**NEWS SENTIMENT ANALYSIS USING GOOGLE GEMINI**

**Problem Statement:** In the digital age, vast amounts of news articles are generated every day, influencing public opinion and decision-making. Understanding the sentiment behind these articles can provide valuable insights into trends, public perception, and market movements. This project aims to develop a news sentiment analysis application that fetches news articles and determines whether they convey a positive, negative, or neutral sentiment using Google's Gemini AI models.

**Aim of the Project:** The objective of this project is to analyze the sentiment of news articles using Google Generative AI (Gemini). The application is designed to:

1. Retrieve real-time news articles on a given topic using NewsAPI.
2. Utilize Google's Gemini AI to perform sentiment analysis.
3. Present sentiment analysis results in an easy-to-understand format.
4. Help users gain insights into the sentiment trends of current news topics.

**Techniques Used in the Project:**

1. **News Data Retrieval:**
   * News articles are fetched using the NewsAPI based on a user-defined topic.
   * The API returns multiple articles with titles and descriptions for analysis.
2. **Natural Language Processing (NLP) Using Gemini:**
   * Google's Generative AI (Gemini) is used to analyze the sentiment of the fetched news articles.
   * The model categorizes sentiment as Positive, Neutral, or Negative based on the text.
3. **Data Preprocessing:**
   * Extract relevant content (title and description) from the fetched articles.
   * Format the data for efficient processing by the AI model.
4. **Model Selection and Analysis:**
   * The application retrieves available AI models from Gemini and allows users to select one.
   * The chosen model processes the news text and provides sentiment insights.
5. **Visualization and User Interaction:**
   * Streamlit is used to build an interactive web application.
   * Results are displayed with sentiment markers for easy interpretation.

**Explanation of the Code:**

1. **Importing Necessary Libraries:**
   * streamlit: Used to create the web-based application interface.
   * requests: Fetches news data from the NewsAPI.
   * google.generativeai: Connects to Gemini AI for sentiment analysis.
   * os: Handles environment variables for API key security.
2. **Fetching News Articles:**
   * The function fetch\_news() makes an API request to NewsAPI to retrieve articles based on a user-input topic.
   * If the API call fails, an error message is displayed.
3. **Fetching Available AI Models:**
   * The function get\_available\_models() retrieves a list of available AI models from Gemini.
   * The user can select a preferred model for sentiment analysis.
4. **Performing Sentiment Analysis:**
   * The function analyze\_sentiment() sends the top five news articles to Gemini for sentiment analysis.
   * The model processes the text and classifies the sentiment of each article.
5. **Building the Streamlit Web Application:**
   * Users input a topic and provide API keys for NewsAPI and Gemini.
   * The application fetches news articles, performs sentiment analysis, and displays results.
   * Sentiment results are visually represented using different color indicators.

**Use of Python in the Project:** Python plays a crucial role in implementing this project through:

1. **Data Collection:** Using requests to interact with the NewsAPI.
2. **NLP Processing:** Using google.generativeai to analyze sentiment using Gemini AI.
3. **Data Handling:** Using Python lists and dictionaries to structure news content.
4. **Web Development:** Using streamlit to create an interactive user interface.
5. **Error Handling:** Implementing exception handling to manage API failures and invalid inputs.

**Conclusion:** The News Sentiment Analysis application successfully retrieves and analyzes news sentiment using Google Gemini AI. It provides users with real-time sentiment insights into news topics, helping them understand trends and public perception. While the current model effectively classifies sentiment, future improvements could include incorporating a broader range of news sources, enhancing AI training for better accuracy, and integrating visualization features for deeper sentiment trend analysis.