

Sentiment Analysis Project Using AI Classification

This project aims to develop a system that analyzes the sentiment (positive, negative, neutral, or etc...) of text data stored in a CSV database for many reasons such as :

Analyze customer reviews to understand their sentiment towards products, services, or brand image.

Track sentiment trends on social media to identify potential brand reputation issues.

Analyze customer sentiment to refine marketing campaigns and target specific audience segments.

Project Goals:

Build a Sentiment Analysis Model

- Train a model to automatically classify text data (e.g., reviews, social media posts) into positive, negative, or neutral sentiment categories.

Understand Public Opinion

- Analyze sentiment trends in the data to gain insights into public opinion on a particular topic, product, or brand.

Improve Customer Experience

- Analyze customer reviews to identify areas for improvement and enhance customer satisfaction.

Database Description:

The CSV database "[sentimentdataset.csv](#)" will contain text data and potentially additional information for context. Here's a possible structure:

ID

- Unique identifier for each text entry.

Text

- The actual text content to be analyzed for sentiment (e.g., product review, social media post).

Source

- Source of the text data (e.g., website review, Twitter post).

Topic

- Topic or category associated with the text (e.g., product category, brand name).

Label

- Pre-defined sentiment label (positive, negative, neutral, or etc...) for training a supervised model.

Project Requirements:

Data Preprocessing

Clean and prepare the text data in the CSV. This might involve:

- Removing punctuation and stop words (common words like "the", "a").
 - Lowercasing text.
 - Text normalization (e.g., converting slang terms).
- Stemming or lemmatization (reducing words to their base form).

Feature Engineering

Create numerical features from the text data that the model can understand. This might involve:

- **Word Embeddings:** Representing words as vectors in a high-dimensional space, capturing semantic relationships.
- **Bag-of-Words:** Representing documents as a collection of words, considering word frequency.
- **TF-IDF:** Representing documents based on word importance, considering both frequency and document rarity.

Model Selection and Training

Choose a suitable machine learning model for sentiment classification, such as:

- **Naive Bayes:** Simple and efficient for text classification.
- **Support Vector Machines (SVM):** Effective for high-dimensional data.
- **Logistic Regression:** Good baseline model, interpretable results.
- **Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) Networks:** Powerful for capturing context in sequential data like text. Train the model on the labeled data in the CSV, splitting it into training and testing sets.

Model Evaluation

Evaluate the performance of the model on the testing set using metrics like:

- **Accuracy:** Overall percentage of correctly classified sentiment labels.
- **Precision:** Ratio of true positives to all positive predictions (reduces false positives).
- **Recall:** Ratio of true positives to all actual positive examples (reduces false negatives).
- **F1-score:** Harmonic mean of precision and recall, provides a balanced view.

Mentor

TA. Andrew Magdy

Email: Andrew.magdy@cis.asu.edu.eg