Department of

CSE (Artificial Intelligence & Machine Learning)
MSR Nagar, MSRIT Post, Bengaluru-560 054

A Machine Learning Project Report
on

"Determining sentiment specific to different aspects of product or service within a single review. For example separating opinions about a product's design from its usability."

Submitted in partial fulfillment of the requirements for the V Semester, CSE (AI & ML)

Submitted by:

- MONICA D - 1MS23CI404
- MANASAVI - 1MS22CI038
- ANJALI C - 1MS22CI010
- TEJASWINI - 1MS22CI055
- PRANJAL - 1MS22CI048

Under the Guidance of:
Dr. A N Ramya Shree
Associate Professor
Dept. of CSE (AI & ML)
Ramaiah Institute of Technology

2024-2025

Certificate

Certified that the mini-project work entitled " Determining sentiment specific to different aspects of product or service within a single review. " has been successfully carried out by **"Monica D"** bearing USN **"1MS23CI404 "**, **"Manasavi"** bearing USN **"1MS22CI038 "**, **"Anjali C"** bearing USN **"1MS22CI010 "**, **"Tejaswini"** bearing USN **"1MS22CI055"** and **"Pranjal"** bearing USN **"1MS22CI048 "**, bonafide students of **"Ramaiah Institute of Technology"** in partial fulfillment of the requirements for the 5th semester of **"Bachelor of Engineering in Computer Science and Engineering (AI & ML)"** of Ramaiah Institute of Technology, Bengaluru-54, during the academic year 2024-2025. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report.

Signatures:

Signature of the Guide:
Dr. A N Ramya Shree
Associate Professor
Dept. of CSE (AI & ML)

Signature of the HoD:
Dr. Siddesh G M
Professor & Head
Dept. of CSE (AI & ML)

Acknowledgment

Any achievement does not depend solely on individual efforts but on the guidance, encouragement, and cooperation of intellectuals, elders, and friends. A number of personalities, in their own capacities, have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We would like to thank **Dr. Siddesh G M**, Professor & Head, Department of CSE (AI & ML), Ramaiah Institute of Technology, Bangalore-54.

We deeply express our sincere gratitude to our guide **Dr. A N Ramya Shree**, Associate Professor, Department of CSE (AI & ML), Ramaiah Institute of Technology, Bengaluru-54, for her able guidance, regular source of encouragement, and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of the Department of CSE (AI & ML), Ramaiah Institute of Technology, Bengaluru-54.

Abstract

Customer reviews often encompass opinions about multiple aspects of a product or service. While general sentiment analysis captures overall sentiment, it lacks the granularity needed to identify specific feature-level insights, such as design, usability, or performance. This project aims to bridge this gap by implementing an Aspect-Based Sentiment Analysis (ABSA) system. Using advanced Natural Language Processing (NLP) techniques and machine learning models, the system extracts and evaluates sentiments for individual aspects within a single review.

The implementation involves preprocessing data, extracting feature-specific feedback using dependency parsing, and classifying sentiments using logistic regression models trained on vectorized text data. The project extends its utility by incorporating a user-friendly Flask-based web application, enabling end-users to provide feedback and administrators to analyze sentiment-driven insights effectively. The system provides granular feedback on product attributes, empowering businesses to make targeted improvements.

This innovative approach to customer feedback analysis addresses a critical need in data-driven decision-making for businesses.

Table of Contents:

➢ INTRODUCTION:

Customer reviews often encompass multiple perspectives, reflecting opinions on various aspects of a product or service. While overall sentiment analysis helps determine general customer satisfaction, it fails to capture nuanced sentiments tied to specific features such as design, usability, performance, or pricing. This limitation restricts businesses from gaining targeted insights necessary for strategic improvements.

Aspect-Based Sentiment Analysis (ABSA) addresses this challenge by breaking down reviews into feature-specific sentiments. This project focuses on identifying and differentiating sentiments related to a product's design and usability within a single review. The implementation combines advanced NLP techniques like dependency parsing and sentiment classification with ML models. The system not only separates opinions but also evaluates their polarity—positive, neutral, or negative—providing a detailed understanding of customer feedback.

The methodology involves preprocessing textual data, extracting relevant aspects using custom-defined rules and ML classifiers, and predicting sentiments using state-of-the-art transformers. The output assists stakeholders in pinpointing strengths and areas for improvement in specific product features.

➢ PROBLEM STATEMENT:

Determining sentiment specific to different aspects of product or service within a single review. For example separating opinions about a product's design from its usability.

➢ IMPLEMENTATION:

Google Collab Code for building our ML Model:

➢ CODE:

```
import pandas as pd
# Load the dataset
file_path = '/content/Phone_Dataset.csv'
data = pd.read_csv(file_path)
# Display the first few rows
print("Dataset Preview:")
print(data.head())
import re
# Preprocessing function
df = data
def preprocess_text(text):
    # Basic text cleaning (remove special characters, convert to lowercase)
    text = re.sub(r"[^\w\s]", "", text)
    return text.lower()
# Apply preprocessing
df["Cleaned Review"] = df["Review"].apply(preprocess_text)
# Split into usability and design text (if needed)
def extract_aspect_reviews(review):
    parts = review.split("additionally")
    usability_part = parts[0].strip() if len(parts) > 0 else ""
    design_part = parts[1].strip() if len(parts) > 1 else ""
    return usability_part, design_part
df["Usability Feedback"], df["Design Feedback"] =
zip(*df["Review"].apply(extract_aspect_reviews))
print("Dataset after preprocessing:")
```

```
print(df.head())

!pip install scikit-learn

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report

# Generate dummy sentiments for usability and design for training

import random

df["Usability Sentiment"] = [random.choice(["Positive", "Negative"]) for _ in
range(len(df))]

df["Design Sentiment"] = [random.choice(["Positive", "Negative"]) for _ in
range(len(df))]

# Prepare data

vectorizer = CountVectorizer()

X = vectorizer.fit_transform(df["Cleaned Review"])

y_usability = df["Usability Sentiment"]

y_design = df["Design Sentiment"]

# Train-test split

X_train_u, X_test_u, y_train_u, y_test_u = train_test_split(X, y_usability,
test_size=0.2, random_state=42)

X_train_d, X_test_d, y_train_d, y_test_d = train_test_split(X, y_design,
test_size=0.2, random_state=42)

# Train models

usability_model = LogisticRegression()

design_model = LogisticRegression()

usability_model.fit(X_train_u, y_train_u)

design_model.fit(X_train_d, y_train_d)

# Evaluate

print("Usability Sentiment Analysis Report:")
```

```
print(classification_report(y_test_u, usability_model.predict(X_test_u)))
print("Design Sentiment Analysis Report:")
print(classification_report(y_test_d, design_model.predict(X_test_d)))
def predict_sentiment(review):
    # Clean the review
    cleaned_review = preprocess_text(review)
    # Transform using the vectorizer
    transformed_review = vectorizer.transform([cleaned_review])
    # Predict usability and design sentiments
    usability_pred = usability_model.predict(transformed_review)[0]
    design_pred = design_model.predict(transformed_review)[0]
    return {"usability_sentiment": usability_pred, "design_sentiment":
design_pred}
# Example new review
new_review = "The phone is easy to use and very responsive. But the design
isn't that good."
predictions = predict_sentiment(new_review)
print("Predictions for new review:", predictions)
import joblib
# Save the trained models
joblib.dump(usability_model, 'usability_model.pkl')  # Saves usability model
joblib.dump(design_model, 'design_model.pkl')        # Saves design model
# Save the vectorizer
joblib.dump(vectorizer, 'vectorizer.pkl')            # Saves the vectorizer
print("Models and vectorizer saved successfully!")
from google.colab import files
# Download the usability model
files.download('usability_model.pkl')
```

```
# Download the design model
files.download('design_model.pkl')
# Download the vectorizer
files.download('vectorizer.pkl')
import pandas as pd
# Load the dataset
file_path = '/content/Phone_Dataset.csv'
df= pd.read_csv(file_path)
# Count occurrences of each sentiment
sentiment_counts = df['Sentiment'].value_counts()
# Calculate percentage distribution
sentiment_percentages = (sentiment_counts / len(df)) * 100
# Display counts and percentages
print("Counts:")
print(sentiment_counts)
print("\nPercentages:")
print(sentiment_percentages)
```

```python
import pandas as pd

# Load the dataset
file_path = '/content/Phone_Dataset.csv'
data = pd.read_csv(file_path)

# Display the first few rows
print("Dataset Preview:")
print(data.head())

import re

# Preprocessing function

df = data
def preprocess_text(text):
    # Basic text cleaning (remove special characters, convert to lowercase)
    text = re.sub(r"[^\w\s]", "", text)
    return text.lower()

# Apply preprocessing
df["Cleaned Review"] = df["Review"].apply(preprocess_text)

# Split into usability and design text (if needed)
def extract_aspect_reviews(review):
    parts = review.split("additionally")
    usability_part = parts[0].strip() if len(parts) > 0 else ""
    design_part = parts[1].strip() if len(parts) > 1 else ""
    return usability_part, design_part

df["Usability Feedback"], df["Design Feedback"] = zip(*df["Review"].apply(extract_aspect_reviews))
```

```python
print("Dataset after preprocessing:")
print(df.head())
!pip install scikit-learn
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

# Generate dummy sentiments for usability and design for training
import random
df["Usability Sentiment"] = [random.choice(["Positive", "Negative"]) for _ in range(len(df))]
df["Design Sentiment"] = [random.choice(["Positive", "Negative"]) for _ in range(len(df))]

# Prepare data
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(df["Cleaned Review"])
y_usability = df["Usability Sentiment"]
y_design = df["Design Sentiment"]

# Train-test split
X_train_u, X_test_u, y_train_u, y_test_u = train_test_split(X, y_usability, test_size=0.2, random_state=42)
X_train_d, X_test_d, y_train_d, y_test_d = train_test_split(X, y_design, test_size=0.2, random_state=42)

# Train models
usability_model = LogisticRegression()
design_model = LogisticRegression()

usability_model.fit(X_train_u, y_train_u)
design_model.fit(X_train_d, y_train_d)
```

```python
# Evaluate
print("Usability Sentiment Analysis Report:")
print(classification_report(y_test_u, usability_model.predict(X_test_u)))

print("Design Sentiment Analysis Report:")
print(classification_report(y_test_d, design_model.predict(X_test_d)))
def predict_sentiment(review):
    # Clean the review
    cleaned_review = preprocess_text(review)
    # Transform using the vectorizer
    transformed_review = vectorizer.transform([cleaned_review])
    # Predict usability and design sentiments
    usability_pred = usability_model.predict(transformed_review)[0]
    design_pred = design_model.predict(transformed_review)[0]
    return {"usability_sentiment": usability_pred, "design_sentiment": design_pred}

# Example new review
new_review = "The phone is easy to use and very responsive. But the design isn't that good."
predictions = predict_sentiment(new_review)
print("Predictions for new review:", predictions)
import joblib

# Save the trained models
joblib.dump(usability_model, 'usability_model.pkl')  # Saves usability model
joblib.dump(design_model, 'design_model.pkl')        # Saves design model

# Save the vectorizer
joblib.dump(vectorizer, 'vectorizer.pkl')            # Saves the vectorizer

print("Models and vectorizer saved successfully!")
from google.colab import files
```

```python
# Download the usability model
files.download('usability_model.pkl')

# Download the design model
files.download('design_model.pkl')

# Download the vectorizer
files.download('vectorizer.pkl')

import pandas as pd

# Load the dataset
file_path = '/content/Phone_Dataset.csv'  # Update path if needed
df= pd.read_csv(file_path)
# Count occurrences of each sentiment

sentiment_counts = df['Sentiment'].value_counts()

# Calculate percentage distribution
sentiment_percentages = (sentiment_counts / len(df)) * 100

# Display counts and percentages
print("Counts:")
print(sentiment_counts)
print("\nPercentages:")
print(sentiment_percentages)
```

