## 

## 

## **Innovation Project Report**



## <Innovation Sector> | <Innovation Title>

Team No:1

Mentor Name: Dr.Sudharson

Student Name: Sanjay S & Roll No: 23BAD097

Student Name: Sanjay Vasan & Roll No: 23BAD098

Student Name: Sanjay Vishwa S & Roll No: 23BAD099

Student Name: Sanjeev Dev SR & Roll No: 23BAD100

Student Name: Santhiya C & Roll No: 23BAD101

|  |  |  |
| --- | --- | --- |
| **A picture containing text, metalware, gear  Description automatically generated** |  | **A picture containing logo  Description automatically generated** |

**BONAFIDE CERTIFICATE**

Kumaraguru College of Technology, Coimbatore-641049

(Autonomous)

Affiliated to Anna University, Chennai

This is to certify that the project report, entitled **“Sentimental analysis of social media posts”** is a bonafide work of Team 1**,** submitted in partial fulfilment of the requirements for the **Engineering /Innovation/ Design/ Ideation Sprint of Innovation practicum**,( Engineering Clinic -Project Based Learning Framework) done during the EVEN Semester - **Feb 2024-June 2024** of Ist year B.Tech. Degree Programme offered by Kumaraguru College of Technology (Autonomous), Coimbatore- 641049 - Affiliated to Anna University, Chennai.

**FACULTY MENTOR(S)**

**DECLARATION**

We, Sanjay S, Sanjay Vasan, Sanjay Vishwa, Sanjeev Dev, Santhiya hereby declare that the project report, entitled **“Sentimental analysis of social media posts”,** submitted to Kumaraguru College of Technology in partial fulfilment of the requirements for the Engineering / Innovation / Design / Ideation Sprint of Innovation practicum,( Engineering Clinic -Project Based Learning Framework) done during the EVEN Semester(-**Feb 2024-June 2024**) of Ist year B.Tech. Degree Programme under the guidance of Dr.Sudharson Kumaraguru College of Technology (Autonomous), Coimbatore- 641049 - Affiliated to Anna University, Chennai.

**Signature of the Candidates**

## 

## #1 Problem Statement & Significance

|  |
| --- |
| **In the fast-paced digital landscape of today, where social media has become a constant companion, the content we encounter daily significantly shapes our mood. Negative posts, in particular, can cast a shadow on our emotions. To tackle this, we've crafted an ingenious algorithm that delves into post taglines, swiftly identifying the negative ones and effortlessly flipping them into positive vibes. It's like having a mood-lifting companion in the online realm. This isn't just tech; it's a digital remedy for the daily mood swings induced by social media. With our algorithm, the online experience becomes a more uplifting and positive journey, providing a real-time mood boost in the ever-evolving world of social media interactions.** |

## #2 Target User and Use-Case

|  |
| --- |
| **Our primary target users are individuals of all age groups who actively engage in social media platforms. This includes children, teens, and adults who experience varying emotional responses to the content they encounter online.**  **Our algorithm is designed to enhance the emotional well-being of social media users by addressing the negative impact of posts on their mood. The use case scenario involves individuals who regularly navigate through social media platforms, encountering posts that may evoke negative emotions. Our algorithm comes into play by analyzing the taglines of these posts and, when necessary, automatically transforming negative sentiments into positive ones. This use case aims to provide a seamless and proactive solution for users to enjoy a more uplifting and positive online experience, promoting better mental health in the context of social media interactions.** |

## #3 Expected outcomes or gains

|  |
| --- |
| **Our algorithm is set to revolutionize your social media game. Imagine scrolling through your feed without those buzzkill posts dragging you down – that's our tech in action. It scans for the mood-killers and magically flips them into something positive. Your online vibe just got a major upgrade – no more letting negativity mess with your chill. It's like having your personal positivity filter on social media, making the whole experience way more enjoyable. Our goal is simple: keep the good vibes rolling and make your online hangout a positive place.** |

## 

## #4 Usability & Deployment Constraints

|  |
| --- |
| **Used for analysis of taglines in a social media post.**  **During the development phase, we encountered several constraints that shaped our approach and design decisions. Limited access to comprehensive training datasets posed challenges in model refinement, necessitating innovative techniques for sentiment classification. Despite these challenges, iterative development methodologies and collaborative problem-solving enabled us to overcome obstacles and deliver a robust sentiment analysis pretotype tailored to user needs.** |

