all of the code wrote over for many hours cannot be applied to any other problem. These problems can be overcome in the dictionary-based approaches. In the future work, performance can be compared with more algorithms like Naïve Bayes, Linear SVM and Artificial Neural Network and optimization techniques can be performed for score adaptation to increase the accuracy in Dictionary based approach. Harshavadantalpada (2019) [3 suggests that lexical and semantic-based methods for sentiment prediction offer better accuracy than Deep Learning methods. When a large enough and evenly distributed training dataset is not available. We observed that domain-specific knowledge affects the prediction accuracy of sentiment, mainly when the target text contains more domain-specific words. Malka N Halgamuge observed that domain-specific knowledge affects the prediction accuracy of sentiment, mainly when the target text contains more domain-specific words. Accuracy of Deep Learning methods is dependent on the quality of the training dataset and the distribution of the classes within the dataset. Nguyentranquocvinh says social media produces a large amount of data which can be utilized for data- driven or information-driven decision making. Among the lexical based methods, VADER shows the highest accuracy as it considers the semantic factor when making the prediction. In our case, we observed that telemedicine has a high number of positive sentiments. It is still in its infancy and has not spread to a broader demographic. Shreya Upadhyay (2017) [4 proposes Massive volumes of data are generated by various users, entities, applications and disseminated online. This copious volume of big data is distributed across millions of websites and is available for various applications. Search engines do provide a simple mechanism to access this data. Accessing this data using search engines requires a user to spend time and resources to manually click and download. Clearly, such a manual approach is not scalable for a vast majority of real life applications at the enterprise and organization level. There exist a number of automated approaches to data extraction from the web. Shivansh Bhasin and Mahantesh K Pattanshetti [4 highlight the benefits of automatic data extraction tools and their role as a significant component in the development of knowledge- based systems. CHAPTER 2: SOFTWARE REQUIREMENTS 2.1 EXISTING SYSTEM: The existing system has many limitations: 1) Sentiment model is a separate unit that hasn’t been integrated with web scrapping 2) It is more complex to implement and its expensive. There is lack of credible user interface. 2.2 PROPOSED SYSTEM: 1) Integration of both modules (web scrapper and sentiment analyzer) into a single unit. 2) Developing a user interface using flask and bootstrap. 3) Providing better result on the scrapped Data by implementing VADER intensity analyzer. 4) VADER allows us to rate the reviews based on the emotions in the text. 5) We generate a word cloud, which is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. 2.3 System Requirements: - The Modules used in this project are: Devices : Local / Personal Computer. OS : Windows(Local Computer) OS Distro : Windows 10 (Local Computer) Storage : SD Card (8GB minimum) Battery : Power Bank / Li-ion(Optional if required) Application : python Browser : Google chrome or any other. CHAPTER 3: DESIGN 3.1 System Design: The user can interact with our product through the web page. It also contains a tutorial and a short explanation of how our product works. It was built using Flask & Bootstrap 4. Generally, text data contains a lot of noise either in the form of symbols or in the form of punctuations and stopwords (Stopwords are the words in any language which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as "The Who" or "Take That".). Therefore, it becomes necessary to clean the text, not just for making it more understandable but also for getting better insights Sentiment Model accepts the cleaned data and determines whether a piece of writing is positive, negative or neutral. The EDA (Exploratory Data Analysis) unit is a crucial part of any project because that's where you get to know more about the data. In this phase, you can reveal hidden patterns in the data and generate insights from it. Fig 3.1.1 The overview of our project CHAPTER 4: IMPLEMENTATION 4.1 Implementation: Front End: When it comes to the front-end deployment, we are using flask and bootstrap templates. Flask is a micro-framework that helps in building reliable, scalable, and maintainable web applications. It manages the requests and responses to the flask server. We are using bootstrap templates to dynamically create new views for our users. Web Scraping Unit: The Web scraping unit is responsible for extracting the required data from the webpages. We are mainly using 1 library; Beautiful soup is a python library for pulling data out of html and xml pages. The extracted data is then stored in a csv file. After the user enters the URL, it is sent to the web scraping unit, where the required information is extracted and saved in a csv file. Fig 4.1.1 The Front End Fig 4.1.2 The Web Scraping Unit Cleaning unit: In the cleaning unit we remove the unwanted data from the text (i.e. scrapped text data) then we convert the text (review column) to lowercase and remove integers or numerical in text. Punctuations, null values and extra (spaces as they don't make any sense) and all these things are done by importing regular expressions (import re) Stopwords are this is in, which does not add much value. so, we remove them to decrease the size of dataset this process is called normalizing text (with spacy) spacy is most versatile and widely used library in NLP Lemmatization is nothing but normalization of words which means reducing a word into its root form. Sentiment analysis using VADER: • VADER (Valence Aware Dictionary for sentiment Reasoning) is an Unsupervised model used that is adaptive to interpret emotional (positive/negative) and emotional (strength) text feelings. • VADER is focused on the lexicons of words related to sentiment. Each of the words in the lexicon is rated as to whether it is positive or negative and assigns scores to them. • It uses the polarity scores () method to get the sentiment metrics for a piece of text. • Vader is an easy-to-use and powerful, this package that is based on lexicons of sentiment-related words. Fig 4.1.3 The Cleaning Unit Fig 4.1.4 The VADER UNIT Exploratory Data Analysis (EDA): • It is the process of exploring data, generating insights, testing hypotheses, and revealing underlying hidden patterns in the data. • In the large resource of data, we pick the required and clean it for Exploratory Data Analysis. • we'll create a Document Term Matrix that we'll later use in our analysis to get the insights in data and with the help of wordcloud we display it in our project. • From textblob sentiment polarity we pick the top most emotions contained words in the data. 4.2 Codes: • app.py from flask import Flask, render_template, request import requests import seaborn as sns import numpy as np import os import os.path from os import path from bs4 import BeautifulSoup import pandas as pd Fig 4.1.1 The EDA Unit import matplotlib.pyplot as plt import re import string from amazon_product_review_scraper import amazon_product_review_scraper import spacy from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer from sklearn.feature_extraction.text import CountVectorizer from wordcloud import WordCloud from textwrap import wrap from textblob import TextBlob import wordcloud_gen app = Flask(__name__) @app.route('/') def index(): if path.exists('scraped_data.csv'): os.remove('scraped_data.csv') return render_template('index.html', product='Product') @app.route('/process', methods=['POST') def process(): url = request.form.get('url') if 'amazon.in' in url: product_asin = url[url.find('dp/')+3: url.find('dp/')+13 try: review_scraper = amazon_product_review_scraper(amazon_site="amazon.in", product_asin=product_asin) except: return '[removed]alert&#40;"Only works with amazon.in"&#41;;window.history.back();[removed]' reviews_df, p_title, p_image = review_scraper.scrape() reviews_df['rating' = reviews_df['rating'].str[:1].astype(int) with open('scraped_data.csv', 'w') as csv_file: reviews_df.to_csv('scraped_data.csv', index=False) else: return '[removed]alert&#40;"Error in Input"&#41;;window.history.back();[removed]' print(p_title, p_image) df = pd.read_csv('scraped_data.csv') no_of_reviews = len(df) df = df[['content', 'rating'] df['cleaned' = df['content'].apply(lambda x: re.sub('\w*\d\w*', '', x)) df['cleaned' = df['cleaned'].apply(lambda x: re.sub('[%s' % re.escape(string.punctuation), '', x)) # Remove Punctuations df['cleaned' = df['cleaned'].apply(lambda x: re.sub(' +', ' ', x)) # python -m spacy download en_core_web_sm nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner') df['lemmatized' = df['cleaned'].apply(lambda x: ' '.join([token.lemma_ for token in list(nlp(x)) if (token.is_stop == False))) df = df[['rating', 'lemmatized'] df_new = df.rename(columns={'lemmatized': 'content'}) df = df_new sent = SentimentIntensityAnalyzer() sentiment_dict = [for i in range(0, len(df)): sentiment_dict.append(sent.polarity_scores(df.iloc[i, 1)) positive = [neutral = [negative = [compound = [for item in sentiment_dict: positive.append(item['pos') neutral.append(item['neu') negative.append(item['neg') compound.append(item['compound') sentiment_df = pd.DataFrame(list(zip(positive, neutral, negative, compound)), columns=['Positive', 'Neutral', 'Negative', 'Compound') df['Positive' =