➢ GOOGLE COLAB OUTPUT:

```
+ Code  + Text

Dataset Preview:
   Customer Name Price Category Camera Quality  Sound Quality  Battery Life  \
0  Customer_1367     Affordable           Poor  Crystal clear      Moderate
1  Customer_2330      Mid-range           Good        Muffled      Moderate
2  Customer_2764      Mid-range        Average        Muffled  Long-lasting
3   Customer_358      Mid-range      Excellent  Crystal clear      Moderate
4   Customer_565     Affordable        Average        Muffled  Long-lasting

                                          Review Sentiment  \
0  The design is outstanding and usability is dec...   Neutral
1  The design is not too good and usability is no...  Negative
2           The design is bad and usability is bad.  Negative
3  The design is outstanding and usability is ama...  Positive
4  The design is outstanding and usability is ama...  Positive

  Usability Sentiment Design Sentiment
0            Neutral         Positive
1           Negative         Negative
2           Negative         Negative
3           Positive         Positive
4           Positive         Positive
Dataset after preprocessing:
   Customer Name Price Category Camera Quality  Sound Quality  Battery Life  \
0  Customer_1367     Affordable           Poor  Crystal clear      Moderate
1  Customer_2330      Mid-range           Good        Muffled      Moderate
2  Customer_2764      Mid-range        Average        Muffled  Long-lasting
3   Customer_358      Mid-range      Excellent  Crystal clear      Moderate
4   Customer_565     Affordable        Average        Muffled  Long-lasting
```

```
+ Code  + Text

                                          Review Sentiment  \
0  The design is outstanding and usability is dec...   Neutral
1  The design is not too good and usability is no...  Negative
2           The design is bad and usability is bad.  Negative
3  The design is outstanding and usability is ama...  Positive
4  The design is outstanding and usability is ama...  Positive

  Usability Sentiment Design Sentiment  \
0            Neutral         Positive
1           Negative         Negative
2           Negative         Negative
3           Positive         Positive
4           Positive         Positive

                                   Cleaned Review  \
0  the design is outstanding and usability is decent
1  the design is not too good and usability is no...
2            the design is bad and usability is bad
3  the design is outstanding and usability is ama...
4  the design is outstanding and usability is ama...

                                Usability Feedback Design Feedback
0  The design is outstanding and usability is dec...
1  The design is not too good and usability is no...
2           The design is bad and usability is bad.
3  The design is outstanding and usability is ama...
4  The design is outstanding and usability is ama...
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.6.0)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
```

```
+ Code  + Text

Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Usability Sentiment Analysis Report:
              precision    recall  f1-score   support

    Negative       0.50      0.45      0.47       478
    Positive       0.54      0.60      0.57       522

    accuracy                           0.52      1000
   macro avg       0.52      0.52      0.52      1000
weighted avg       0.52      0.52      0.52      1000

Design Sentiment Analysis Report:
              precision    recall  f1-score   support

    Negative       0.45      0.46      0.45       465
    Positive       0.52      0.52      0.52       535

    accuracy                           0.49      1000
   macro avg       0.49      0.49      0.49      1000
weighted avg       0.49      0.49      0.49      1000

Predictions for new review: {'usability_sentiment': 'Positive', 'design_sentiment': 'Positive'}
Models and vectorizer saved successfully!
Counts:
Sentiment
Neutral     1666
Negative    1666
Positive    1666
Name: count, dtype: int64
```

```
+ Code  + Text

Negative     1666
Positive     1666
Name: count, dtype: int64

Percentages:
Sentiment
Neutral     33.333333
Negative    33.333333
Positive    33.333333
Name: count, dtype: float64
```

➢ EXPLANATION:

This code performs sentiment analysis on a dataset of phone reviews using a machine learning approach. Here's a concise explanation:

1. Load and Preview Data: Reads a CSV dataset (Phone_Dataset.csv) and displays the first few rows for inspection.

2. Text Preprocessing: Cleans the reviews by removing special characters, converting text to lowercase, and splitting feedback into usability and design aspects.
3. Dummy Sentiment Labels: Generates random "Positive" or "Negative" sentiments for usability and design feedback as placeholders.
4. Feature Extraction: Converts cleaned reviews into numerical vectors using CountVectorizer for training models.
5. Data Splitting: Splits the dataset into training and test sets for usability and design sentiments.
6. Model Training: Trains two separate Logistic Regression models, one for usability sentiment and another for design sentiment.
7. Evaluation: Evaluates the models using the test set and prints classification reports for both.
8. Prediction: Defines a function to predict usability and design sentiments for new reviews using the trained models.
9. Save Models: Saves the trained models and vectorizer using joblib for future use.
10. File Downloads: Allows downloading of the saved models and vectorizer in a Google Colab environment.
11. Sentiment Analysis Summary: Calculates and displays sentiment counts and percentage distribution for the dataset.

This workflow demonstrates preprocessing, training, evaluation, and deployment of sentiment analysis models.

Creating Website using Flask:

Step 1: Open File Explorer

1. Navigate to a location where you want to create your project folder (e.g., Documents, Desktop).
2. Right-click in the folder and select New > Folder.
3. Name the folder as Flask_ML_Project.

Step 2: Create Sub-Folders and Files

1. Open the Flask_ML_Project folder.
2. Inside it, create the following:
   - A new folder named templates.
   - A new file named app.py.

Step 3: Add Files

1. Place HTML Files in the templates Folder:
   - Navigate to the templates folder.
   - Open Notepad or a code editor (e.g., VS Code or Sublime Text).

