INTERNSHIP PROJECT REPORT

Submitted By:-

Sonal Sharma

Roll No:-2001506035

UNDER THE JOINT SUPERVISION OF

Mr. Mayank Arora (Manager)

Technophilia Solutions

Industrial Area, New Delhi

Submitted To:-

Mr.Somendra Tripathi

Internship Certificate

Technophilia Solutions

B - 92 Second Floor GT Karnal Road
Industrial Area Delhi 110033

Summer Internship Letter

Date:18 August 2023   KTPL-2023-T-ST-1575

To Whom So Ever It May Concern

This letter clarifies that **Ms. Sonal Sharma** from **Rama University** has completed her Summer Internship with our organization on **Data Science** in association of our technology partner **Technophilia Solutions** from **01-Jul-2023** to **12-Aug-2023**.

During the Internship span with us, she was actively and diligently involved in the projects and tasks assigned. Shedevelopedthe project **"Sentiment Analysis"**, which implicated the practical execution of the courseware.

Our organization thanks her and wish her all the best for her future.

Sincerely,

Mr.MayankArora

Manager – Human Resource

Technophilia Solutions

Address: 1009 IndraprakashBuilding, Barakhamba Road Connaught Place, New Delhi 110001

Email: hr@technophilia.in **Website:** www.technophilia.

Technophilia Solutions

Regd. Address: B - 92 Second Floor GT Karnal Road Industrial Area, Delhi 110033

Website: technophilia.in   **Email:** hello@technophilia.in

Acknowledgement

I would like to thank **Mr. Mayank Arora**, The Project Manager of my Summer Internship for allowing me to do an internship within the organization.

I also like to thank those people for helped me in doing my internship and put faith in me.

It was a pleasure for me to work with such wonderful people and an immense sense of gratitude toward them.

I thank all those who helped me in the successful completion of the internship.

Sonal Sharma

ABSTRACT

Sentiment analysis, a burgeoning field in natural language processing, focuses on discerning and categorizing sentiments expressed in textual data. This study presents an exploration of sentiment analysis using machine learning techniques within the context of a Flask-based web application. The primary goal was to develop an efficient system capable of classifying sentiments in text data as positive, negative, or neutral.

The project utilized a diverse dataset comprising user-generated reviews from various sources. Employing preprocessing techniques such as tokenization and feature extraction methods including Bag-of-Words and TF-IDF, multiple machine learning models were trained and evaluated for sentiment classification. The chosen model was integrated into a Flask web app, enabling real-time sentiment analysis input by users.

Table of contents

Introduction

Sentiment analysis, an important area in Natural Language Processing, is the process of automatically detecting affective states of text. Sentiment analysis is widely applied to voice-of-customer materials such as product reviews in online shopping websites like Amazon, movie reviews or social media. It can be just a basic task of classifying the polarity of a text as being positive/negative or it can go beyond polarity, looking at sentiment states etc.

Sentiment analysis refers to analyzing an opinion or feelings about something using data like text or images, regarding almost anything. Sentiment analysis helps companies in their decision-making process. For instance, if public sentiment towards a product is not so good, a company may try to modify the product or stop the production altogether in order to avoid any losses.There are many sources of public sentiment e.g. public interviews, opinion polls, surveys, etc. However, with more and more people joining social media platforms, websites like Facebook and Twitter can be parsed for public sentiment.

DOCUMENTATION

Documentation plays a crucial role in capturing the essence of the summer internship project, providing a comprehensive understanding of the development process and the resulting Sentiment Analysis using Machine learning, Flask framework, and Python.

1. Data Collection and Preparation:

- **Data Sources**: Gather text data from diverse sources such as social media, reviews, surveys, or news articles.
- **Data Cleaning**: Remove noise, irrelevant characters, HTML tags, punctuation, and other non-essential elements.
- **Normalization**: Standardize text by converting to lowercase, removing accents, and handling contractions or abbreviations.
- **Tokenization**: Segment text into words, phrases, or sentences for further analysis.

2. Flask Integration: Details on integrating the trained sentiment analysis model with Flask to create a web application.

3.Model Performance: Analysis of the model's performance metrics (accuracy, precision, recall, F1-score) and comparisons between different models or techniques used.

4.Size:

- **Number of Instances**: Specify the total number of records or instances in the dataset.
- **Text Length**: Describe the average or range of text lengths per instance (e.g., average number of words or characters per entry).