sentiment_df['Positive'] df['Negative' = sentiment_df['Negative'] df['Neutral' = sentiment_df['Neutral'] df['Compound' = sentiment_df['Compound'] print(df.columns) df_temp = df[['rating', 'content'] df_temp = df_temp.assign(new="1") df_grouped = df_temp[['new', 'content'].groupby(by='new').agg(lambda x: ' '.join(x)) cv = CountVectorizer(analyzer='word') data = cv.fit_transform(df_grouped['content') df_dtm = pd.DataFrame(data.toarray(), columns=cv.get_feature_names()) df_dtm.index = df_grouped.index def generate_wordcloud(data, title): wc = WordCloud(width=400, height=330, max_words=150, background_color='white'). generate_from_frequencies(data) plt.figure(figsize=(10, 8)) plt.imshow(wc, interpolation='bilinear') plt.axis("off") plt.title(' '.join(wrap(title, 60)), fontsize=13) # plt.show() if path.exists('Project/static/wordcloud.png'): os.remove('Project/static/wordcloud.png') # wc.to_file&#40;'wordcloud.png'&#41; # return plt,wc plt.savefig('Project/static/wordcloud.png', format='png', dpi=500) df_dtm = df_dtm.transpose() for index, product in enumerate(df_dtm.columns): generate_wordcloud(df_dtm[product, product) # wordcloud_gen.generate_wordcloud(df_dtm[product, product) highest_polarity = pd.DataFrame(columns=['content') lowest_polarity = pd.DataFrame(columns=['content') df['polarity' = df['content'].apply(lambda x: TextBlob(x).sentiment.polarity) for index, Review in enumerate(df.iloc[df['polarity'.sort_values(ascending=False)[:3].index]['content']): highest_polarity = highest_polarity.append({'content': str(Review)}, ignore_index=True) for index, Review in enumerate(df.iloc[df['polarity'.sort_values(ascending=True)[:3].index]['content']): lowest_polarity = lowest_polarity.append({'content': str(Review)}, ignore_index=True) if float(df['Positive'.mean() * 10) > 6: verdict = 'This product is highly recommended!!!' elif float(df['Negative'.mean() * 10) < 0 verdict = 'This product is not recommended!' verdict = 'This product is recommended' asin_id=product_asin, p_image=p_image, p_title=p_title, len_r=no_of_reviews, row_data=list(highest_polarity.values.tolist()), row2_data=list(lowest_polarity.values.tolist()), titles=highest_polarity.columns.values, total_reviews=len(df), pos=str(df[xss=removed neutral=str(df[t=ve xss=removed app.run(debug=True)> &lt;html lang="en"&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt; &lt;meta http-equiv="X-UA-Compatible" content="IE=edge"&gt; &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt; &lt;title&gt; REVIEW ANALYZER &lt;/title&gt; &lt;link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet"integrity="sha384+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxY bOOl7+AMvyTG2x" crossorigin="anonymous"&gt; &lt;/head&gt; &lt;body style="background-color: #CDF0EA;"&gt; REVIEW ANALYZER &lt;form class="d-flex" action="/process" method="POST"&gt; &lt;input class="form-control me-2" type="search" placeholder="Amazon product link" aria- label="text" name="url"&gt; &lt;button class="btn btn-outline-success" type="submit"&gt;Search&lt;/button&gt; &lt;/form&gt; Do you want to buy a product but you dont trust the seller? This REVIEW ANALYZER, analyzes all the reviews left behind by previous buyers and helps you make an educated decision. By using machine learning technique Vader Sentimental analyzer, Reviewaholic can help you find the right reasons for buying the right products. Tutorial 1) Select the desired product 2) Select and copy the url 3) Enter the url and click 'Submit' [removed][removed] &lt;/body&gt; &lt;/html&gt; • process.html &lt;!DOCTYPE html> &lt;html lang="en"&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt; &lt;meta http-equiv="X-UA-Compatible" content="IE=edge"&gt; &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt; &lt;title&gt; {{ p_title }} &lt;/title&gt; &lt;link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyT G2x" crossorigin="anonymous"&gt; &lt;/head&gt; &lt;body style="background-color: #CDF0EA;"&gt; REVIEW ANALYZER &lt;form class="d-flex" action="/process" method="POST"&gt; &lt;input class="form-control me-2" type="search" placeholder="Amazon product link" aria- label="text" name="url"&gt; &lt;button class="btn btn-outline-success" type="submit"&gt;Search&lt;/button&gt; &lt;/form&gt; {{ p_title }} {{ len_r }} Reviews & {{ verdict }} {{pos}} Positive {{neg}} Negative {{neutral}} Neutral Thats a WordClud --&gt; Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites 3 Random Reviews with Highest Polarity: &lt;!-- Tab panes --&gt; Review {% for row in row_data %} {{ row[0 }} {% endfor %} 3 Random Reviews with Lowest Polarity: &lt;!-- Tab panes --&gt; Review {% for row in row2_data %} {{ row[0 }} {% endfor %} &lt;/body&gt; &lt;/html&gt; CHAPTER 5: SNAPSHOTS Fig 5.1.1 The Amazon page Fig 5.1.2 The Main Page Fig 5.1.3 The Result Page CHAPTER 6: CONCLUSION AND FUTURE WORK Conclusion: The task of sentiment analysis especially in the domain of web scrapping is still in developing stage and far from complete. We propose a model which we feel are worth exploring in the future and may result in the further improved performance. In this way, the effects of human confidence can be visualized in sentiment analysis Future Work: • If partnered with amazon.in you do multiple requests without using any round about method. • If Vader dataset improved it can access more words • If Vader algorithm is improved we get better result • Implement continuous learning • Improve loading time CHAPTER 7: REFERENCES • [1 https://librarycarpentry.org/lcwebscraping/reference • [2 https://ieeexplore.ieee.org/document/7724353 (IEEE Paper) • [3 List of Web Harvester, Data Scrapper, Web Scraping Software and Tools, and WebData Scraping. URL http://webdatascraping.com/webscraping-software/ • [4 S.C.M. de S Sirisuriya,2015, A Comparative Study on Web Scraping. Proceedings of 8th International Research Conference, KDU. • [5 Web Data Extraction, Applications and Techniques: A SurveyEmilio Ferraraa, Pasquale DeMeob, Giacomo Fiumarac, Robert Baumgartnerd • https://www.happiestminds.com/whitepapers/website-scraping.pdf, • www.IJARIIT.com. • https://arxiv.org/abs/1304.4520 • https://www.researchgate.net/publication/236203597_Sentiment_Analysis_A_Literature_Survey •