❖ **<u>The HTML FILES with CODE are as below:</u>**

<u>#home_page.html:</u>

```
<!DOCTYPE html>
<html>
<head>
  <title>Welcome</title>
  <style>
    /* General Styling */
    body {
      font-family: 'Arial', sans-serif;
      margin: 0;
      padding: 0;
      background: linear-gradient(135deg, #e0eafc, #cfdef3);
```

```
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
/* Center container styling */
.container {
    text-align: center;
    background: rgba(255, 255, 255, 0.9);
    padding: 30px;
    border-radius: 20px;
    box-shadow: 0px 10px 20px rgba(0, 0, 0, 0.2);
    max-width: 500px;
    width: 90%;
}
/* Heading styling */
h2 {
    font-size: 2.5rem;
    color: #333;
    margin-bottom: 20px;
    text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2);
}
/* Button styling */
button {
    font-size: 1.2rem;
    font-weight: bold;
    color: #fff;
```

```
    background: linear-gradient(135deg, #6a11cb, #2575fc);
    padding: 15px 30px;
    margin: 10px;
    border: none;
    border-radius: 50px;
    cursor: pointer;
    transition: all 0.3s ease-in-out;
    box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1);
    text-transform: uppercase;
}
/* Button hover effects */
button:hover {
    background: linear-gradient(135deg, #2575fc, #6a11cb);
    box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.3);
    transform: translateY(-5px);
}
/* Add button animation */
button:active {
    transform: translateY(2px);
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);
}
/* Footer text */
.footer {
    margin-top: 20px;
    font-size: 0.9rem;
    color: #555;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.1);
```

```
}
/* Shape decorations */
.circle {
    position: absolute;
    background: rgba(255, 255, 255, 0.5);
    border-radius: 50%;
    filter: blur(50px);
}
/* Circle styles */
.circle-1 {
    width: 200px;
    height: 200px;
    top: -50px;
    left: -50px;
}
.circle-2 {
    width: 300px;
    height: 300px;
    bottom: -100px;
    right: -100px;
}
.circle-3 {
    width: 150px;
    height: 150px;
    top: 50px;
    right: 20px;
}
```

```
    </style>
</head>
<body>
    <!-- Background circles for decoration -->
    <div class="circle circle-1"></div>
    <div class="circle circle-2"></div>
    <div class="circle circle-3"></div>
    <div class="container">
        <h2>Welcome to</h2>
          <h2> One Plus Nord 3 5G Phone's</h2>
          <h2> Feedback Portal</h2>
        <button onclick="window.location.href='/feedback'">Provide
Feedback</button>
        <button onclick="window.location.href='/admin-login'">Admin
Login</button>
        <div class="footer">
            <p>We value your feedback and insights!</p>
        </div>
    </div>
</body>
</html>
```

#feedback_form.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Feedback Form</title>
<style>
    /* Global styles */
    body {
        font-family: 'Poppins', sans-serif;
        background: linear-gradient(45deg, #4facfe, #00f2fe);
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
        overflow: auto;
    }
    /* Decorative background shapes */
    .circle {
        position: absolute;
        border-radius: 50%;
        opacity: 0.1;
    }
    .circle-1 {
        width: 200px;
        height: 200px;
        background-color: #fff;
        top: 10%;
        left: 10%;
```

```
}
.circle-2 {
    width: 300px;
    height: 300px;
    background-color: #fff;
    top: 60%;
    right: 20%;
}
.circle-3 {
    width: 250px;
    height: 250px;
    background-color: #fff;
    bottom: 15%;
    left: 50%;
}
/* Form container */
.container {
    background-color: rgba(255, 255, 255, 0.95);
    padding: 30px 40px;
    border-radius: 15px;
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
    width: 80%;
    max-width: 750px;
    z-index: 10;
}
h2 {
    color: #003f8a;
```

```
    text-align: center;
    margin-bottom: 20px;
    font-size: 30px;
    font-weight: bold;
    text-transform: uppercase;
}
/* Form input styles */
label {
    font-size: 18px;
    font-weight: bold;
    color: #007acc;
    text-decoration: underline;
    margin-top: 20px;
    display: block;
}
input[type="text"], input[type="email"], textarea {
    width: 100%;
    padding: 14px;
    margin-top: 6px;
    border-radius: 8px;
    border: 1px solid #ddd;
    font-size: 16px;
    font-family: 'Poppins', sans-serif;
    transition: border-color 0.3s ease;
    margin-bottom: 15px; /* Added space after each field */
}
input[type="text"]:focus, input[type="email"]:focus, textarea:focus {
```

```
    border-color: #007acc;
    outline: none;
    box-shadow: 0 0 6px rgba(0, 122, 204, 0.5);
}
textarea {
    resize: vertical;
    height: 120px;
}
/* Radio button group */
.radio-group {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(100px, 1fr));
    gap: 10px 20px;
    margin: 10px 0 20px;
}
.radio-group div {
    display: flex;
    align-items: center;
}
.radio-group input[type="radio"] {
    margin-right: 8px;
    transform: scale(1.2);
    accent-color: #007acc; /* Stylish checkbox */
}
.radio-group label {
    font-size: 15px;
    color: #555;
```

```
    font-weight: normal;
}
/* Submit button */
button[type="submit"] {
    background-color: #007acc;
    color: white;
    border: none;
    padding: 14px 20px;
    border-radius: 8px;
    font-size: 18px;
    font-family: 'Poppins', sans-serif;
    cursor: pointer;
    width: 100%;
    margin-top: 20px;
    transition: background-color 0.3s ease, box-shadow 0.3s ease;
}
button[type="submit"]:hover {
    background-color: #005ea6;
    box-shadow: 0 0 10px rgba(0, 122, 204, 0.6);
}
/* Responsive styles */
@media (max-width: 600px) {
    .container {
        width: 90%;
        padding: 20px;
    }
    h2 {
```

```
            font-size: 26px;
        }
    }
</style>
</head>
<body>
    <!-- Decorative background shapes -->
    <div class="circle circle-1"></div>
    <div class="circle circle-2"></div>
    <div class="circle circle-3"></div>
    <!-- Form container -->
    <div class="container">
        <h2>Feedback Form</h2>
        <form method="POST" action="/submit-feedback">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
            <label for="phone">Phone:</label>
            <input type="text" id="phone" name="phone" required>
            <!-- New Design Review -->
            <label for="design_review">Design Review:</label>
            <div class="radio-group">
                <div>
                    <input type="radio" id="design_outstanding"
name="design_review" value="outstanding">
                    <label for="design_outstanding">Outstanding</label>
                </div>
```

```
            <div>
                <input type="radio" id="design_best" name="design_review"
value="best">
                <label for="design_best">Best</label>
            </div>
            <div>
                <input type="radio" id="design_good" name="design_review"
value="good">
                <label for="design_good">Good</label>
            </div>
            <div>
                <input type="radio" id="design_poor" name="design_review"
value="poor">
                <label for="design_poor">Poor</label>
            </div>
            <div>
                <input type="radio" id="design_worst" name="design_review"
value="worst">
                <label for="design_worst">Worst</label>
            </div>
            <div>
                <input type="radio" id="design_terrible" name="design_review"
value="terrible">
                <label for="design_terrible">Terrible</label>
            </div>
        </div>
        <!-- New Usability Review -->
        <label for="usability_review">Usability Review:</label>
        <div class="radio-group">
```

```
            <div>
                <input type="radio" id="usability_outstanding"
name="usability_review" value="outstanding">
                <label for="usability_outstanding">Outstanding</label>
            </div>
            <div>
                <input type="radio" id="usability_best" name="usability_review"
value="best">
                <label for="usability_best">Best</label>
            </div>
            <div>
                <input type="radio" id="usability_good" name="usability_review"
value="good">
                <label for="usability_good">Good</label>
            </div>
            <div>
                <input type="radio" id="usability_poor" name="usability_review"
value="poor">
                <label for="usability_poor">Poor</label>
            </div>
            <div>
                <input type="radio" id="usability_worst" name="usability_review"
value="worst">
                <label for="usability_worst">Worst</label>
            </div>
            <div>
                <input type="radio" id="usability_terrible"
name="usability_review" value="terrible">
                <label for="usability_terrible">Terrible</label>
```

```
            </div>
        </div>
        <!-- New Price Review -->
        <label for="price_review">Price Review:</label>
        <div class="radio-group">
            <div>
                <input type="radio" id="price_outstanding" name="price_review"
value="outstanding">
                <label for="price_outstanding">Outstanding</label>
            </div>
            <div>
                <input type="radio" id="price_best" name="price_review"
value="best">
                <label for="price_best">Best</label>
            </div>
            <div>
                <input type="radio" id="price_good" name="price_review"
value="good">
                <label for="price_good">Good</label>
            </div>
            <div>
                <input type="radio" id="price_poor" name="price_review"
value="poor">
                <label for="price_poor">Poor</label>
            </div>
            <div>
                <input type="radio" id="price_worst" name="price_review"
value="worst">
                <label for="price_worst">Worst</label>
```

```
            </div>
            <div>
                <input type="radio" id="price_terrible" name="price_review"
value="terrible">
                <label for="price_terrible">Terrible</label>
            </div>
        </div>
        <!-- New Sound Quality Review -->
        <label for="sound_quality_review">Sound Quality Review:</label>
        <div class="radio-group">
            <div>
                <input type="radio" id="sound_quality_outstanding"
name="sound_quality_review" value="outstanding">
                <label for="sound_quality_outstanding">Outstanding</label>
            </div>
            <div>
                <input type="radio" id="sound_quality_best"
name="sound_quality_review" value="best">
                <label for="sound_quality_best">Best</label>
            </div>
            <div>
                <input type="radio" id="sound_quality_good"
name="sound_quality_review" value="good">
                <label for="sound_quality_good">Good</label>
            </div>
            <div>
                <input type="radio" id="sound_quality_poor"
name="sound_quality_review" value="poor">
                <label for="sound_quality_poor">Poor</label>
```

