AI CONTENT GENARATOR + SENTIMENT ANALYZER

Presented By
Student Name: Anjali Niranjan
College Name: Delhi Technical Campus, Greater Noida (Affiliated to Guru Gobind Singh Indraprastha University, Delhi)
Department: B.Tech. Computer Science & Engineering
Email ID: anjaliniranjan.official@gmail.com
AICTE Student ID: STU6407465d2f9be1678198

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach**
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

With the exponential growth of digital content consumption, businesses and individuals are under pressure to generate engaging, relevant, and emotionally resonant text. Manually creating such content is time-consuming and often inconsistent. Moreover, ensuring that the generated content carries the right emotional tone is crucial in applications like customer service, marketing, and journalism.

Key Challenges:

- Manual writing is slow and effort-intensive.
- Lack of tools that combine content generation with emotion detection.
- No real-time feedback on whether generated content is positive or negative.

PROPOSED SOLUTION

We propose an AI-based solution that combines text generation using Hugging Face transformer models with a custom-trained sentiment analysis classifier. The web app:

- Accepts a user prompt.
- Generates creative text using models like GPT-2.
- Analyzes the sentiment (positive/negative) of the generated text.
- Displays sentiment probabilities.
- Allows exporting results to CSV.

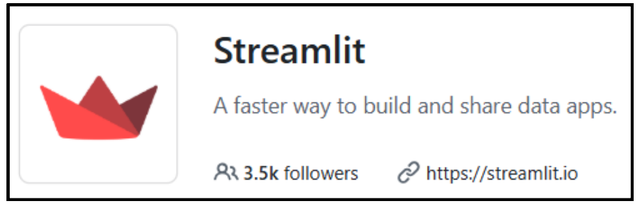
SYSTEM APPROACH

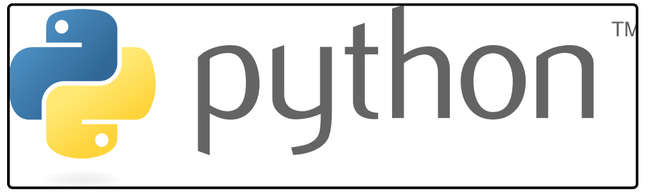
Technologies Used:

- Frontend/UI: Streamlit
- Backend: Python
- ML Frameworks: Hugging Face Transformers, Scikit-learn
- Deployment: Localhost

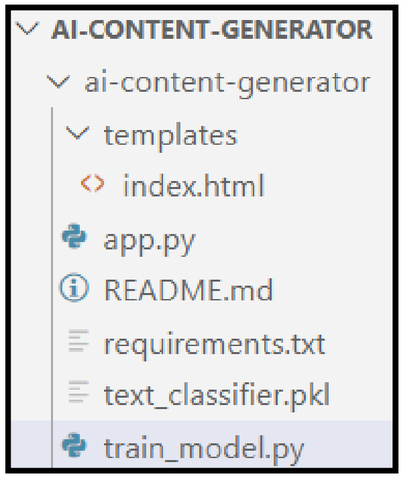
Libraries Required:

- transformers
- torch
- streamlit
- joblib
- pandas





FOLDER STRUCTURE & CODE SNIPPETS



app.py

ai-content-generator > app.py

```python
import streamlit as st
from transformers import pipeline, set_seed
import joblib
import pandas as pd
import base64
import time

# Load sentiment classifier
classifier = joblib.load("text_classifier.pkl")

# Page setup
st.set_page_config(page_title="✨ AI Generator Pro", layout="wide")
st.markdown("<h1 style='text-align:center;'>🧠 AI Content Generator + Sentiment Analyzer</h1>", unsafe_allow_html=True)

# Sidebar Config
with st.sidebar:
    st.header("⚙️ Settings")
    model_choice = st.selectbox("🤗 Choose Hugging Face Model", ["gpt2", "distilgpt2"])
    max_len = st.slider("📏 Max Generation Length", 50, 300, 100)
    num_return = st.selectbox("🎯 Number of Outputs", [1, 2, 3])
    show_probs = st.checkbox("🔍 Show Sentiment Probabilities", value=False)

# Load generator
@st.cache_resource
def load_generator(model_name):
    return pipeline("text-generation", model=model_name)

generator = load_generator(model_choice)
set_seed(42)

# Session history
if "history" not in st.session_state:
    st.session_state.history = []

# Input
prompt = st.text_input("✍️ Enter your creative prompt:")
```