https://ieeexplore.ieee.org/document/8821809 • https://ieeexplore.ieee.org/document/9032456 • https://ieeexplore.ieee.org/document/8919363 • https://ieeexplore.ieee.org/document/8117827

Report Title:	Plagiarism report
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/358586414dc289a4fb7f1520cf81d27e7a6bd1626415873
Report Generated Date:	16 July, 2021
Total Words:	3868
Total Characters:	27233
Keywords/Total Words Ratio:	0%
Excluded URL:	No
Unique:	**75%**
Matched:	**25%**

Sentence wise detail:

SENTIMENTAL ANALYSIS WITH WEB- SCRAPING ABSTRACT: E-commerce is getting used more and more these days to purchase products in an online store.

A product review is usually used see if the product is worth buying or not. In that sense, the present work proposes an innovative solution by combing a web-scraping unit which is used to read the (0)

reviews from a website, Vader sentimental analyzer which is used to analyze the reviews of a product and return if the review

is positive or negative or neutral, and a wordcloud which is an image containing positive words the positive words are selected by

textblob module.

This product used by selecting URL a product in amazon.

in website opened in a browser, copy the URL first and open this product website and paste it

in the input box and press enter, then this product processes the request and shows the score that the

selected product has got and a wordcloud image The result of this project has shown success by show the result. (1)

ACKNOWLEDGMENTS We express our humble pranams to His Holiness Jagadguru Sri Sri Sri Shivarathreeshwara

Mahaswamiji for showering his blessings on us to receive good education and have a successful career. (2)

Mahaswamiji for showering his blessings on us to receive good education and have a successful The completion of any project involves the efforts of many people. (3)

We have been lucky enough to have received a lot of support from all ends during this project.

So, we take this opportunity to express our gratitude to all whose guidance and encouragement helped us emerge successful.

We are thankful for the resourceful guidance, timely assistance and graceful gesture of our guide Mrs. SAVITA S and Mrs.

RANJITA S R, Assistant professors of Department Computer Science and Engineering, who has helped us in every aspect of our project work. We are also grateful to Dr. (4)

Naveen N C, Head of the Department, Computer Science and Engineering, for his unending support, guidance and encouragement in all our ventures. We express our sincere thanks to our beloved principal, Dr. (3)

Mrityunjaya V Latte for having supported us in all academic endeavours. Last but not the least, we would be immensely please to express our heartfelt thanks to all the (6)

teaching and non-teaching staff of the Department of CSE and our friend for their timely help, support and guidance.

Dhruva V TABLE OF CONTENTS TITLE PAGE NO 1. INTRODUCTION 1.1 Introduction 6 1. 2 Aims and Objectives 7 1. (7)

3 Problem Statement 7 1.4 Motivation 7 1.

5 Literature Survey 8 2. SOFTWARE REQUIREMENTS 2. (8)

5 Literature Survey 8 2. 1 Existing System 10 2. (9)

2 Proposed System 10 2.

3 System Requirements 10 3. DESIGN 3.

1 System Design 11 4. IMPLEMENTATION 4.

1 Implementation 12 4.2 Codes 14 5. SNAPSHOTS 27 6.

CONCLUSION AND FUTURE WORK 28 7.

REFERENCES 29 TABLE OF FIGURES TITLE PAGE NO 1. Fig 3.1.

1 The overview of our project 11 2. Fig 4.1.

1 The Front End 12 3. Fig 4.1.

2 The Web Scraping Unit 12 4. Fig 4.1.

3 The Cleaning Unit 13 5. Fig 4.1. (10)

4 The VADER UNIT 13 6. Fig 4.1.

1 The EDA Unit 14 7. Fig 5.1.

1 The Amazon page 27 8. Fig 5.1.

2 The Main Page 27 9. Fig 5.1.

3 The Result Page 28 CHAPTER 1: INTRODUCTION 1.

1 Introduction Sentiment analysis also known as opinion mining in the field of natural language

processing (NLP) that builds systems that tries to identify and extract the opinions within text.

Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall tonality of the document.