```
            </div>
            <div>
                <input type="radio" id="sound_quality_worst"
name="sound_quality_review" value="worst">
                <label for="sound_quality_worst">Worst</label>
            </div>
            <div>
                <input type="radio" id="sound_quality_terrible"
name="sound_quality_review" value="terrible">
                <label for="sound_quality_terrible">Terrible</label>
            </div>
        </div>
        <!-- New Camera Quality Review -->
        <label for="camera_quality_review">Camera Quality Review:</label>
        <div class="radio-group">
            <div>
                <input type="radio" id="camera_quality_outstanding"
name="camera_quality_review" value="outstanding">
                <label for="camera_quality_outstanding">Outstanding</label>
            </div>
            <div>
                <input type="radio" id="camera_quality_best"
name="camera_quality_review" value="best">
                <label for="camera_quality_best">Best</label>
            </div>
            <div>
                <input type="radio" id="camera_quality_good"
name="camera_quality_review" value="good">
                <label for="camera_quality_good">Good</label>
```

```
            </div>
            <div>
                <input type="radio" id="camera_quality_poor"
name="camera_quality_review" value="poor">
                <label for="camera_quality_poor">Poor</label>
            </div>
            <div>
                <input type="radio" id="camera_quality_worst"
name="camera_quality_review" value="worst">
                <label for="camera_quality_worst">Worst</label>
            </div>
            <div>
                <input type="radio" id="camera_quality_terrible"
name="camera_quality_review" value="terrible">
                <label for="camera_quality_terrible">Terrible</label>
            </div>
        </div>
        <!-- New Battery Life Review -->
        <label for="battery_life_review">Battery Life Review:</label>
        <div class="radio-group">
            <div>
                <input type="radio" id="battery_life_outstanding"
name="battery_life_review" value="outstanding">
                <label for="battery_life_outstanding">Outstanding</label>
            </div>
            <div>
                <input type="radio" id="battery_life_best"
name="battery_life_review" value="best">
                <label for="battery_life_best">Best</label>
```

```
            </div>
            <div>
                <input type="radio" id="battery_life_good"
name="battery_life_review" value="good">
                <label for="battery_life_good">Good</label>
            </div>
            <div>
                <input type="radio" id="battery_life_poor"
name="battery_life_review" value="poor">
                <label for="battery_life_poor">Poor</label>
            </div>
            <div>
                <input type="radio" id="battery_life_worst"
name="battery_life_review" value="worst">
                <label for="battery_life_worst">Worst</label>
            </div>
            <div>
                <input type="radio" id="battery_life_terrible"
name="battery_life_review" value="terrible">
                <label for="battery_life_terrible">Terrible</label>
            </div>
        </div>
        <!-- Renamed Suggestions Field -->

          <label for="suggestions">Suggestions:</label>
          <textarea id="suggestions" name="suggestions" placeholder="Your
suggestions..." required></textarea>
        <button type="submit">Submit</button>
    </form>
```

```
    </div>
</body>
</html>
```

#submission_confirmation.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Feedback Submitted</title>
    <style>
        /* General body styling */
        body {
            font-family: 'Arial', sans-serif;
            margin: 0;
            padding: 0;
            background: linear-gradient(135deg, #ff9a9e, #fad0c4);
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            color: #333;
        }
        /* Center container styling */
        .container {
            text-align: center;
            background: rgba(255, 255, 255, 0.9);
```

```
    padding: 40px;
    border-radius: 20px;
    box-shadow: 0px 10px 20px rgba(0, 0, 0, 0.2);
    max-width: 600px;
    width: 90%;
}
/* Heading styling */
h2 {
    font-size: 2.5rem;
    color: #333;
    margin-bottom: 20px;
    text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2);
}
/* Paragraph styling */
p {
    font-size: 1.2rem;
    margin-bottom: 20px;
    color: #555;
    line-height: 1.6;
}
/* Link button styling */
a {
    text-decoration: none;
    color: #fff;
    background: linear-gradient(135deg, #36d1dc, #5b86e5);
    padding: 15px 30px;
    font-size: 1.2rem;
```

```
    font-weight: bold;
    border-radius: 50px;
    box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1);
    transition: all 0.3s ease-in-out;
}
/* Hover effects for link button */
a:hover {
    background: linear-gradient(135deg, #5b86e5, #36d1dc);
    box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.3);
    transform: translateY(-5px);
}
/* Add click animation */
a:active {
    transform: translateY(2px);
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);
}
/* Decorative circles for aesthetic design */
.circle {
    position: absolute;
    background: rgba(255, 255, 255, 0.3);
    border-radius: 50%;
    filter: blur(50px);
}
/* Different circle sizes and positions */
.circle-1 {
    width: 200px;
    height: 200px;
```

```
            top: -50px;
            left: -50px;
        }
        .circle-2 {
            width: 300px;
            height: 300px;
            bottom: -100px;
            right: -100px;
        }
        .circle-3 {
            width: 150px;
            height: 150px;
            top: 50px;
            right: 20px;
        }
    </style>
</head>
<body>
    <!-- Decorative circles -->
    <div class="circle circle-1"></div>
    <div class="circle circle-2"></div>
    <div class="circle circle-3"></div>
    <!-- Main container -->
    <div class="container">
        <h2>Thank You!</h2>
        <p>Your feedback has been submitted successfully. We appreciate your
effort in helping us improve!</p>
        <a href="/">Go back to Home</a>
```

```
    </div>
</body>
</html>
```

#admin_login.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Admin Login</title>
    <style>
        /* General body styling */
        body {
            font-family: 'Arial', sans-serif;
            margin: 0;
            padding: 0;
            background: linear-gradient(135deg, #ff9a9e, #fad0c4);
            height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;
            color: #333;
        }
        /* Login container styling */
        .login-container {
            background: rgba(255, 255, 255, 0.9);
            border-radius: 15px;
```

```
    padding: 40px;
    box-shadow: 0px 10px 20px rgba(0, 0, 0, 0.2);
    width: 300px;
    text-align: center;
}
/* Heading styling */
h2 {
    font-size: 2rem;
    color: #ff6b6b;
    margin-bottom: 20px;
}
/* Input field styling */
input[type="text"], input[type="password"] {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ddd;
    border-radius: 8px;
    font-size: 1rem;
}
input[type="text"]:focus, input[type="password"]:focus {
    outline: none;
    border: 1px solid #ff6b6b;
    box-shadow: 0px 2px 5px rgba(255, 107, 107, 0.4);
}
/* Button styling */
button {
```

```
    width: 100%;
    padding: 10px;
    background: #ff6b6b;
    color: white;
    font-size: 1rem;
    font-weight: bold;
    border: none;
    border-radius: 8px;
    cursor: pointer;
    transition: all 0.3s ease-in-out;
    margin-top: 10px;
}
button:hover {
    background: #ff7878;
    box-shadow: 0px 5px 15px rgba(255, 107, 107, 0.3);
}
button:active {
    transform: translateY(2px);
}
/* Flash message styling */
p {
    color: red;
    font-size: 0.9rem;
    margin-top: 10px;
}
/* Decorative shapes */
.circle {
```

```
            position: absolute;
            background: rgba(255, 255, 255, 0.3);
            border-radius: 50%;
            filter: blur(50px);
        }
        .circle-1 {
            width: 250px;
            height: 250px;
            top: -50px;
            left: -80px;
        }
        .circle-2 {
            width: 200px;
            height: 200px;
            bottom: -60px;
            right: -60px;
        }
    </style>
</head>
<body>
    <!-- Decorative background shapes -->
    <div class="circle circle-1"></div>
    <div class="circle circle-2"></div>
    <!-- Login container -->
    <div class="login-container">
        <h2>Admin Login</h2>
        <form method="POST" action="/admin-login">
```