train_model.py

ai-content-generator > train_model.py

```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
import joblib

texts = [
    "I love this product. It's amazing and works perfectly.",
    "This is the best thing I've ever bought.",
    "Absolutely wonderful! Highly recommend it.",
    "Terrible experience. I hate it.",
    "Worst service ever. I'm so disappointed.",
    "It was a waste of money and time."
]
labels = [1, 1, 1, 0, 0, 0]

pipeline = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('clf', LogisticRegression(max_iter=1000))
])

pipeline.fit(texts, labels)
joblib.dump(pipeline, 'text_classifier.pkl')

print("✅ Model trained and saved.")
```

ALGORITHM & DEPLOYMENT

Algorithm Selection

- Text Generation: GPT-2 or DistilGPT-2 from Hugging Face for natural language generation.
- Sentiment Classification: Pretrained classifier loaded via joblib.

Input Features

- User text prompt for generation.
- Generated text passed into the sentiment classifier.

Training Process

- The sentiment classifier was trained (not shown in current code) and saved as text_classifier.pkl.
- Model uses binary classification: Positive or Negative sentiment.

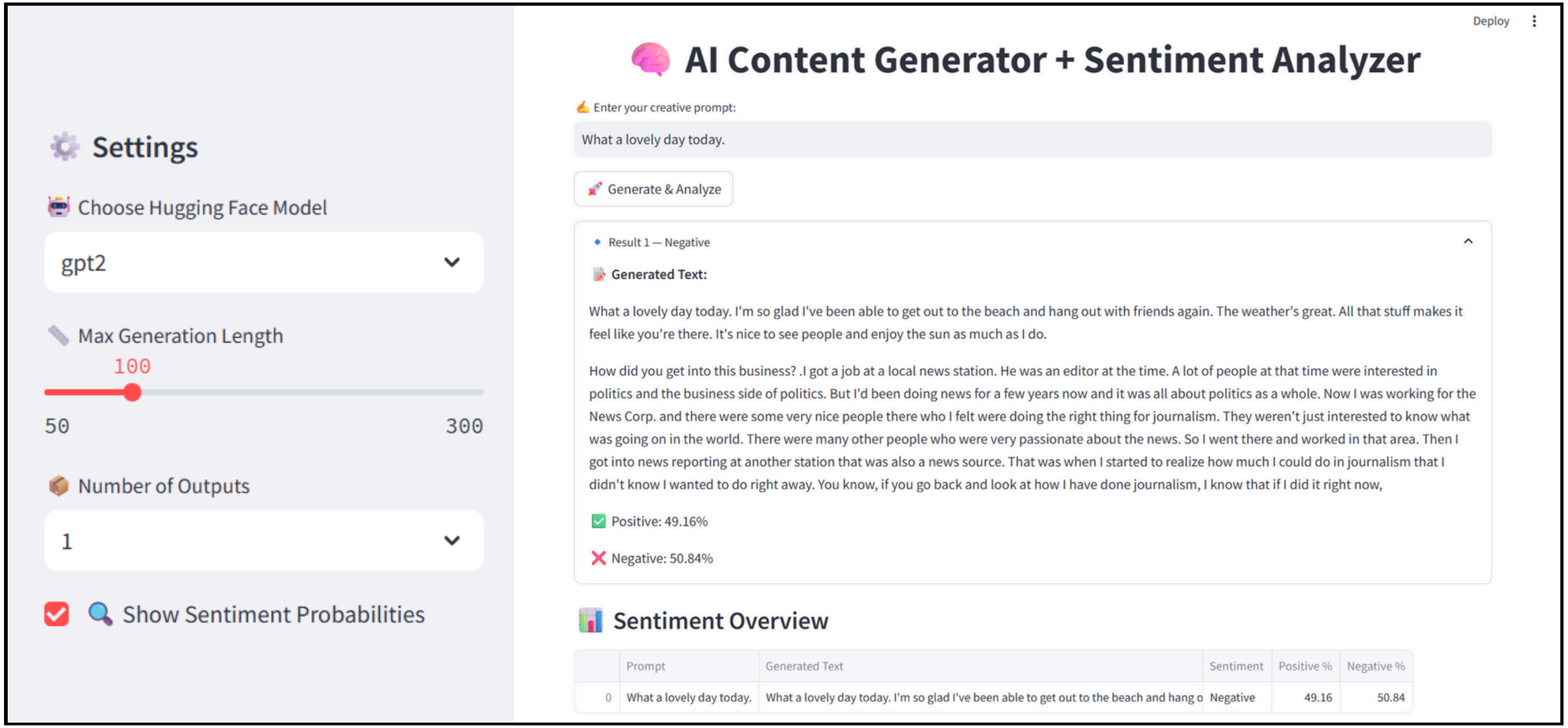
Prediction Process

- For each output text:
  - Use predict() and predict_proba() to determine sentiment and probability.
  - Display results interactively using Streamlit widgets.