In the recent years, the exponential increase in the internet usage and exchange of public opinions is driving the force behind the sentiment analysis today. The web is a huge repository of structured and unstructured data. (11)

In the recent years, the exponential increase in the internet usage and exchange of public opinions is driving the force behind the sentiment analysis today. The analysis of this data to extract latent public opinion and sentiment is a challenging task. (11)

Technology has turned into a fundamental piece of everybodys life. Social media technology is already used widely by the public to speak out once mind openly. (13)

Technology has turned into a fundamental piece of everybodys life. This data can be leveraged to have a better understanding of the current state of decision making. (14)

Machine learning approach is used for analyzing sentiments from the text .

by the sentiment analysis in the specific domain, it is possible to identify the effect of domain info in sentiment classification.

Web scrapping (web harvesting, web data extraction) is a technique employed to extract the large amounts of data from the websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format. (15)

Web scrapping may access the world wide web directly using the hypertext transfer protocol, or through the web browser.

Web scrapping, a web page involves fetching it and extracting from it.

Fetching is the downloading of the page, once fetched then extraction can take place. The content of the page may be parsed, searched, reformatted, its data copied into a spreadsheet and so on. (16)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web mining and data mining, online price change monitoring and price comparison, product review scraping (to watch the competition), gathering (17)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web real estate listings, weather data monitoring, website change detection, research, tracking online presence and reputation, web mashup and, web (18)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web data integration. (19)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. (20)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web However, most web pages are designed for human end-users and not for ease of automated use. (21)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web As a result, specialized tools and software have been developed to facilitate the scraping of web pages. (17)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web Flask is a micro web framework written in Python. (23)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web It is classified as a micro framework because it does not require particular tools or libraries. (23)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. (25)

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web However, Flask supports extensions that can add application features as if they were implemented in Flask itself. (26)

Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. 1.

2 Aims and Objectives AIMS: 1) Our first aim is by using the web scrapping we generate the unstructured data from the amazon product pages.

2) To the scrapped data, we create a machine learning model and it is used for analyzing the sentiments of the cleaned data.

3) By doing so, with the help of the model we can classify the sentiment intensity (positive, negative or neutral) of the scrapped data.

4) At last the output will be integrated with the help of flask and bootstrap.

(extra: bootstrap--&gt;it is the most popular html, CSS and JS library in the world) OBJECTIVES: 1) The main objective behind

this project is to provide a platform which will enable users to check the credibility of a retailer/product by scanning reviews.

2) Instead of going through potentially hundreds of reviews, our platform offers a one-click result. 1.

3 Problem Statement The Primary focus of this project is to find the sentimental scores of a product review by the customers. By doing this we can find the sentimental scores of amazon product in amazon. (27)

in website, we can use this score to see if this product has good scores think about buying it.

If the product has good positive reviews we can say that the product has satisfied the customers

and if it has high negative reviews then we can say that it has not satisfied the customers.

By using this project result we can think of buying the product or not.

It shows a word cloud with words from the reviews.

It shows the reviews with highest and lowest polarity. 1. 4 Motivation The main motivation behind the project is to understand the reviews submitted online by varying customers for a product. (28)

With knowing the product has positive or negative reviews we can have a clear understanding of the reviews submitted. 1.

5 Literature Survey Vidhi singrodia (2019) [1 proposed Web scraping is a recognizable phrase which

has expanded significance owing to the requirement of “free” data accumulated in PDF documents or web pages. (29)

Numerous professionals and researchers require the data for processing, analysis and extraction of significant consequences.

Alternatively, people dealing with B2B use cases require the admittance of data from several

sources for its integration into innovative applications which will offer supplementary values and novelty.

Throughout this paper we have reviewed the various aspects of Web Scrapper.

Anirban mitra and subratapaul throughout this paper reviewed the various aspects of Web Scraper.

Starting with the tools and software for web scraping, the operating principle, strength and drawbacks and finally the applications of web scraping systems.

There are numerous features which can be reflected while the usage of a web scraper.

This may be generalized on the solicitations of scraping.

Likewise, the imprecise authority of construction of a project on the basis of

scraped data creates it challenging to differentiate projects which are dependent on scraping machineries.

We can, nevertheless provide an overview on the maximum characteristic arenas the technology is used in.

Through some of these arenas will provide small instances on the means by which such a project could provide. D. (30)

Deepa (2019) [2 proposed Natural Language Processing (NLP) is a study of computational treatment of human language in order to make it understandable for computers. It is used in the research fields like artificial intelligence, information engineering, statistics, sentiment analysis and linguistics. (31)

Deepa (2019) [2 proposed Natural Language Processing (NLP) is a study of computational treatment of human language in order to make it understandable for computers. Sentiment analysis plays significant role in NLP which performs the series of operations computationally to identify and categorize the sentiment conveyed in segment of (32)

Deepa (2019) [2 proposed Natural Language Processing (NLP) is a study of computational treatment of human language in order to make it understandable for computers. words. (33)

Deepa (2019) [2 proposed Natural Language Processing (NLP) is a study of computational treatment of human language in order to make it understandable for computers. The proposed approach is to detect the polarity of words from twitter using feature extraction and dictionary-based methods. (31)

Raaji and Tamilarasj experimented for both feature engineering and dictionary-based techniques are trained and tested.

The better results are obtained from the feature engineering.

But the drawback of feature engineering is tedious and time consuming and all of the code wrote over for many hours cannot be applied to any other problem.

These problems can be overcome in the dictionary-based approaches.

In the future work, performance can be compared with more algorithms like Naïve Bayes, Linear SVM and Artificial

Neural Network and optimization techniques can be performed for score adaptation to increase the accuracy in Dictionary based approach.

Harshavadantalpada (2019) [3 suggests that lexical and semantic-based methods for sentiment prediction offer better accuracy than Deep Learning methods.

When a large enough and evenly distributed training dataset is not available. We observed that domain-specific knowledge affects the prediction accuracy of sentiment, mainly when the target text contains more domain-specific words. (13)

Malka N Halgamuge observed that domain-specific knowledge affects the prediction accuracy of sentiment, mainly when the target text contains more domain-specific words.

Accuracy of Deep Learning methods is dependent on the quality of the training dataset and the distribution of the classes within the dataset.