```
        <label for="username">Username:</label><br>
        <input type="text" id="username" name="username" required><br>
        <label for="password">Password:</label><br>
        <input type="password" id="password" name="password"
required><br>
        <button type="submit">Login</button>
    </form>
    {% with messages = get_flashed_messages() %}
        {% if messages %}
            <p>{{ messages[0] }}</p>
        {% endif %}
    {% endwith %}
  </div>
</body>
</html>
```

#admin_dashboard.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Admin Dashboard</title>
  <style>
    /* General Styling */
    body {
        font-family: 'Arial', sans-serif;
        background-color: #f4f7fc;
        margin: 0;
```

```
    padding: 0;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}
h1 {
    font-size: 2.8em;
    font-weight: 600;
    color: #1f78d1;
    text-align: center;
    margin-top: 20px;
}
.logout-button-container {
    width: 100%;
    display: flex;
    justify-content: flex-end;
    margin: 20px 0;
    padding: 0 10px;
}
.logout-button {
    background-color: #f56b2a;
    color: white;
    padding: 12px 30px;
    border-radius: 50px;
    font-size: 1.1em;
    text-decoration: none;
```

```
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.15);
    border: none;
    cursor: pointer;
    transition: all 0.3s ease;
}
.logout-button:hover {
    background-color: #d75a22;
    transform: translateY(-4px);
}
.table-container {
    width: 100%;
    max-height: 500px; /* Restrict height for vertical scrolling */
    overflow-y: auto;
    padding: 0 10px; /* Add some padding to prevent edge overlap */
}
table {
    width: 100%;
    border-collapse: collapse;
    font-size: 1.1em;
    color: #333;
}
th, td {
    padding: 15px;
    text-align: left;
    border-bottom: 1px solid #ddd;
}
th {
```

```
            background-color: #1f78d1;
            color: white;
        }
        tr:nth-child(even) {
            background-color: #f9f9f9;
        }
        tr:hover {
            background-color: #f7f7f7;
        }
        /* Responsive Design */
        @media (max-width: 768px) {
            h1 {
                font-size: 2.2em;
            }
            .logout-button {
                font-size: 1em;
            }
            table {
                font-size: 1em;
            }
            th, td {
                padding: 10px;
            }
        }
    </style>
</head>
<body>
```

```
<h1>Admin Dashboard</h1>
<div class="logout-button-container">
    <a href="/logout" class="logout-button">Log Out</a>
</div>
<div class="table-container">
    <table>
        <thead>
            <tr>
                <th>Name</th>
                <th>Email</th>
                <th>Phone</th>
                <th>Price Sentiment</th>
                <th>Sound Quality Sentiment</th>
                <th>Battery Life Sentiment</th>
                <th>Camera Sentiment</th>
                <th>Usability Sentiment</th>
                <th>Design Sentiment</th>
                <th>Customer's Suggestion</th>
                <th>Product Review</th>
            </tr>
        </thead>
        <tbody>
            {% for feedback in feedbacks %}
            <tr>
                <td>{{ feedback['name'] }}</td>
                <td>{{ feedback['email'] }}</td>
                <td>{{ feedback['phone'] }}</td>
```

```
                <td>{{ feedback['price_sentiment'] }}</td>
                <td>{{ feedback['sound_sentiment'] }}</td>
                <td>{{ feedback['battery_life_sentiment'] }}</td>
                <td>{{ feedback['camera_sentiment'] }}</td>
                <td>{{ feedback['usability_sentiment'] }}</td>
                <td>{{ feedback['design_sentiment'] }}</td>
                <td>{{ feedback['suggestions'] }}</td>
                <td>{{ feedback['product_review'] }}</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
  </div>
</body>
</html>
```

---

2. Place Python Code in app.py:
   - Go back to the main Flask_ML_Project folder.
   - Open app.py in your code editor.
   - Save the file.

#app.py:

```
from flask import Flask, render_template, request, redirect, url_for, flash, session
import joblib
app = Flask(__name__)
```

```
app.secret_key = 'your_secret_key'
# Load models and vectorizer
usability_model = joblib.load('usability_model.pkl')
design_model = joblib.load('design_model.pkl')
vectorizer = joblib.load('vectorizer.pkl')
# Hardcoded admin credentials
ADMIN_USERNAME = "admin"
ADMIN_PASSWORD = "123"
# List to store feedback data
feedback_list = []
# Positive and Negative Sentiment Keywords
POSITIVE_KEYWORDS = ["good", "nice", "very good", "best",
"outstanding", "fantastic", "great", "excellent"]
NEGATIVE_KEYWORDS = ["bad", "poor", "very bad", "disappointed",
"worst", "terrible"]
# Helper function to extract sentiment
def extract_sentiment(review_type):
    if review_type in POSITIVE_KEYWORDS:
        return "Positive"
    elif review_type in NEGATIVE_KEYWORDS:
        return "Negative"
    return "Neutral"
# Helper function to calculate product review sentiment
def calculate_product_review(sentiments):
    positive_count = sentiments.count("Positive")
    negative_count = sentiments.count("Negative")
    if positive_count > 3:
        return "Positive"
```

```
    elif negative_count > 3:
        return "Negative"
    return "Neutral"
# Main home page with options
@app.route('/')
def main_home():
    return render_template('home_page.html')
# Feedback form
@app.route('/feedback')
def feedback():
    return render_template('feedback_form.html')
# Process feedback
@app.route('/submit-feedback', methods=['POST'])
def submit_feedback():
    if request.method == 'POST':
        name = request.form.get('name')
        email = request.form.get('email')
        phone = request.form.get('phone')
        suggestions = request.form.get('suggestions')
        # Sentiments for different aspects
        usability_review = extract_sentiment(request.form.get('usability_review'))
        design_review = extract_sentiment(request.form.get('design_review'))
        price_review = extract_sentiment(request.form.get('price_review'))
        sound_review =
extract_sentiment(request.form.get('sound_quality_review'))
        battery_review =
extract_sentiment(request.form.get('battery_life_review'))
```

```
        camera_review =
extract_sentiment(request.form.get('camera_quality_review'))
        # Calculate overall product review
        sentiments = [usability_review, design_review, price_review,
sound_review, battery_review, camera_review]
        product_review = calculate_product_review(sentiments)
# Store feedback
        feedback_list.append({
            'name': name,
            'email': email,
            'phone': phone,
            'usability_sentiment': usability_review,
            'design_sentiment': design_review,
            'price_sentiment': price_review,
            'sound_sentiment': sound_review,
            'battery_life_sentiment': battery_review,
            'camera_sentiment': camera_review,
            'suggestions': suggestions,
            'product_review': product_review
        })
        return render_template('submission_confirmation.html')
# Admin login
@app.route('/admin-login', methods=['GET', 'POST'])
def admin_login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
```

```
        if username == ADMIN_USERNAME and password ==
ADMIN_PASSWORD:
            session['logged_in'] = True
            return redirect(url_for('admin_dashboard'))
        else:
            flash("Invalid username or password!")
            return redirect(url_for('admin_login'))
    return render_template('admin_login.html')
# Admin dashboard
@app.route('/admin-dashboard')
def admin_dashboard():
    if not session.get('logged_in'):
        return redirect(url_for('admin_login'))
    return render_template('admin_dashboard.html', feedbacks=feedback_list)
# Admin logout
@app.route('/logout')
def logout():
    session['logged_in'] = False
    return redirect(url_for('admin_login'))
if __name__ == '__main__':
    app.run(debug=True)
```

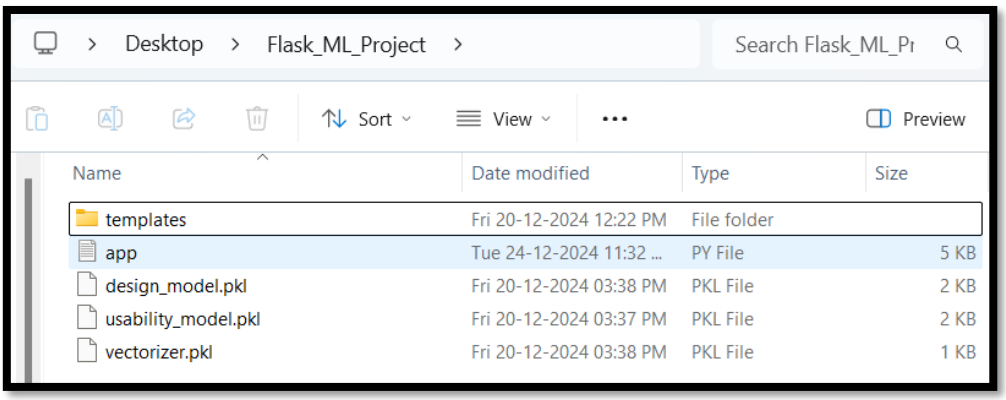
Step 4: Place .pkl Files

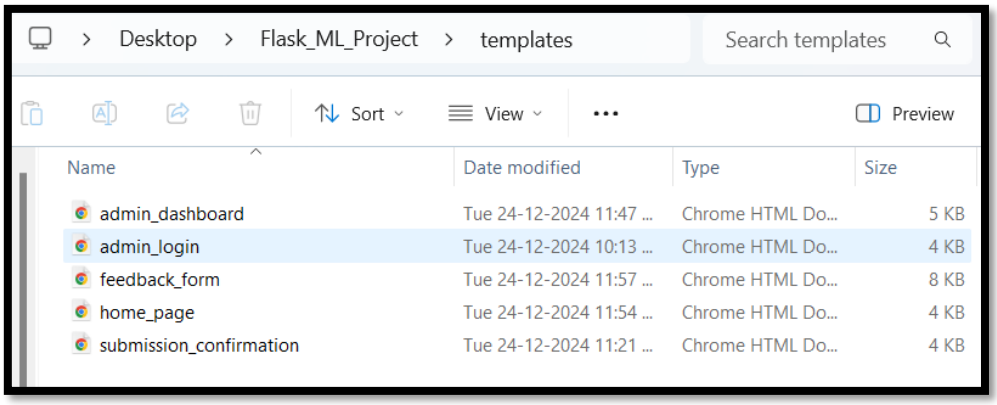
1. Download the usability_model.pkl, design_model.pkl, and vectorizer.pkl files from your Colab.
    - Place the downloaded files directly into the Flask_ML_Project folder (not in the templates sub-folder).