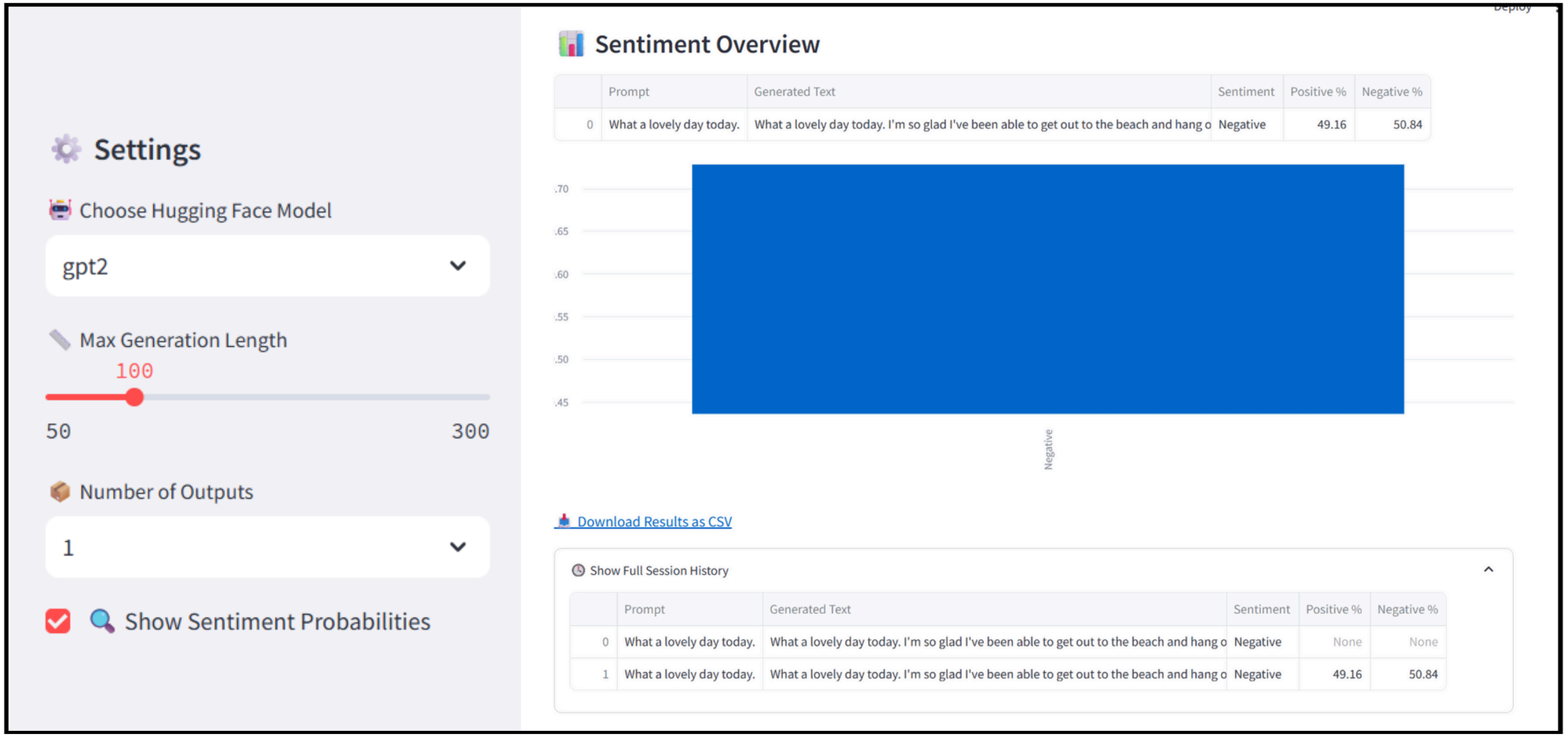
WORKFLOW

- User inputs prompt and configures settings.
- App generates multiple text outputs using the selected transformer model.
- Each output is analyzed for sentiment.
- Results are visualized using bar_chart() and dataframe().
- Download option provided for CSV export.

RESULT


Deploy
AI Content Generator + Sentiment Analyzer
Settings
Choose Hugging Face Model
gpt2
Max Generation Length
100
50
300
Number of Outputs
1
Show Sentiment Probabilities
Enter your creative prompt:
What a lovely day today.
Generate & Analyze
Result 1 — Negative
Generated Text:
What a lovely day today. I'm so glad I've been able to get out to the beach and hang out with friends again. The weather's great. All that stuff makes it feel like you're there. It's nice to see people and enjoy the sun as much as I do.
How did you get into this business? .I got a job at a local news station. He was an editor at the time. A lot of people at that time were interested in politics and the business side of politics. But I'd been doing news for a few years now and it was all about politics as a whole. Now I was working for the News Corp. and there were some very nice people there who I felt were doing the right thing for journalism. They weren't just interested to know what was going on in the world. There were many other people who were very passionate about the news. So I went there and worked in that area. Then I got into news reporting at another station that was also a news source. That was when I started to realize how much I could do in journalism that I didn't know I wanted to do right away. You know, if you go back and look at how I have done journalism, I know that if I did it right now,
Positive: 49.16%
Negative: 50.84%
Sentiment Overview
Prompt
Generated Text
Sentiment
Positive %
Negative %
0
What a lovely day today.
What a lovely day today. I'm so glad I've been able to get out to the beach and hang o
Negative
49.16
50.84

RESULT


Settings
Choose Hugging Face Model
gpt2
Max Generation Length
100
50
300