Nguyentranquocvinh says social media produces a large amount of data which can be utilized for data- driven or information-driven decision making. Among the lexical based methods, VADER shows the highest accuracy as it considers the semantic factor when making the prediction. (36)

Nguyentranquocvinh says social media produces a large amount of data which can be utilized for data- driven or information-driven decision making. In our case, we observed that telemedicine has a high number of positive sentiments. (37)

Nguyentranquocvinh says social media produces a large amount of data which can be utilized for data- driven or information-driven decision making. It is still in its infancy and has not spread to a broader demographic. (13)

Shreya Upadhyay (2017) [4 proposes Massive volumes of data are generated by various users, entities, applications and disseminated online. This copious volume of big data is distributed across millions of websites and is available for various applications. (39)

Search engines do provide a simple mechanism to access this data. Accessing this data using search engines requires a user to spend time and resources to manually click and download. (40)

Search engines do provide a simple mechanism to access this data. Clearly, such a manual approach is not scalable for a vast majority of real life applications at the enterprise and organization level. (39)

Search engines do provide a simple mechanism to access this data. There exist a number of automated approaches to data extraction from the web. (39)

Shivansh Bhasin and Mahantesh K Pattanshetti [4 highlight the benefits of automatic data

extraction tools and their role as a significant component in the development of knowledge- based systems. (43)

extraction tools and their role as a significant component in the development of knowledge- CHAPTER 2: SOFTWARE REQUIREMENTS 2. (44)

1 EXISTING SYSTEM: The existing system has many limitations: 1) Sentiment model is a separate unit

that hasn’t been integrated with web scrapping 2) It is more complex to implement and its expensive. (45)

There is lack of credible user interface. 2.

2 PROPOSED SYSTEM: 1) Integration of both modules (web scrapper and sentiment analyzer) into a single unit.

2) Developing a user interface using flask and bootstrap.

3) Providing better result on the scrapped Data by implementing VADER intensity analyzer.

4) VADER allows us to rate the reviews based on the emotions in the text.

5) We generate a word cloud, which is a data visualization technique used for representing text data in which the size of each word indicates its frequency (46)

5) We generate a word cloud, which is a data visualization technique used or importance. 2. (47)

3 System Requirements: - The Modules used in this project are: Devices : Local / Personal Computer.

OS : Windows(Local Computer) OS Distro : Windows 10 (Local Computer) Storage : SD

Card (8GB minimum) Battery : Power Bank / Li-ion(Optional if required) Application : python Browser : Google chrome or any other. (48)

Card (8GB minimum) Battery : Power Bank / Li-ion(Optional if required) Application : python Browser CHAPTER 3: DESIGN 3. (49)

1 System Design: The user can interact with our product through the web page.

It also contains a tutorial and a short explanation of how our product works.

It was built using Flask &amp; Bootstrap 4.

Generally, text data contains a lot of noise either in the form of symbols or in the form

of punctuations and stopwords (Stopwords are the words in any language which does not add much meaning to a sentence. (50)

of punctuations and stopwords (Stopwords are the words in any language which does not add much meaning to a They can safely be ignored without sacrificing the meaning of the sentence. (51)

of punctuations and stopwords (Stopwords are the words in any language which does not add much meaning to a For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. (52)

of punctuations and stopwords (Stopwords are the words in any language which does not add much meaning to a In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as “The Who” or “Take That”.). (53)

Therefore, it becomes necessary to clean the text, not just for making it more understandable but also for getting

better insights Sentiment Model accepts the cleaned data and determines whether a piece of writing is positive, negative or neutral.

The EDA (Exploratory Data Analysis) unit is a crucial part of any project because that’s where you get to know more about the data.

In this phase, you can reveal hidden patterns in the data and generate insights from it. Fig 3.1. 1 The overview of our project CHAPTER 4: IMPLEMENTATION 4. (54)

1 Implementation: Front End: When it comes to the front-end deployment, we are using flask and bootstrap templates.

Flask is a micro-framework that helps in building reliable, scalable, and maintainable web applications.

It manages the requests and responses to the flask server. We are using bootstrap templates to dynamically create new views for our users. (55)

It manages the requests and responses to the flask server. Web Scraping Unit: The Web scraping unit is responsible for extracting the required data from the webpages. (56)

We are mainly using 1 library; Beautiful soup is a python library for pulling data out of html and xml pages.

The extracted data is then stored in a csv file.

After the user enters the URL, it is sent to the web scraping unit, where the required information is extracted and saved in a csv file. Fig 4.1.

1 The Front End Fig 4.1.

2 The Web Scraping Unit Cleaning unit: In the cleaning unit we remove the unwanted data from the text (i. e.

scrapped text data) then we convert the text (review column) to lowercase and remove integers or numerical in text.

Punctuations, null values and extra (spaces as they dont make any sense) and all these things are

done by importing regular expressions (import re) Stopwords are this is in, which does not add much value.

so, we remove them to decrease the size of dataset this process is called normalizing text (with spacy) spacy is most

versatile and widely used library in NLP Lemmatization is nothing but normalization of words which means reducing a word into its root form. (57)

Sentiment analysis using VADER: • VADER (Valence Aware Dictionary for sentiment Reasoning) is an

Unsupervised model used that is adaptive to interpret emotional (positive/negative) and emotional (strength) text feelings.

• VADER is focused on the lexicons of words related to sentiment.

Each of the words in the lexicon is rated as to whether it is positive or negative and assigns scores to them.

• It uses the polarity scores () method to get the sentiment metrics for a piece of text.

• Vader is an easy-to-use and powerful, this package that is based on lexicons of sentiment-related words. Fig 4.1.

3 The Cleaning Unit Fig 4.1.

4 The VADER UNIT Exploratory Data Analysis (EDA): • It is the process of exploring data, generating insights, testing hypotheses, and revealing underlying hidden patterns in (58)

4 The VADER UNIT Exploratory Data Analysis (EDA): • It is the process the data. (59)

• In the large resource of data, we pick the required and clean it for Exploratory Data Analysis.

• we'll create a Document Term Matrix that we'll later use in our analysis to get the insights in data and with the help of wordcloud we display it in our project.