Step 5: Verify the Directory Structure

Your folder should now look like this:

```
Flask_ML_Project/
├── app.py
├── templates/
│   ├── home_page.html
│   ├── feedback_form.html
│   ├── submission_confirmation.html
│   ├── admin_login.html
│   ├── admin_dashboard.html
├── usability_model.pkl
├── design_model.pkl
├── vectorizer.pkl
```





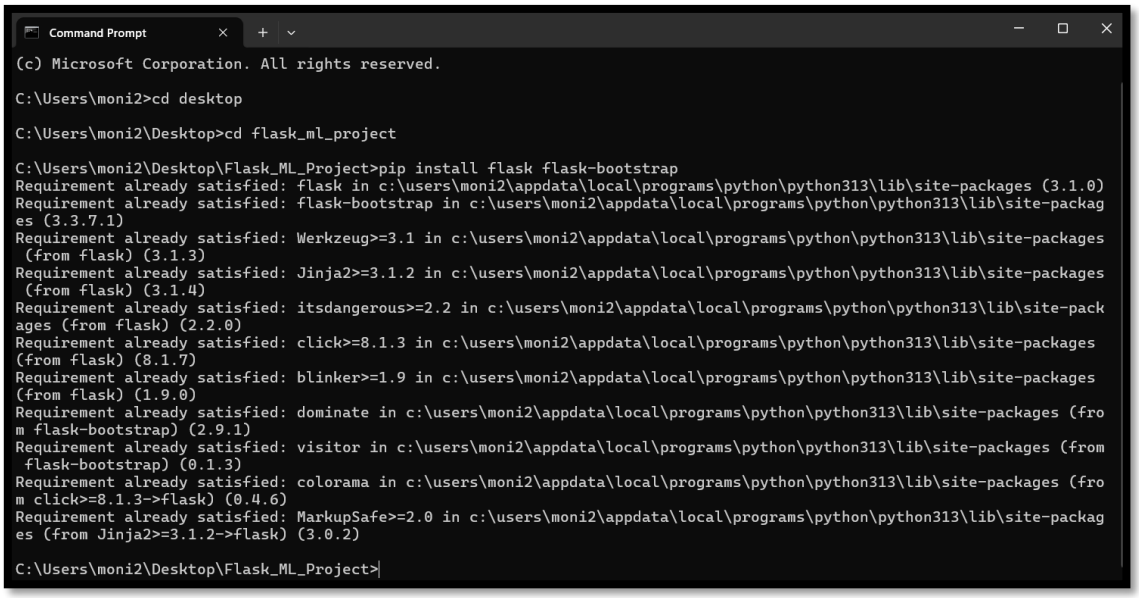
Step 6: Install Required Tools

1. Open Command Prompt:
   - Press Win + R, type cmd, and hit Enter.
2. Navigate to the Project Folder:
   - Use the cd command to navigate to the folder:
   - cd path\to\Flask_ML_Project

For example:

cd C:\Users\YourUsername\Desktop\Flask_ML_Project

3. Install Flask: Run the following command:
4. pip install flask flask-bootstrap



```
Command Prompt
(c) Microsoft Corporation. All rights reserved.

C:\Users\moni2>cd desktop

C:\Users\moni2\Desktop>cd flask_ml_project

C:\Users\moni2\Desktop\Flask_ML_Project>pip install flask flask-bootstrap
Requirement already satisfied: flask in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packages (3.1.0)
Requirement already satisfied: flask-bootstrap in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packag
es (3.3.7.1)
Requirement already satisfied: Werkzeug>=3.1 in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packages
 (from flask) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packages
 (from flask) (3.1.4)
Requirement already satisfied: itsdangerous>=2.2 in c:\users\moni2\appdata\local\programs\python\python313\lib\site-pack
ages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packages
(from flask) (8.1.7)
Requirement already satisfied: blinker>=1.9 in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packages
(from flask) (1.9.0)
Requirement already satisfied: dominate in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packages (fro
m flask-bootstrap) (2.9.1)
Requirement already satisfied: visitor in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packages (from
 flask-bootstrap) (0.1.3)
Requirement already satisfied: colorama in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packages (fro
m click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\moni2\appdata\local\programs\python\python313\lib\site-packag
es (from Jinja2>=3.1.2->flask) (3.0.2)

C:\Users\moni2\Desktop\Flask_ML_Project>
```

Step 7: Run the Flask App

1. In the same command prompt, run:
2. python app.py
3. If everything is set up correctly, you'll see an output like:
4. * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
5. Open your browser and visit http://127.0.0.1:5000.

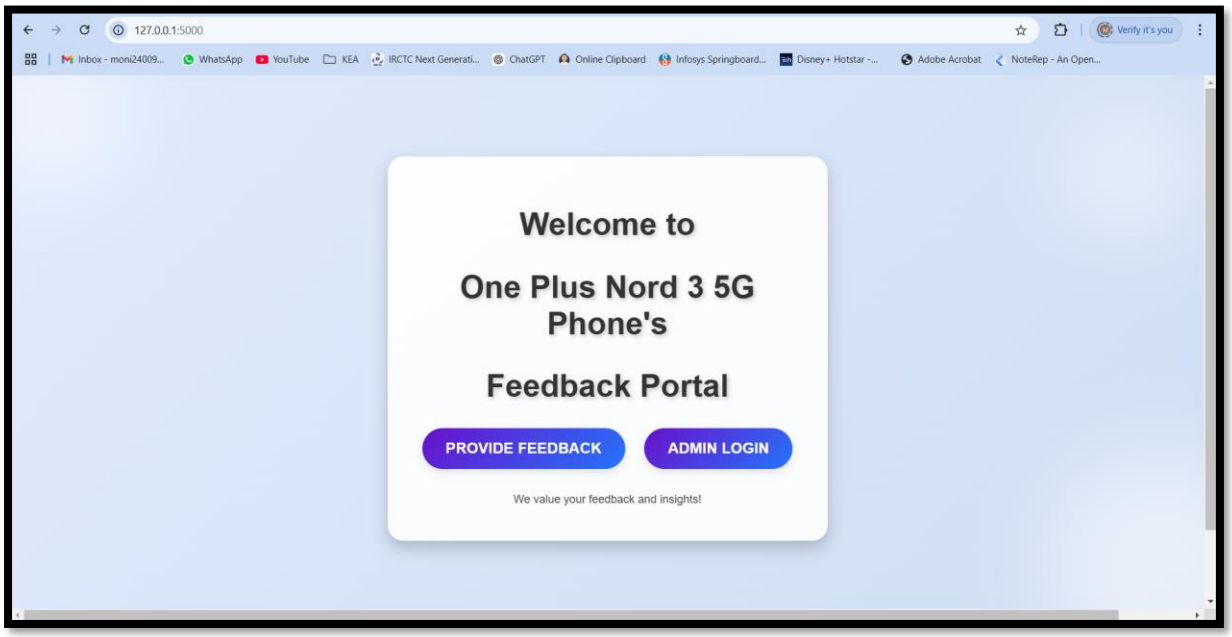
```
C:\Users\moni2\Desktop\Flask_ML_Project>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 113-363-681
```

Step 8: Test the App

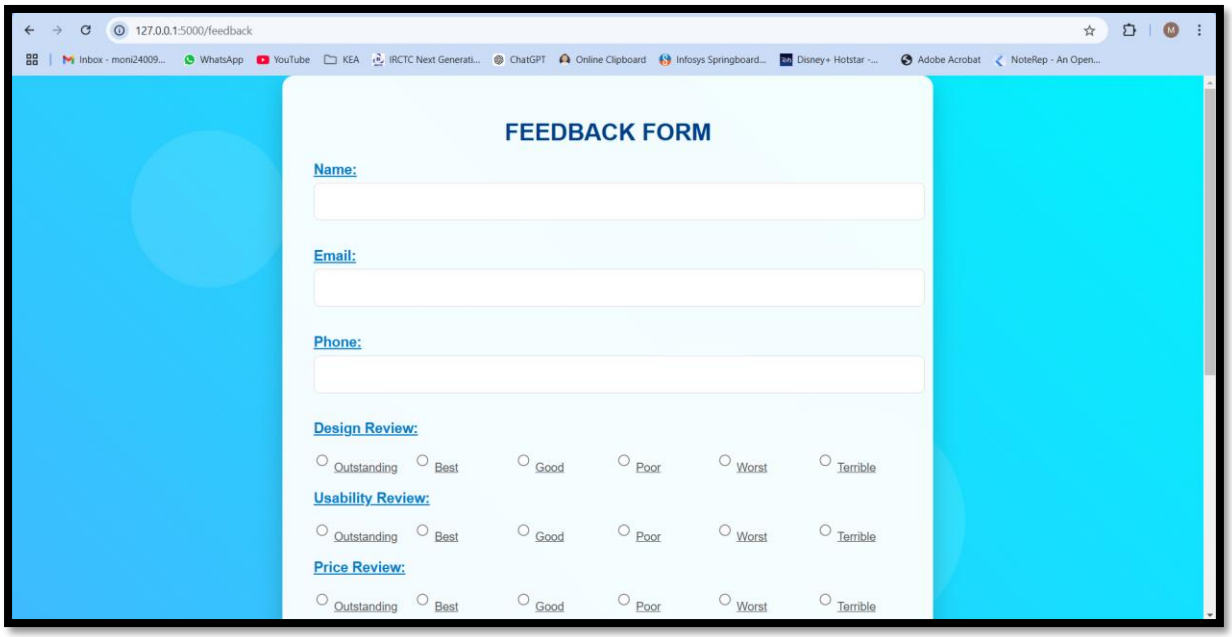
1. Fill in the feedback form on the homepage.
2. Submit it and check the admin dashboard for feedback analysis.