Number of Outputs
1
Show Sentiment Probabilities
Sentiment Overview
Prompt
Generated Text
Sentiment
Positive %
Negative %
0
What a lovely day today.
What a lovely day today. I'm so glad I've been able to get out to the beach and hang o
Negative
49.16
50.84
.70
.65
.60
.55
.50
.45
Negative
Download Results as CSV
Show Full Session History
Prompt
Generated Text
Sentiment
Positive %
Negative %
0
What a lovely day today.
What a lovely day today. I'm so glad I've been able to get out to the beach and hang o
Negative
None
None
1
What a lovely day today.
What a lovely day today. I'm so glad I've been able to get out to the beach and hang o
Negative
49.16
50.84

CONCLUSION

The AI Content Generator + Sentiment Analyzer:

- Empowers users to generate creative, relevant content quickly.
- Provides emotional analysis, increasing confidence in content tone.
- Offers flexibility with transformer model selection.
- Enhances productivity for marketers, content creators, and educators.

Achievements:

- Combined NLP generation and classification in a single app.
- Created a user-friendly and interactive interface.
- Exportable data supports further analytical use.

FUTURE SCOPE

- **Multi-class Emotion Detection** : Expand from binary sentiment to emotions like anger, joy, sadness, etc.
- **Fine-Tuned Transformer Models** : Use domain-specific fine-tuning (e.g., legal, medical, marketing).
- **Voice & Image Inputs** : Accept speech or images as input and generate corresponding text.
- **Advanced UI Features** : Use rich visualizations like pie charts, sentiment timelines.
- **Cloud Deployment** : Host on Heroku, AWS, or Streamlit Cloud for public access.
- **Improvement Of Model** : Model is not accurate at generating relevant content.

REFERENCES

- Hugging Face Transformers – https://huggingface.co/transformers/
- Streamlit – https://streamlit.io/
- Scikit-learn – https://scikit-learn.org/
- Python Documentation – https://docs.python.org/
- GitHub Repository - https://github.com/anjaliniranjan027/ai-content-sentiment-analyzer

Thank you...