• From textblob sentiment polarity we pick the top most emotions contained words in the data. 4.2 Codes: • app.

py from flask import Flask, render_template, request import requests import seaborn as sns import numpy as np import os import os.

path from os import path from bs4 import BeautifulSoup import pandas as pd Fig 4.1. 1 The EDA Unit import matplotlib. (60)

pyplot as plt import re import string from amazon_product_review_scraper import amazon_product_review_scraper import spacy from vaderSentiment. vaderSentiment import SentimentIntensityAnalyzer from sklearn. (61)

pyplot as plt import re import string from amazon_product_review_scraper import amazon_product_review_scraper import spacy from vaderSentiment. feature_extraction. (62)

text import CountVectorizer from wordcloud import WordCloud from textwrap import wrap from textblob import TextBlob import wordcloud_gen app = Flask(__name__) @app.

route(&#039;/&#039;) def index(): if path.

exists(&#039;scraped_data. csv&#039;): os.

remove(&#039;scraped_data.

csv&#039;) return render_template(&#039;index.

html&#039;, product=&#039;Product&#039;) @app.

route(&#039;/process&#039;, methods=[&#039;POST&#039;) def process(): url = request. form. get(&#039;url&#039;) if &#039;amazon. (63)

in&#039; in url: product_asin = url[url. find(&#039;dp/&#039;)+3: url. (64)

find(dp/&#039;)+13 try: review_scraper = amazon_product_review_scraper(amazon_site=amazon.

in&quot;, product_asin=product_asin) except: return &#039;[removed]alert(&quot;Only works with amazon. in&quot;);window. history.

back();[removed]&#039; reviews_df, p_title, p_image = review_scraper.

scrape() reviews_df[&#039;rating&#039; = reviews_df[&#039;rating&#039;]. str[:1].

astype(int) with open(&#039;scraped_data.

csv&#039;, &#039;w&#039;) as csv_file: reviews_df. to_csv(&#039;scraped_data. (65)

csv&#039;, index=False) else: return &#039;[removed]alert(&quot;Error in Input&quot;);window. history.

back();[removed]&#039; print(p_title, p_image) df = pd. read_csv(&#039;scraped_data. (65)

csv&#039;) no_of_reviews = len(df) df = df[[&#039;content&#039;, &#039;rating&#039;] df[&#039;cleaned&#039; = df[&#039;content&#039;]. apply(lambda x: re. (65)

csv&#039;) no_of_reviews = len(df) df = df[[&#039;content&#039;, &#039;rating&#039;] df[&#039;cleaned&#039; = df[&#039;content&#039;]. sub(&#039;\w*\d\w*&#039;, &#039;&#039;, x)) df[&#039;cleaned&#039; = df[&#039;cleaned&#039;]. (65)

apply(lambda x: re. sub(&#039;[%s&#039; % re. escape(string.

punctuation), &#039;&#039;, x)) # Remove Punctuations df[&#039;cleaned&#039; = df[&#039;cleaned&#039;]. apply(lambda x: re. (65)

sub(&#039; +&#039;, &#039; &#039;, x)) # python -m spacy download en_core_web_sm nlp = spacy.

load(&#039;en_core_web_sm&#039;, disable=[&#039;parser&#039;, &#039;ner&#039;) df[&#039;lemmatized&#039; = df[&#039;cleaned&#039;].

apply(lambda x: &#039; &#039;. join([token.

lemma_ for token in list(nlp(x)) if (token.

is_stop == False))) df = df[[&#039;rating&#039;, &#039;lemmatized&#039;] df_new = df.

rename(columns={&#039;lemmatized&#039;: &#039;content&#039;}) df = df_new sent = SentimentIntensityAnalyzer() sentiment_dict = [for i in range(0, len(df)): sentiment_dict. append(sent.

polarity_scores(df.

iloc[i, 1)) positive = [neutral = [negative = [compound = [for item in sentiment_dict: positive.

append(item[&#039;pos&#039;) neutral.

append(item[&#039;neu&#039;) negative.

append(item[&#039;neg&#039;) compound.

append(item[&#039;compound&#039;) sentiment_df = pd.

DataFrame(list(zip(positive, neutral, negative, compound)), columns=[&#039;Positive&#039;, &#039;Neutral&#039;, &#039;Negative&#039;, &#039;Compound&#039;) df[&#039;Positive&#039; = sentiment_df[&#039;Positive&#039;] df[&#039;Negative&#039; = sentiment_df[&#039;Negative&#039;] df[&#039;Neutral&#039;

= sentiment_df[&#039;Neutral&#039;] df[&#039;Compound&#039; = sentiment_df[&#039;Compound&#039;] print(df.

columns) df_temp = df[[&#039;rating&#039;, &#039;content&#039;] df_temp = df_temp.

assign(new=&quot;1&quot;) df_grouped = df_temp[[&#039;new&#039;, &#039;content&#039;]. groupby(by=&#039;new&#039;). agg(lambda x: &#039; &#039;.

join(x)) cv = CountVectorizer(analyzer=&#039;word&#039;) data = cv.

fit_transform(df_grouped[&#039;content&#039;) df_dtm = pd. DataFrame(data. toarray(), columns=cv. (65)

get_feature_names()) df_dtm.

index = df_grouped.

index def generate_wordcloud(data, title): wc = WordCloud(width=400, height=330, max_words=150, background_color=white&#039;).

generate_from_frequencies(data) plt.

figure(figsize=(10, 8)) plt.

imshow(wc, interpolation=&#039;bilinear&#039;) plt. axis(off&quot;) plt. title(&#039; &#039;.

join(wrap(title, 60)), fontsize=13) # plt. show() if path.

exists(&#039;Project/static/wordcloud. png&#039;): os.

remove(&#039;Project/static/wordcloud. png&#039;) # wc.

to_file(&#039;wordcloud. png&#039;) # return plt,wc plt. (65)

savefig(&#039;Project/static/wordcloud.

png&#039;, format=&#039;png&#039;, dpi=500) df_dtm = df_dtm.

transpose() for index, product in enumerate(df_dtm.

columns): generate_wordcloud(df_dtm[product, product) # wordcloud_gen.

generate_wordcloud(df_dtm[product, product) highest_polarity = pd.

DataFrame(columns=[&#039;content&#039;) lowest_polarity = pd.

DataFrame(columns=[&#039;content&#039;) df[&#039;polarity&#039; = df[&#039;content&#039;].

apply(lambda x: TextBlob(x). sentiment.

polarity) for index, Review in enumerate(df.

iloc[df[&#039;polarity&#039;.

sort_values(ascending=False)[:3].

index][&#039;content&#039;]): highest_polarity = highest_polarity.

append({&#039;content&#039;: str(Review)}, ignore_index=True) for index, Review in enumerate(df.

iloc[df[&#039;polarity&#039;.

sort_values(ascending=True)[:3].

index][&#039;content&#039;]): lowest_polarity = lowest_polarity.

append({&#039;content&#039;: str(Review)}, ignore_index=True) if float(df[&#039;Positive&#039;.

mean() * 10) &gt; 6: verdict = &#039;This product is highly recommended!!!

&#039; elif float(df[&#039;Negative&#039;.

mean() * 10) &lt; 0 verdict = &#039;This product is not recommended!