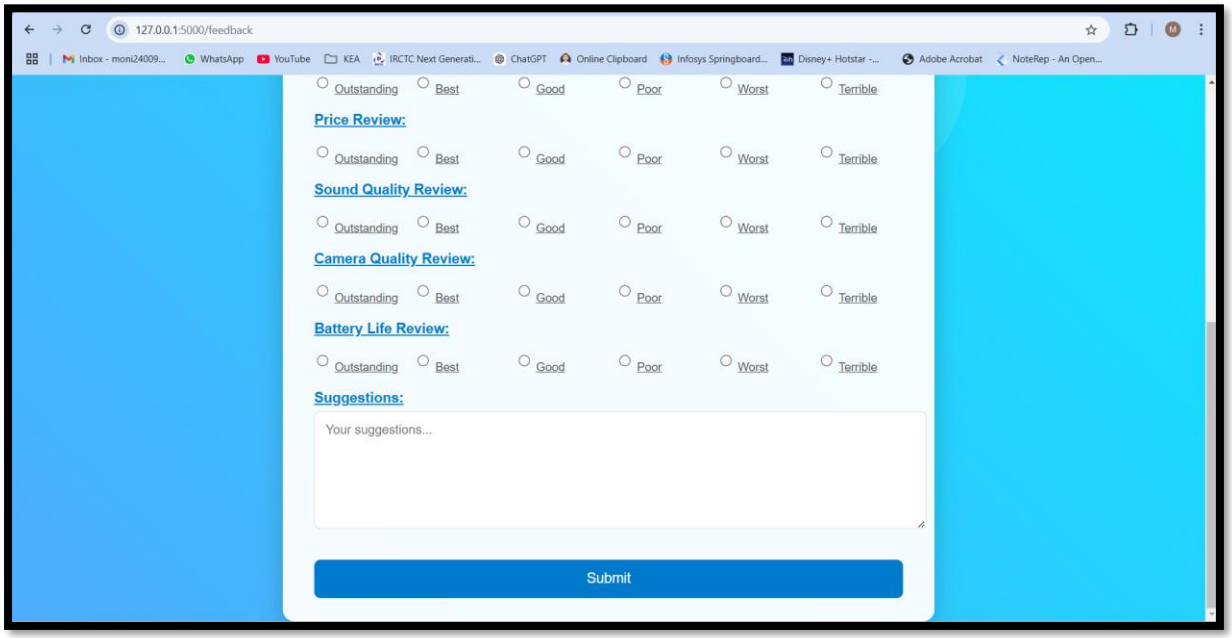
➢ **OUTPUT:**

home_page.html

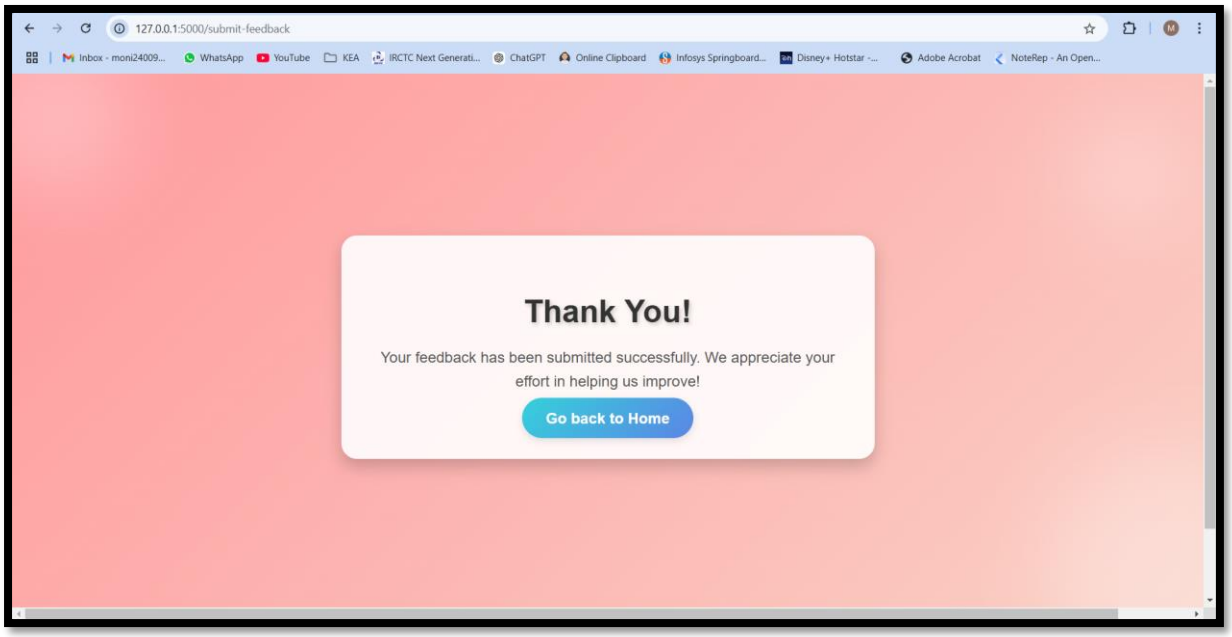


feedback_form.html

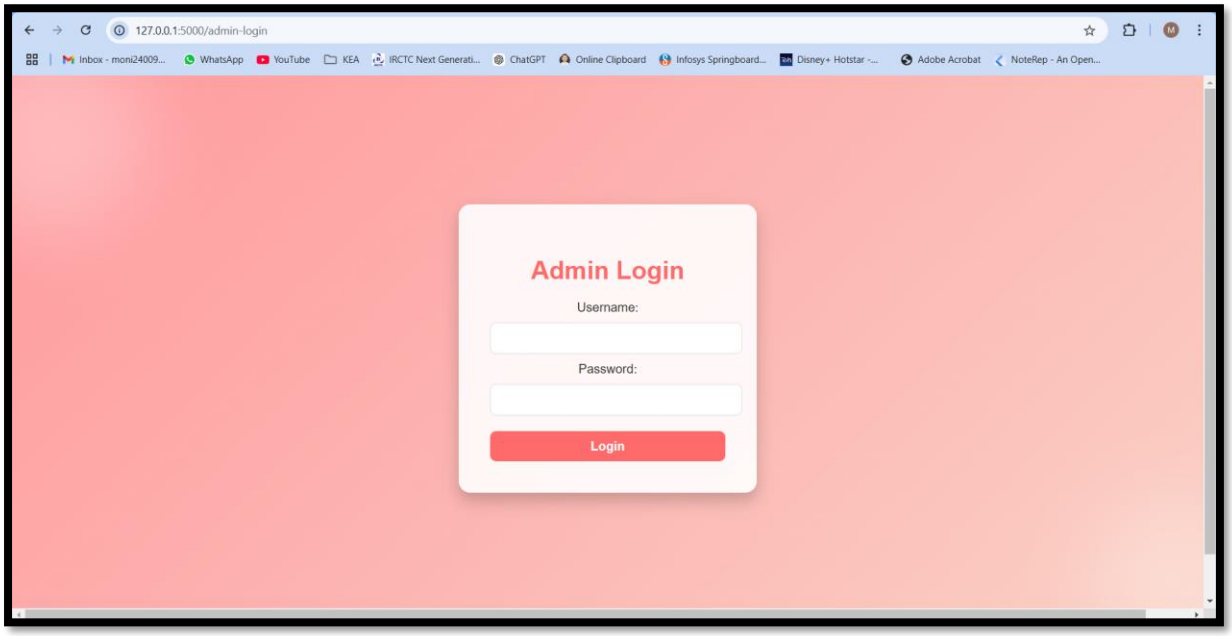




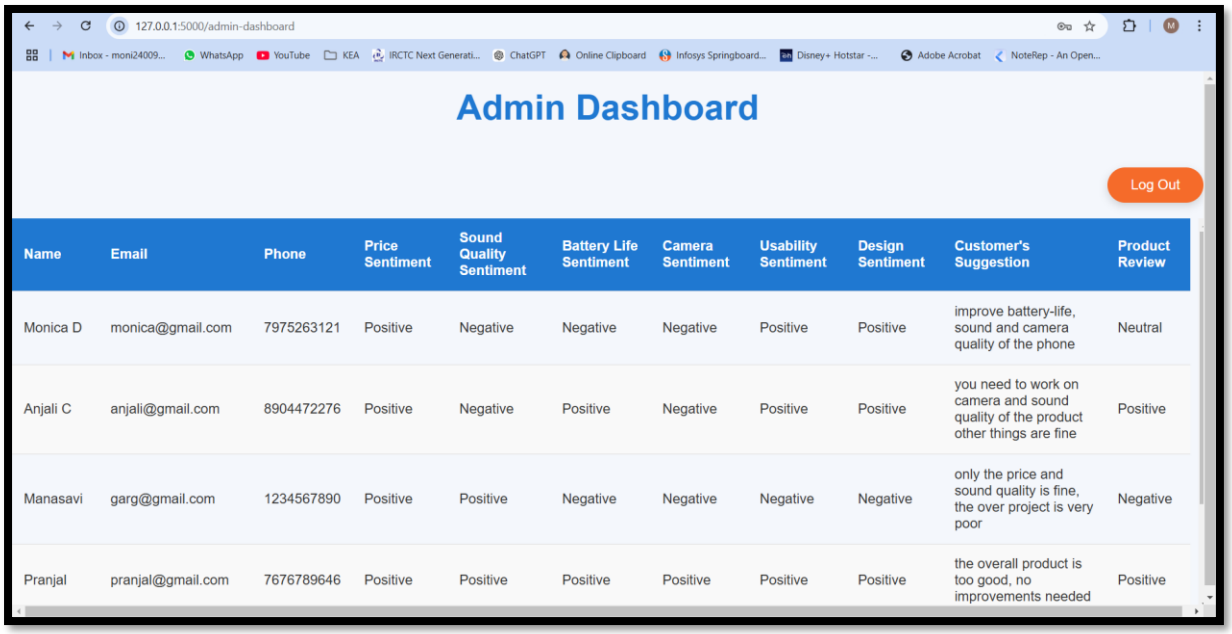
submission_confirmation.html



admin_login.html



admin_dashboard.html

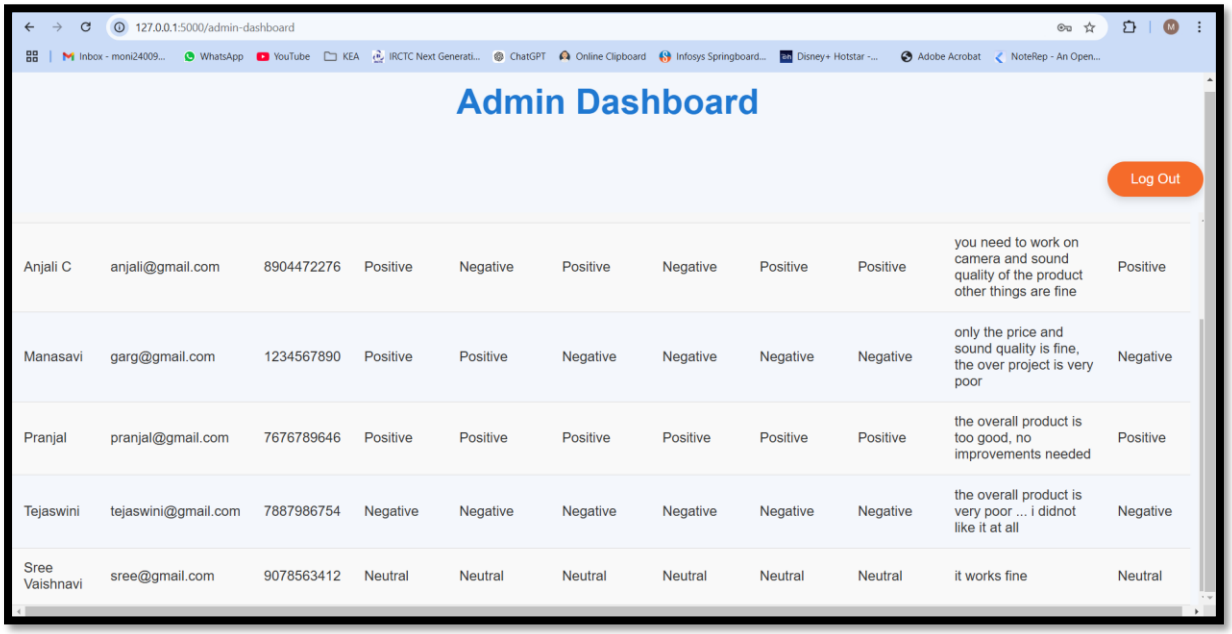


127.0.0.1:5000/admin-dashboard

Admin Dashboard

Log Out

Name	Email	Phone	Price Sentiment	Sound Quality Sentiment	Battery Life Sentiment	Camera Sentiment	Usability Sentiment	Design Sentiment	Customer's Suggestion	Product Review
Monica D	monica@gmail.com	7975263121	Positive	Negative	Negative	Negative	Positive	Positive	improve battery-life, sound and camera quality of the phone	Neutral
Anjali C	anjali@gmail.com	8904472276	Positive	Negative	Positive	Negative	Positive	Positive	you need to work on camera and sound quality of the product other things are fine	Positive
Manasavi	garg@gmail.com	1234567890	Positive	Positive	Negative	Negative	Negative	Negative	only the price and sound quality is fine, the over project is very poor	Negative
Pranjal	pranjal@gmail.com	7676789646	Positive	Positive	Positive	Positive	Positive	Positive	the overall product is too good, no improvements needed	Positive



127.0.0.1:5000/admin-dashboard

Admin Dashboard

Log Out

Anjali C	anjali@gmail.com	8904472276	Positive	Negative	Positive	Negative	Positive	Positive	you need to work on camera and sound quality of the product other things are fine	Positive
Manasavi	garg@gmail.com	1234567890	Positive	Positive	Negative	Negative	Negative	Negative	only the price and sound quality is fine, the over project is very poor	Negative
Pranjal	pranjal@gmail.com	7676789646	Positive	Positive	Positive	Positive	Positive	Positive	the overall product is too good, no improvements needed	Positive
Tejaswini	tejaswini@gmail.com	7887986754	Negative	Negative	Negative	Negative	Negative	Negative	the overall product is very poor ... i didnot like it at all	Negative
Sree Vaishnavi	sree@gmail.com	9078563412	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	it works fine	Neutral

➢ CONCLUSION:

This project successfully demonstrates the ability to determine aspect-specific sentiment within customer reviews, offering a granular understanding of customer opinions. By isolating sentiments for attributes like design and usability, the system provides actionable insights, enabling businesses to address specific customer concerns and enhance targeted attributes. The application of advanced NLP techniques and ML models ensures robust and accurate sentiment classification.