5.Format:

File Format: Indicate the format in which the dataset is stored (CSV, JSON, TXT, etc.).

Sentiment Labels: Describe the sentiment labels/categories used in the dataset (positive, negative, neutral, or other sentiment categories).

6.Web Application Integration: Integrate the trained model with a web framework like Flask or Django to create a user-friendly interface.

7.Real-Time Analysis: Implement functionalities allowing users to input text and receive immediate sentiment analysis results.

8.Feedback Incorporation: Continuously improve the model by incorporating user feedback and retraining on updated datasets.

SYSTEM ANALYSIS AND DESIGN

System Analysis and Design are crucial phases in the development lifecycle, serving as the foundation for the successful implementation of any software project. In the context of the summer internship project focused on developing Sentiment Analysis using Machine learning, Flask framework, and Python.The system analysis and design phases play a pivotal role in shaping the features, structure, and functionality of the system.

System Analysis

Requirement Analysis:-

- **Data Requirements:** Assess the quality, quantity, and variety of data needed for effective sentiment analysis.
- **Computational Resources:** Analyze the computing resources required for training models and running the web application.
- **Performance Metrics:** Define the criteria for evaluating the effectiveness of sentiment analysis (accuracy, precision, recall).
- **Scalability and Maintenance:** Consider the system's scalability and ongoing maintenance requirements for long-term usage.

System Design

1. Data Collection and Preparation:

- **Data Sources:** Identify sources (social media, reviews, etc.) to collect text

data.

- **Data Acquisition:** Gather and store the data in a suitable format (CSV, JSON, TXT etc.).
- **Preprocessing:** Clean the text data by removing noise, handling missing values, and normalizing text (lowercasing, removing punctuation, etc.).

2.Model Development and Training:

- **Model Selection:** Choose appropriate ML models (Naive Bayes, SVM, LSTM, etc.) for sentiment analysis.
- **Data Splitting:** Divide the dataset into training, validation, and test sets.
- **Model Training:** Train the selected model using the training set.
- **Hyperparameter Tuning:** Optimize model hyperparameters for better performance.

3.Security and Privacy:

- **Data Encryption:** Ensure secure handling of user data.
- **Access Controls:** Implement user authentication and authorization.
- **Compliance:** Adhere to data privacy laws and regulations.

4.Integration with Flask Web Application:

- **Frontend Design:** Create user-friendly interfaces for text input and result display.
- **Backend Structure:** Develop backend routes to handle user requests and model predictions.
- **Model Integration:** Incorporate the sentiment analysis model into the backend of the application.

5. Deployment and Monitoring:

Deployment Strategy: Choose cloud platforms or servers for hosting.

Monitoring Tools: Implement monitoring for system health, performance, and errors.

6. **Data Collection:**

- **Sources:** Identify sources like social media, reviews, or surveys.
- **Tokenization:** Break text into tokens (words, phrases).
- **APIs/Scraping:** Determine methods to gather textual data (APIs, web scraping, databases).
- **Real-time/Periodic:** Decide if data collection is real-time or periodic batches.

Objective of the Project

Scrapping product reviews on various application featuring various products

Analyze and categorize review data.

Analyze sentiment on dataset from document level (review level).>

Categorization or classification of opinion sentiment into-

i . Positive   ii. Negative   iii. Neutral

Implementation

Data Collection and Preprocessing:

- **NLTK (Natural Language Toolkit):** Python library offering tools for tokenization, stemming, lemmatization, and other text preprocessing tasks.
- **Scrapy:** A web crawling and scraping framework used to extract data from websites.
- **Tweepy:** Python library for accessing the Twitter API to collect tweets for analysis.
- **Pandas:** Data manipulation library useful for handling and cleaning datasets.

Feature Extraction and Text Representation:

- **Scikit-learn:** Provides tools for creating Bag-of-Words, TF-IDF, and other feature extraction techniques.
- **Gensim:** Library for topic modeling and creating Word2Vec embeddings.
- **SpaCy:** An NLP library that can be used for word tokenization, part-of-speech tagging, and dependency parsing.

Web Application and Deployment:

- **Flask / Django:** Web frameworks used for building web applications where users can input text for sentiment analysis.
- **FastAPI:** A modern, fast (high-performance), web framework for building APIs.