&#039; verdict = &#039;This product is recommended&#039; asin_id=product_asin, p_image=p_image, p_title=p_title, len_r=no_of_reviews, row_data=list(highest_polarity. values.

tolist()), row2_data=list(lowest_polarity. values.

tolist()), titles=highest_polarity. columns.

values, total_reviews=len(df), pos=str(df[xss=removed neutral=str(df[t=ve xss=removed app.

run(debug=True)&gt; &lt;html lang=&quot;en&quot;&gt; &lt;head&gt; &lt;meta charset=&quot;UTF-8&quot;&gt; &lt;meta http-equiv=&quot;X-UA-Compatible&quot; content=&quot;IE=edge&quot;&gt; &lt;meta name=&quot;viewport&quot; content=&quot;width=device-width, initial-scale=1.

0&quot;&gt; &lt;title&gt; REVIEW ANALYZER &lt;/title&gt; &lt;link href=&quot;https://cdn. jsdelivr.

net/npm/bootstrap@5.0.

1/dist/css/bootstrap. min.

css&quot; rel=&quot;stylesheet&quot;integrity=&quot;sha384+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x&quot; crossorigin=&quot;anonymous&quot;&gt; &lt;/head&gt; &lt;body style=&quot;background-color: #CDF0EA;&quot;&gt; REVIEW ANALYZER &lt;form class=&quot;d-flex&quot; action=&quot;/process&quot; method=&quot;POST&quot;&gt; &lt;input class=&quot;form-control me-2&quot; type=&quot;search&quot; placeholder=&quot;Amazon product link&quot;

aria- label=&quot;text&quot; name=&quot;url&quot;&gt; &lt;button class=&quot;btn btn-outline-success&quot; type=&quot;submit&quot;&gt;Search&lt;/button&gt; &lt;/form&gt; Do you want to buy a product but you dont trust the seller?

This REVIEW ANALYZER, analyzes all the reviews left behind by previous buyers and helps you make an educated decision.

By using machine learning technique Vader Sentimental analyzer, Reviewaholic can help you find the right reasons for buying the right products.

Tutorial 1) Select the desired product 2) Select and copy the url 3) Enter the url and click Submit&#039; [removed][removed] &lt;/body&gt; &lt;/html&gt; • process. html &lt;!

DOCTYPE html&gt; &lt;html lang=en&quot;&gt; &lt;head&gt; &lt;meta charset=&quot;UTF-8&quot;&gt; &lt;meta http-equiv=&quot;X-UA-Compatible&quot; content=&quot;IE=edge&quot;&gt; &lt;meta name=&quot;viewport&quot; content=&quot;width=device-width, initial-scale=1.

0&quot;&gt; &lt;title&gt; {{ p_title }} &lt;/title&gt; &lt;link href=&quot;https://cdn. jsdelivr.

net/npm/bootstrap@5.0.

1/dist/css/bootstrap. min.

css&quot; rel=&quot;stylesheet&quot; integrity=&quot;sha384+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyT G2x&quot; crossorigin=&quot;anonymous&quot;&gt; &lt;/head&gt; &lt;body style=&quot;background-color: #CDF0EA;&quot;&gt; REVIEW ANALYZER &lt;form class=&quot;d-flex&quot; action=&quot;/process&quot; method=&quot;POST&quot;&gt; &lt;input class=&quot;form-control me-2&quot; type=&quot;search&quot; placeholder=&quot;Amazon product link&quot; aria-

label=&quot;text&quot; name=&quot;url&quot;&gt; &lt;button class=&quot;btn btn-outline-success&quot; type=&quot;submit&quot;&gt;Search&lt;/button&gt; &lt;/form&gt; {{ p_title }} {{ len_r }} Reviews &amp; {{ verdict }} {{pos}} Positive {{neg}} Negative {{neutral}} Neutral

Thats a WordClud --&gt; Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. (65)

Thats a WordClud --&gt; Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates Significant textual data points can be highlighted using a word cloud. (46)

Thats a WordClud --&gt; Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates Word clouds are widely used for analyzing data from social network websites 3 Random Reviews with Highest Polarity: &lt;! (65)

-- Tab panes --&gt; Review {% for row in row_data %} {{ row[0 }} {% endfor %} 3 Random Reviews with Lowest

Polarity: &lt;!

-- Tab panes --&gt; Review {% for row in row2_data %} {{ row[0 }} {% endfor %} &lt;/body&gt; &lt;/html&gt; CHAPTER 5: SNAPSHOTS Fig 5.1.

1 The Amazon page Fig 5.1.

2 The Main Page Fig 5.1.

3 The Result Page CHAPTER 6: CONCLUSION AND FUTURE WORK Conclusion: The task of sentiment

analysis especially in the domain of web scrapping is still in developing stage and far from complete. (65)

We propose a model which we feel are worth exploring in the future and may result in the further improved performance.

In this way, the effects of human confidence can be visualized in sentiment analysis Future Work: • If partnered with amazon.

in you do multiple requests without using any round about method.

• If Vader dataset improved it can access more words • If Vader

algorithm is improved we get better result • Implement continuous learning • Improve loading

time CHAPTER 7: REFERENCES • [1 https://librarycarpentry.

org/lcwebscraping/reference • [2 https://ieeexplore. ieee.

org/document/7724353 (IEEE Paper) • [3 List of Web Harvester, Data Scrapper, Web Scraping Software and Tools, and WebData Scraping.

URL http://webdatascraping.

com/webscraping-software/ • [4 S. C. M. de S Sirisuriya,2015, A Comparative Study on Web Scraping. (65)

com/webscraping-software/ • [4 S. C. M. Proceedings of 8th International Research Conference, KDU. (65)

• [5 Web Data Extraction, Applications and Techniques: A SurveyEmilio Ferraraa, Pasquale DeMeob, Giacomo Fiumarac, Robert Baumgartnerd • https://www. happiestminds.

com/whitepapers/website-scraping. pdf, • www. IJARIIT. com. • https://arxiv. org/abs/1304.