## #5 Solution Concept

|  |
| --- |
| **We've crafted a pretotype showcasing our algorithm's potential, offering a hands-on experience for developers to understand its functionality and potential impact. This pretotype lays the groundwork for a user-friendly app that seamlessly integrates into various social media platforms. Developers benefit from detailed analysis tools, streamlining the refining process for enhanced algorithm performance. For consumers, the app offers an intuitive interface, empowering them to easily customize sentiment transformation based on personal preferences. The real-time analysis ensures an immediate positive shift in mood, providing users with a novel and uplifting social media experience. We've prioritized transparent operation, allowing users to effortlessly navigate and control sentiment transformation levels. Our app's adaptability ensures continuous mood enhancement as users interact with evolving online content, fostering a positive digital space. Ongoing monitoring and user feedback mechanisms guarantee a responsive approach to addressing user needs and refining the app for optimal usability.** |

## #6 Pretotyping

|  |
| --- |
| **A machine learning model for sentimental analysis:**  **import tkinter as tk**  **from tkinter import filedialog, messagebox, ttk**  **import pandas as pd**  **import numpy as np**  **import matplotlib.pyplot as plt**  **import seaborn as sns**  **import re**  **import string**  **import nltk**  **from wordcloud import WordCloud**  **from sklearn.feature\_extraction.text import CountVectorizer**  **from sklearn.model\_selection import train\_test\_split**  **from sklearn.linear\_model import LogisticRegression**  **from sklearn.metrics import f1\_score, accuracy\_score**  **# Function to preprocess data and generate sentiment analysis**  **def preprocess\_and\_analyze():**  **try:**  **# Read data**  **file\_path = file\_path\_entry.get()**  **df = pd.read\_csv(file\_path)**  **# Data preprocessing**  **df['clean\_tweet'] = np.vectorize(remove\_pattern)(df['tweet'], "@[\w]\*")**  **df['clean\_tweet'] = df['clean\_tweet'].str.replace("[^a-zA-Z#]", " ")**  **df['clean\_tweet'] = df['clean\_tweet'].apply(lambda x: " ".join([w for w in x.split() if len(w) > 3]))**  **tokenised\_tweet = df['clean\_tweet'].apply(lambda x: x.split())**  **tokenised\_tweet = tokenised\_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])**  **for i in range(len(tokenised\_tweet)):**  **tokenised\_tweet[i] = " ".join(tokenised\_tweet[i])**  **df['clean\_tweet'] = tokenised\_tweet**  **# Plot selected type of graph**  **selected\_plot = plot\_type\_var.get()**  **if selected\_plot == "Word Cloud":**  **plot\_word\_cloud(df)**  **elif selected\_plot == "Bar Plot":**  **plot\_bar(df)**  **elif selected\_plot == "Pie Chart":**  **plot\_pie(df)**  **elif selected\_plot == "Scatter Plot":**  **plot\_scatter(df)**  **elif selected\_plot == "Most Used Word":**  **plot\_most\_used\_word(df)**  **# Display statistics**  **pos\_count = df[df['label'] == 0].shape[0]**  **neg\_count = df[df['label'] == 1].shape[0]**  **stat\_text = f"Total Tweets: {pos\_count+neg\_count}\nPositive Tweets: {pos\_count}\nNegative Tweets: {neg\_count}"**  **statistics\_label.config(text=stat\_text)**  **except Exception as e:**  **messagebox.showerror("Error", f"An error occurred: {str(e)}")**  **# Function to plot bar plot**  **def plot\_bar(df):**  **plt.figure(figsize=(8, 6))**  **df['label'].value\_counts().plot(kind='bar', color=['green', 'red'])**  **plt.title('Bar Plot of Positive and Negative Tweets')**  **plt.xlabel('Sentiment')**  **plt.ylabel('Count')**  **plt.legend()**  **plt.xticks([0, 1], ['Positive', 'Negative'], rotation=0)**  **plt.show()**  **# Function to plot pie chart**  **def plot\_pie(df):**  **plt.figure(figsize=(8, 6))**  **df['label'].value\_counts().plot(kind='pie', autopct='%1.1f%%', colors=['green', 'red'])**  **plt.title('Pie Chart of Positive and Negative Tweets')**  **plt.ylabel('')**  **plt.legend()**  **plt.show()**  **# Function to plot scatter plot**  **def plot\_scatter(df):**  **# Generate random data for scatter plot**  **np.random.seed(0)**  **x = np.random.rand(100)**  **y = np.random.rand(100)**  **colors = np.where(df['label'] == 0, 'green', 'red')**  **plt.figure(figsize=(8, 6))**  **plt.scatter(x, y, c=colors, alpha=0.5)**  **plt.title('Scatter Plot of Positive and Negative Tweets')**  **plt.xlabel('X-axis')**  **plt.ylabel('Y-axis')**  **plt.show()**  **# Function to plot word cloud**  **def plot\_word\_cloud(df):**  **# Wordcloud for positive tweets**  **positive\_tweets = df[df['label'] == 0]['clean\_tweet']**  **positive\_words = " ".join([tweet for tweet in positive\_tweets])**  **positive\_wordcloud = WordCloud(width=800, height=500, random\_state=42, max\_font\_size=100).generate(positive\_words)**    **# Wordcloud for negative tweets**  **negative\_tweets = df[df['label'] == 1]['clean\_tweet']**  **negative\_words = " ".join([tweet for tweet in negative\_tweets])**  **negative\_wordcloud = WordCloud(width=800, height=500, random\_state=42, max\_font\_size=100).generate(negative\_words)**  **# Plot word clouds**  **plt.figure(figsize=(15, 8))**  **plt.subplot(1, 2, 1)**  **plt.imshow(positive\_wordcloud, interpolation='bilinear')**  **plt.title('Word Cloud for Positive Tweets')**  **plt.axis('off')**  **plt.subplot(1, 2, 2)**  **plt.imshow(negative\_wordcloud, interpolation='bilinear')**  **plt.title('Word Cloud for Negative Tweets')**  **plt.axis('off')**  **plt.show()**  **# Function to plot the most used word**  **def plot\_most\_used\_word(df):**  **all\_words = " ".join([tweet for tweet in df['clean\_tweet']])**  **wordcloud = WordCloud(width=800, height=500, random\_state=42, max\_font\_size=100).generate(all\_words)**    **plt.figure(figsize=(10, 6))**  **plt.imshow(wordcloud, interpolation='bilinear')**  **plt.axis('off')**  **plt.title('Most Used Tweet Word')**  **plt.show()**  **# Function to remove Twitter handles**  **def remove\_pattern(input\_txt, pattern):**  **r = re.findall(pattern, input\_txt)**  **for word in r:**  **input\_txt = re.sub(word, " ", input\_txt)**  **return input\_txt**  **# Initialize NLTK**  **nltk.download('punkt')**  **nltk.download('stopwords')**  **# Initialize Stemmer**  **stemmer = nltk.stem.PorterStemmer()**  **# Set up GUI**  **root = tk.Tk()**  **root.title("Twitter Sentiment Analysis")**  **root.geometry("800x600")**  **# Header Label**  **header\_label = tk.Label(root, text="Twitter Sentiment Analysis", font=("Helvetica", 20))**  **header\_label.pack(pady=10)**  **# File Path Entry**  **file\_path\_label = tk.Label(root, text="Enter File Path:")**  **file\_path\_label.pack()**  **file\_path\_entry = tk.Entry(root, width=50)**  **file\_path\_entry.pack()**  **# Plot Type Label**  **plot\_type\_label = tk.Label(root, text="Select Plot Type:")**  **plot\_type\_label.pack()**  **# Plot Type Variable**  **plot\_type\_var = tk.StringVar(root)**  **plot\_type\_var.set("Bar Plot")**  **# Plot Type Dropdown Menu**  **plot\_type\_menu = tk.OptionMenu(root, plot\_type\_var, "Word Cloud", "Bar Plot", "Pie Chart", "Scatter Plot", "Most Used Word")**  **plot\_type\_menu.pack()**  **# Analyze Button**  **analyze\_button = tk.Button(root, text="Analyze", command=preprocess\_and\_analyze, bg="#4CAF50", fg="white",**  **font=("Helvetica", 14))**  **analyze\_button.pack(pady=10)**  **# Instructions Label**  **instructions\_label = tk.Label(root, text="Analysis Results will be displayed below:", font=("Helvetica", 12))**  **instructions\_label.pack(pady=10)**  **# Statistics Label**  **statistics\_label = tk.Label(root, text="", font=("Helvetica", 12))**  **statistics\_label.pack()**  **# Separator**  **separator = ttk.Separator(root, orient='horizontal')**  **separator.pack(fill='x', padx=20, pady=10)**  **# Footer Label**  **footer\_label = tk.Label(root, text="Developed by Team 1 (AI&DS)", font=("Helvetica", 10))**  **footer\_label.pack(side='bottom', pady=10)**  **# Run the GUI**  **root.mainloop()** |