Visualization and Reporting:

- **Matplotlib / Seaborn:** Libraries for data visualization to represent analysis results.
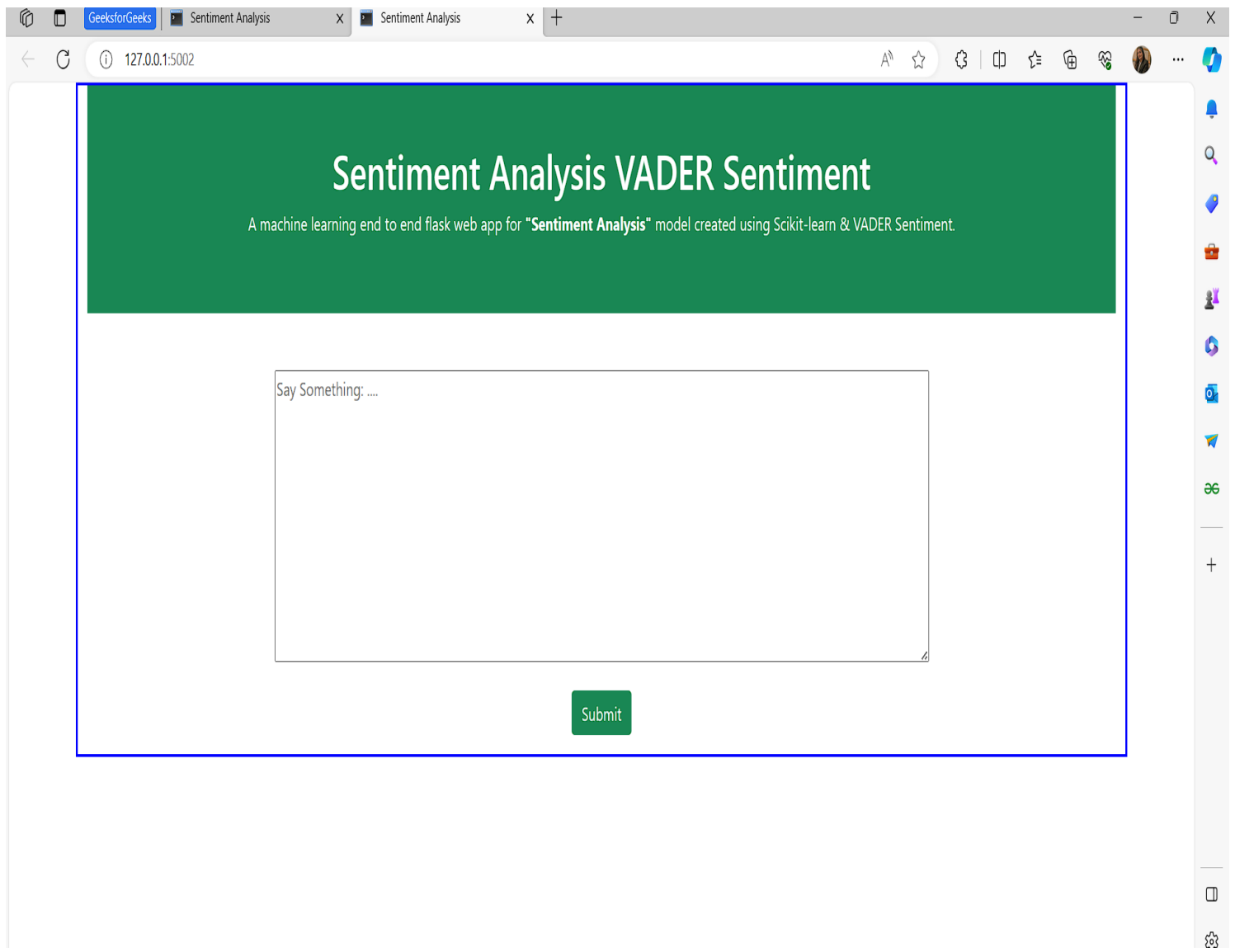- **Plotly:** Interactive visualization library to create appealing dashboards or charts.