4520 • https://www. researchgate.

net/publication/236203597_Sentiment_Analysis_A_Literature_Survey • https://ieeexplore. ieee. org/document/8821809 • https://ieeexplore. ieee. (65)

org/document/9032456 • https://ieeexplore. ieee.

org/document/8919363 • https://ieeexplore. ieee.

org/document/8117827

Match Urls:

0: https://github.com/eugenp/tutorials/blob/master/spring-mvc-java/src/test/java/com/baeldung/htmlunit/HtmlUnitWebScrapingLiveTest.java

1: https://www.merriam-webster.com/dictionary/result

2: https://www.merriam-webster.com/dictionary/career

3: https://www.scribd.com/document/448608637/MedicalReport

4: https://www.va.gov/covidtraining/docs/ONS_Leading_Psychoeducational_Groups_The_Nurse_Role.pdf

5: https://www.instagram.com/p/CBWh6mKHzVj/

6: https://www.borderscollege.ac.uk/downloads/business_continuity_plandecember2017.pdf

7: https://www.amazon.com/Software-Requirements-Wiegers-published-Microsoft/dp/B00E283G9S

8: https://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=7746&context=rtd

9: https://www.ncbi.nlm.nih.gov/books/NBK543528/figure/ch4.Fig1/?report=objectonly

10: http://www-scf.usc.edu/~ppremkum/portfolio/pdf/report.pdf

11: https://core.ac.uk/display/268882850

12: https://www.researchgate.net/profile/Malka-Halgamuge

13: https://www.red-gate.com/simple-talk/blogs/web-scrapping-with-python-using-beautifulsoap/

14: https://stackoverflow.com/questions/46882653/phps-json-encode-sometimes-lost-its-last-bracket-when-parsed-by-js?answertab=active

15: https://www.coursehero.com/file/98983348/web-scrapingdocx/

16: https://www.grepsr.com/blog/what-is-web-scraping/

17: https://www.safe.com/what-is/data-integration/

18: https://app.real.discount/offer/automate-web-scraping-using-python-scripts-and-spiders-6113

19: https://www.wakeupdata.com/blog/enriching-through-scraping-and-merging-data

20: https://medium.com/@ashutoshvarma683/flask-576c3f08b56a

21: https://towardsdatascience.com/spelling-rectification-app-using-textblob-pyspellchecker-cb3ad0504fc7

22: https://www.codersarts.com/flask-assignment-help

23: https://truesteamachievements.com/game/Sentimental-K/scores

24: https://www.reddit.com/r/Essays/comments/anz8dz/the_reason_behind_my_motivation/

25: https://docs.microsoft.com/en-us/aspnet/web-pages/

26: https://www.thefreedictionary.com/D

27: https://ieeexplore.ieee.org/document/9032456/

28: https://www.semanticscholar.org/paper/Sentiment-Analysis-using-Feature-Extraction-and-Deepa-Raaji/2954350d0ce300fa66fee33b5854b91e1135d6f5

29: https://www.dictionary.com/browse/words

30: https://www.quora.com/What-happens-in-the-Star-Wars-universe-that-shows-Anakin-Skywalker-Darth-Vader-being-very-powerful/answers/99018237

31: https://caserighted.com/telemedicine-case-analysis/

32: https://ieeexplore.ieee.org/document/8117827

33: https://in.linkedin.com/in/vishal-pant

34: https://www.nasa.gov/consortium/ModelBasedSystems/

35: https://www.sci.utah.edu/images/software/SCIRun/MagneticalBrainStimulationTutorial.pdf

36: https://www.thefreedictionary.com/expensive

37: https://www.geeksforgeeks.org/generating-word-cloud-python/

38: https://grammarist.com/usage/substantial-substantive/

39: https://www.usna.edu/HRO/_files/documents/Public%20-%20Docs/Training/Uncle%20Sams%20OPSEC.pdf

40: https://fdocuments.net/document/chapter-3-55845e35ab13b.html

41: https://www.grammar-monster.com/glossary/sentences.htm

42: https://medium.com/@saitejaponugoti/stop-words-in-nlp-5b248dadad47

43: https://www.kaggle.com/heeraldedhia/stop-words-in-28-languages

44: https://en.wikipedia.org/wiki/Stopwords

45: https://www.waterboards.ca.gov/rwqcb9/water_issues/programs/basin_plan/bio_objectives/doc/Final_Amendments_to_Chapter_4_of_the_Basin_Plan.pdf

46: https://stackoverflow.com/questions/21877437/dynamically-updating-the-templates-html-in-an-angular-directive/?lastactivity

47: https://www.alibaba.com/herbal-extracting-concentrator-unit-suppliers.html

48: https://www.merriam-webster.com/dictionary/form

49: https://www.xing.com/social/share/spi?url=https%3A%2Fcoursesmap.com%2Fudemy%2Ftesting-statistical-hypotheses-in-data-science-with-python-3%3Futm_source%3Dxing

50: https://data.hrsa.gov/data/about

51: https://github.com/vecuenca/matplotlib/tree/master/unit

52: https://stackoverflow.com/questions/56065837/is-the-a-way-of-getting-the-degree-of-positiveness-or-negativeness-when-using-lo

53: https://en.wikipedia.org/wiki/Feature_extraction

54: https://github.com/dowelldev/Url-Query-Items-Extension

55: https://www.yelp.com/user_details?userid=dp6url0p88f9pocVk6YyEA

56: https://neptune.ai/blog/web-scraping-and-knowledge-graphs-machine-learning

57: https://stackoverflow.com/questions/61399659/applying-lambda-regex-to-pandas-dataframe-and-getting-correct-result-back-but-su

58: https://nextdoor.com/for_sale_and_free/d3f61389-df54-4b9f-a1c7-c7492da4caa8/

59: https://stackoverflow.com/questions/64094175/get-memory-error-when-trying-toarray-todense-with-sklearns-countvectorizer

60: https://www.facebook.com/public/Muhammad-Png-Plt

61: https://www.lexico.com/definition/word_cloud

62: https://www.coursehero.com/file/p6v334m/SWOT-Analysis-Are-most-widely-used-basic-techniques-for-analyzing-firm-and/

63: https://ludwig.guru/s/far+from+complete

64: http://ir.kdu.ac.lk/bitstream/handle/345/1051/com-059.pdf?sequence=1&isAllowed=y

65: https://www.facebook.com/pushtechh/posts/137615128048197

Keywords Density

One Word	2 Words	3 Words
data 3.1%	web scraping 0.52%	web scraping unit 0.24%
time 2.05%	sentiment analysis 0.4%	ieeexplore ieee org 0.2%
sentiment 1.85%	df cleaned 0.24%	ieee org document 0.2%
word 1.81%	scraped data 0.24%	scraped data csv 0.2%
review 1.77%	product review 0.24%	positive neutral negative 0.12%

Plagiarism Report

By check-plagiarism.com