Plagiarism Scan Report By SmallSEOTools

Report Generated on: Apr 04,2025

0%

Plagiarized Content

0%

Exact Plagiarized

0%

Partial Plagiarized

100%

Unique Content

Total Words: 367 **Total Characters:** 2544 **Plagiarized Sentences:** 0 **Unique Sentences:** 17 (100%)

Content Checked for Plagiarism

Text Sentiment Predictor: A Lightweight NLP Tool
Name: Sarthak Waghmode
Roll No: 21103A0016
Branch: BE ETRX A
Abstract
This project introduces a web-based sentiment prediction tool that classifies text as either positive or negative. By using a mix of Natural Language Processing (NLP) techniques and a Decision Tree-based machine learning model, it offers real-time predictions, supports bulk CSV processing, and includes interactive visualizations for deeper insights.
1. System Overview
The application consists of two parts:

Backend: A Flask-based API handling text cleaning, tokenization, negation handling, stemming, and prediction using a trained model.
Frontend: A Streamlit interface for entering text, uploading files, and viewing results with interactive charts.
2. Key Features
Instant Sentiment Analysis with confidence scores
Bulk Predictions via CSV file uploads
Smart Negation Detection (e.g., “not bad” → positive)
CSV Preview & Downloadable Results
Interactive Graphs showing sentiment breakdown

3. NLP Workflow
1. Text Cleaning: Converts input to lowercase, removes special characters
2. Tokenization: Splits into individual words
3. Stemming: Uses Porter Stemmer to reduce words
4. Negation & Keyword Detection: Identifies key sentiment indicators
5. Vectorization & Scaling: Prepares text for prediction
6. Prediction: Uses model + rule-based logic

4. Model Training & Rules
The classifier is trained on labeled data including positive, negative, and negated phrases. Additional rules are applied for:

Negation flipping

Handling expressions like 'not good'

Confidence boosting for key sentiment words

5. CSV Support

Accepts files with “Text”, “Sentence”, etc. as columns

Displays preview before processing

Adds a “Predicted sentiment” column to output

7. Requirements

1. Python 3.6+
2. Flask, Streamlit
3. scikit-learn, NLTK, Pandas, NumPy, Matplotlib

Result:

1. Step 1: Upload the .csv file / text:

I will be using Kaggle for downloading dataset

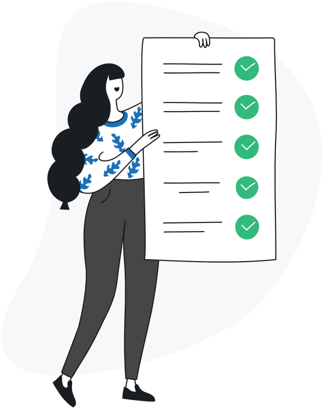
For instance, I have downloaded an Airline Review dataset from Kaggle.

https://www.kaggle.com/datasets/khushipitroda/airline-reviews

I will upload this onto my StreamLit Website for Bulk Prediction

2. Step 2: The model goes through the entire document and analyzes the column labelled as Text/Sentence/Text-Sentence and analyzes the entire column

3. Step 3: Once we click on Predict it will give us a new document with an additional column in it, stating Predicted sentiment. This new document is downloadable or for quick results it will provide us with a pie chart.



No Plagiarism Found