VADER (Valence Aware Dictionary and sEntiment Reasoner): Lexicon and rule-based sentiment analysis tool, useful for quick analysis or baseline models.
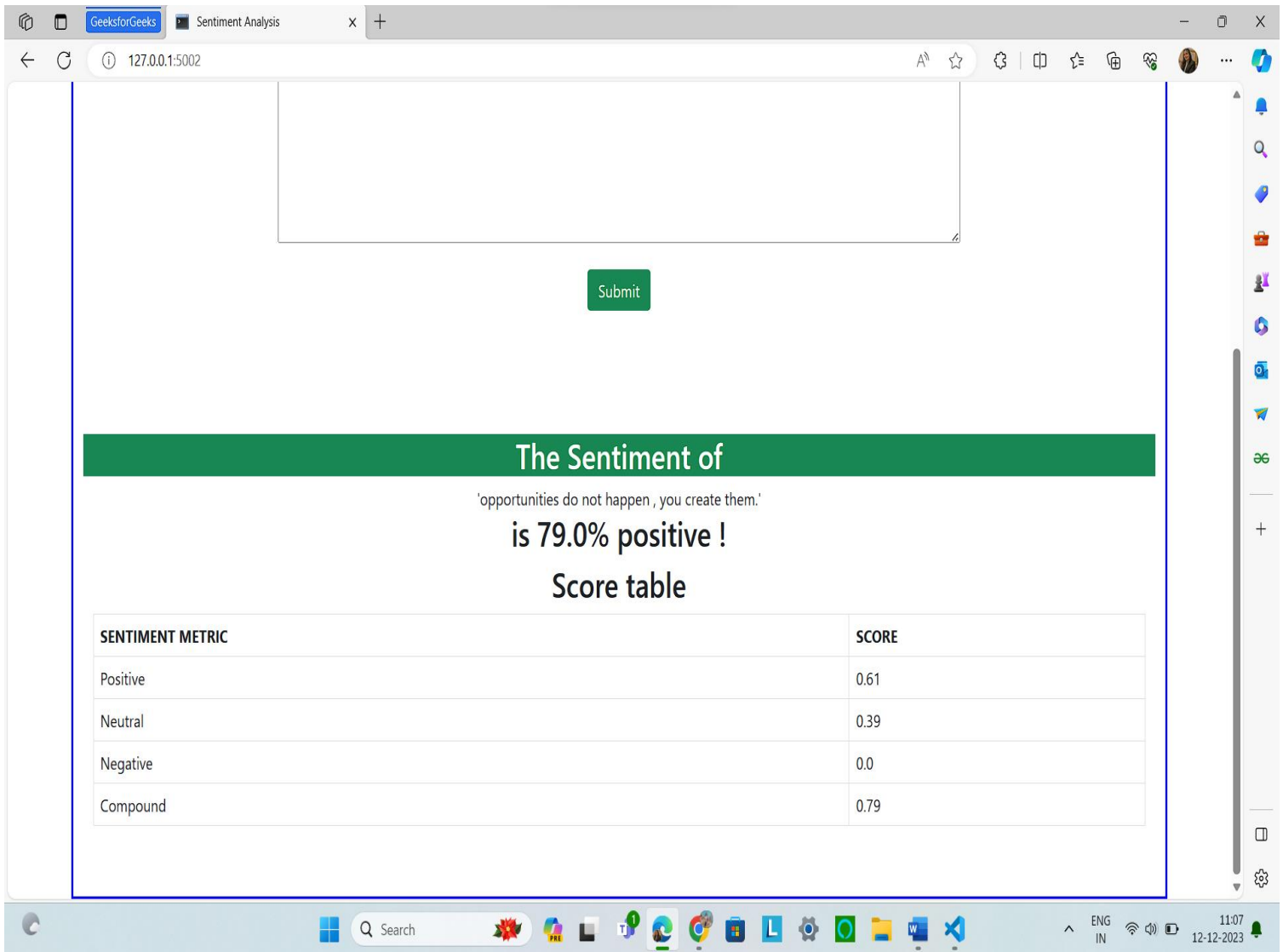
Stanford NLP: Offers a suite of NLP tools, including sentiment analysis, although it requires more advanced setup compared to Python libraries.

Result and Discussion

In a sentiment analysis Flask app, after analyzing the text, you'd typically return the result or prediction. This could be the sentiment itself (positive, negative, neutral), a sentiment score, or a classification label (like 'positive', 'negative', or 'neutral'). The result will be what the model predicts based on the input text's sentiment.

Submit

The Sentiment of

'opportunities do not happen , you create them.'

is 79.0% positive !

Score table

SENTIMENT METRIC	SCORE
Positive	0.61
Neutral	0.39
Negative	0.0
Compound	0.79

Conclusion

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