➢ FUTURE ENHANCEMENTS:

1. **Dynamic Aspect Discovery:** Enhance the system with dynamic aspect discovery capabilities, allowing it to identify and analyze emerging aspects of products or services without requiring predefined categories.
2. **Multimodal Sentiment Analysis**: Integrate multimodal analysis by combining textual data with other forms of feedback like images, videos, or audio to provide richer insights.
3. **Cross-Domain Adaptability**: Expand the model's adaptability to analyze feedback across multiple domains, such as hospitality, healthcare, and e-commerce, while maintaining high accuracy.
4. **Personalized Feedback Analysis**: Introduce customer segmentation to provide personalized insights based on user demographics, preferences, or past interactions.
5. **Real-Time Feedback Integration**: Enable the system to process and analyze feedback in real-time, supporting applications in live customer support or product launches.
6. **Emotion Recognition**: Enhance sentiment analysis with emotion recognition capabilities to identify underlying feelings like joy, frustration, or excitement.

These enhancements aim to make the system more versatile, accurate, and user-centric while addressing current limitations and preparing for future challenges.

➢ REFERENCES:

1. **Sentiment Analysis of Customer Feedback in Online Food Ordering Services**

- **Authors:** Bang Nguyen, Van-Ho Nguyen, Thanh Ho
- **Published Date:** December 2021
- **Link:** https://www.researchgate.net/publication/359862972_Sentiment_Analysis_of_Customer_Feedback_in_Online_Food_Ordering_Services

2. **Prediction of the Customers' Interests Using Sentiment Analysis in E-commerce Data for Comparison of Arabic, English, and Turkish Languages**

- **Authors:** Pelin Savcı, Banu Diri
- **Published Date:** March 2023
- **Link:** https://www.sciencedirect.com/science/article/pii/S131915782300054X

3. **Aspect-Oriented Sentiment Analysis: A Topic Modeling-Powered Approach**

- **Authors:** Anoop V. S., S. Asharaf
- **Published Date:** December 2018
- **Link:** https://www.researchgate.net/publication/329948113_Aspect-Oriented_Sentiment_Analysis_A_Topic_Modeling-Powered_Approach