|  |  |
| --- | --- |
| A black and white logo  Description automatically generated | Sentiment Analysis  Dipali Gite |
|  |  |
|  |  |
|  | A white bird with wings in a circle  Description automatically generatedAI Center of Excellence  Coherent Corporation  Revision Number: 1.0.0  Status: Production  Published: 12/31/2024 |

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description | Author |
| 1.0 | 12/26/2024 | Initial Requirement | Dipali Gite |
| 1.0 | 12/27/2024 | Project Setup and API Creation | Dipali Gite |
| 1.0 | 12/30/2024 | Streamlit and setup for API | Dipali Gite |
| 1.0 | 12/31/2024 | Sentiment Analysis | Dipali Gite |

Table of Contents

[1 Working Flow 3](#_Toc186724154)

[2 Initial Requirements 4](#_Toc186724155)

[2.1 Project Overview 4](#_Toc186724156)

[3 Features 4](#_Toc186724157)

[4 Tech Stack 4](#_Toc186724158)

[5 Setup Instructions 5](#_Toc186724159)

[5.1 Create Virtual Environment 5](#_Toc186724160)

[5.2 Install Dependencies 5](#_Toc186724161)

[6 Usage 5](#_Toc186724162)

[6.1 Running the application 5](#_Toc186724163)

[6.2 ui-design 6](#_Toc186724164)

# Working Flow

The working flow of the Sentiment Analysis system follows a structured sequence of tasks based on user input. The process is as follows:

1. **User Input**:
   * **File Upload:** Users can upload files in formats like CSV, Excel, TXT, or JPG.
   * **Manual Input:** Users can directly enter text in the provided text area for analysis.
2. **Document Display**:
   * The content of the uploaded document or entered text is processed and displayed for review.
   * Text from files (e.g., CSV, Excel, or JPG) is extracted using appropriate libraries, such as Pandas for CSV/Excel and PyTesseract for JPG.
   * The content is displayed under the section titled **"Uploaded Document"** in the app.
3. **Task Selection**:
   * The user can perform the following task:
     + **Sentiment Analysis:** Analyze the text to predict its sentiment Positive Negative.
4. **Task-based Processing**:  
   The system performs sentiment analysis on the provided input:
   * **Sentiment Prediction:**
     + Text is tokenized using Hugging Face’s AutoTokenizer.
     + The sentiment (Positive or Negative) is predicted using DistilBERT, a pre-trained model fine-tuned on SST-2.
     + The confidence score for the sentiment is also provided.
   * **Question Answering**:
     + The user can add any sentence and document file upload csv, excel, txt etc
     + The system processes the question, and generates an answer using an AI model in sentiment analysis.
5. **Task Output**:
   * The results are presented as follows:
     + **Input Text:** The original text provided by the user.
     + **Predicted Sentiment:** Either **Positive**, **Negative**.
     + **Confidence Score:** A percentage score indicating the model’s confidence in the prediction.
     + The output is displayed in a tabular format and can be downloaded as a CSV file.

# Initial Requirements

## Project Overview

The Sentiment Analysis App is a tool designed to predict the sentiment (Positive or Negative) of input text. The app enables users to upload documents in multiple formats or directly input text for analysis. It leverages a fine-tuned BERT-based model (DistilBERT) for accurate sentiment prediction:

# Features

* **Multi-format Input Support:** Users can upload CSV, Excel, TXT, or JPG files or enter text manually for analysis.
* **Sentiment Prediction:** Determines whether the sentiment of the text is Positive or Negative with associated confidence scores.
* **Results Display:** Outputs predictions in a tabular format within the app for easy review.
* **Downloadable Results:** Allows users to download the sentiment analysis results as a CSV file.

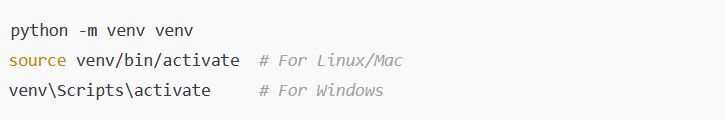
# Tech Stack

The Sentiment Analysis App utilizes the following technologies:

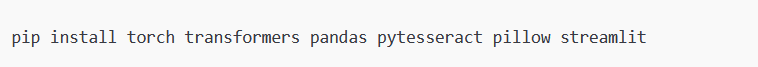
* **Python 3.8+**
* **Streamlit For building the user interface.**
* **PyTorch for Deep learning framework for running the DistilBERT model.**
* **Hugging Face Transformers: Provides pre-trained models and tokenizers.**
* **Pandas: For handling CSV/Excel data.**
* **PyTesseract: For text extraction from JPG images.**
* **Pillow: For image processing.Groq API for AI-based text analysis**

# Setup Instructions

## Create Virtual Environment

It is recommended to create a virtual environment to manage dependencies for this project. Use the following command: 

## Install Dependencies

Install the required Python libraries using the following command: 

# Usage

## Running the application

To start the application locally, use the following command:  
 ```streamlit run app.py```

A screenshot of a computer

Description automatically generated

## ui-design

A screenshot of a computer

Description automatically generated